Auton Robot (2012) 32:189–205
DOI 10.1007/s10514-011-9265-9

Pareto-optimal coordination of multiple robots with safety guarantees

Rongxin Cui · Bo Gao · Ji Guo

Received: 23 January 2011 / Accepted: 28 November 2011 / Published online: 14 December 2011

Abstract This paper investigates the coordination of multiple robots with pre-specified paths, considering motion safety and minimizing the traveling time. A method to estimate possible collision point along the local paths of the robots is proposed. The repulsive potential energy is computed based on the distances between the robots and the potential collision points. This repulsive potential energy is used as the cost map of the probabilistic roadmap (PRM), which is constructed in the coordination space for multiple robots taking into account both motion time cost and safety cost. We propose a search method on the PRM to obtain the Pareto-optimal coordination solution for multiple robots. Both simulation and experimental results are presented to demonstrate the effectiveness of the algorithms.

Keywords Path planning · Multiple robots · Coordination space · Probabilistic roadmap (PRM) · Pareto-optimal · Safety guarantees

1 Introduction

Multiple autonomous robots, including automatic ground vehicles (AGVs), unmanned underwater vehicles (UUVs), unmanned ariel vehicles (UAVs), etc., are being employed in a growing number of areas, such as military applications, space/subsea explorations, and disaster relief (Durfee 1999; Fua et al. 2007; Ge and Fua 2004; Shanmugavel et al. 2009; Cui et al. 2010). A common scenario is when a group of agents are dispatched to perform tasks, each agent has to avoid other dispatched agents while moving towards its respective destination. Therefore, the problem of motion coordination for multiple robots should be considered, which is beyond the scope of simple path planning problem to avoid static obstacles. We should consider the inter-collision/confliction between robots and plan the motion of several robots simultaneously.

Several approaches have been proposed to solve the collision avoidance problem for multiple robots. An overview of recently proposed methods was presented in Kuchar and Yang (2002), where the methods can be categorized into *centralized* and *decentralized*. In the *centralized* algorithms, a multiple robots system is always being treated as a single complex system and each robot in the system is considered as a part of component of this complex system. Then the motion coordination problem can be solved as planning the motions of different components in the complex system. It could be solved by searching the configuration space of the complex system (Barraquand and Latombe 1991). This method is suitable for the off-line computation as it takes the information of all the robots. It uses semi-definite programming or non-linear dynamic programming to solve a global optimization problem (Hu et al. 2002), and always can get the optimal solution. The disadvantage is that it is a challenging task to gather all information at a central location in practice. In the *decentralized* methods (Purwin et al. 2008; Schouwenaars et al. 2004; Mataric 1995; Ge and Cui 2002b), each robot is treated as separated individuals and its motion is planned independently by treating other robots as moving obstacles. A cost-based negotiation process based algorithm was proposed for the cooperative decentralized path-planning of multiple agents in Purwin et

R. Cui (✉) · B. Gao
College of Marine Engineering, Northwestern Polytechnical University, Xi'an 710072, P.R. China
e-mail: rongxin.cui@gmail.com

J. Guo
College of Physics and Electrical Engineering, Anyang Normal University, Anyang 455000, P.R. China

al. (2008), of which the basic algorithm was augmented by the introduction of way-points. The agents considered in Purwin et al. (2008) can come to a full stop, so the algorithm needs to be extended for the applications to the aircrafts which cannot have zero forward speed. A decentralized receding horizon strategy based on mixed integer linear programming (MILP) was proposed for safe decentralized trajectory planning of multiple aircraft in Schouwenaars et al. (2004), where the safety is guaranteed by maintaining a dynamically feasible trajectory for each aircraft that terminates in a loiter pattern. Conflicts between multiple aircrafts were resolved in a sequential and decentralized fashion, in which each aircraft takes into account the latest trajectory and loiter pattern of the other aircrafts. The decentralized algorithms are easy to implement and can be used for the real-time implementation based on the information gathered by the onboard sensors. However, the global optimal solution cannot be guaranteed.

In this work, we consider the coordination motion planning for multiple robots in a common workspace, such as managing a collection of AGVs in a factory. Assuming that the information of all robots can be gathered at a central location, we use the *centralized* algorithm to obtain the global optimal solution. The centralized algorithms can be categorized into *direct* or *decoupled* (Choset et al. 2005). A *direct* approach typically solves the problem considering all the constraints simultaneously, and it is usually challenging to find a solution which satisfied all constraints. A *decoupled* approach divides the process of solving problems into several stages. For coordinated trajectory planning, the problem is decoupled into collision-free path planning followed by time scaling in two stages (Kant and Zucker 1986; Choset et al. 2005; Chitsaz et al. 2004; Ghrist et al. 2005a). Initially, collision-free paths are computed for each robot individually, without taking into account the other robots but simply considering the obstacles of the workspace. In the second stage, coordination is achieved by computing the relative velocities of the robots along their individual paths and avoiding collisions among them (Kant and Zucker 1986). Decoupled planning does not increase the dimensionality of the configuration space, and is quite practical (LaValle 2006).

Motivated by Chitsaz et al. (2004), Ghrist et al. (2005a), we propose a new decoupled coordinated trajectory planning approach. On the basis of the pre-specified paths, this approach uses PRM to solve the problem in the coordination space. The *coordination space* is a state space formed by the pre-specified paths of all the robots and schedules the motions of the robots along their paths so that they will not collide (Ghrist et al. 2005a). Besides of the motion cost, it also considers the motion safety of coordination.

Each robot has an independent criterion to evaluate its performance, such as finish its path in minimum time to maximize the productivity. As there may be intersections between the pre-specified paths resulting from the fact that the robot-robot interaction was ignored, the motion of the robots will affect others. Hence, the objectives of the robots are conflicting. In this case, we are interested in finding the *Pareto-optimal* coordination solutions by treating multiple robots coordination as a multi-objective optimization problem (Ghrist et al. 2005a, 2005b). This notion of *Pareto optimality* is widely used in mathematical economics to model individual consumers striving to optimize distinct economic goals, and the Pareto solutions are the ones that there exist no solutions that are better for all robots.

The main contributions of this work are: (i) a method to estimate the potential collision point along the local paths of the robots is proposed, and the repulsive potential is computed based on the distances to the potential collision points of the robots; (ii) a new approach to evaluate the motion time cost and safety cost is presented, which facilities the construction of the coordination roadmap; and (iii) the searching heuristic function is defined taking into account both safety cost and motion cost, following by Dijkstra algorithm to search the optimal solution. The coordination methodology of multiple robots proposed in this work can be applied to the areas of automatic air traffic conflict detection and resolution, and that for land or sea based vehicles as well.

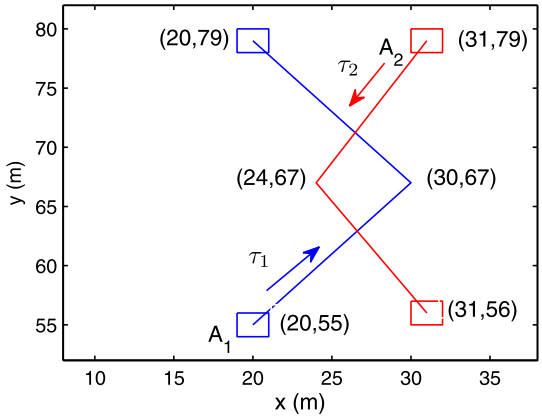
The remainder of the paper is organized as follows. In Sect. 2, preliminaries and the problem formulation are presented. The safety cost map construction procedure is presented in Sect. 3, followed by roadmap construction in Sect. 4. The motion cost and the search strategy are described in Sect. 5. Simulation and experimental results are presented in Sects. 6 and 7, respectively. Conclusions are drawn in Sect. 8.

2 Preliminaries and problem formulation

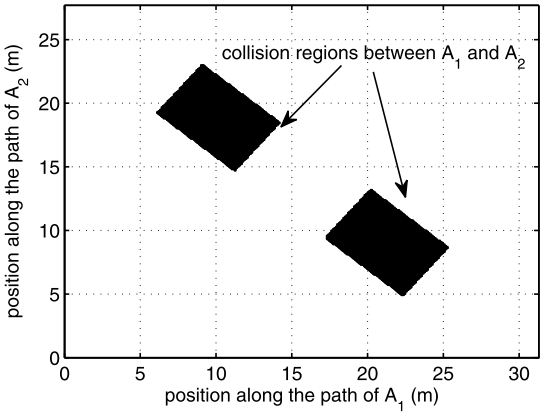
In this work, we investigate the problem of coordinating multiple robots with pre-specified paths considering motion safety and minimizing the elapse time. Our approach is to determine the velocity of each robot along its pre-specified path.

Definition 1 Let Q be a topological space, and a *path*, τ, is defined as a continuous function, $\tau : [0, 1] \to Q$. Alternatively, $\mathbb{R}$, $\mathbb{R}^2$, or $\mathbb{R}^3$ may be used for the domain of τ. Each point along the path is given by $\tau(s)$ for some $s \in [0, 1]$. A *trajectory* is defined as a path with an associated velocity profile on each point along the path, i.e., the trajectory needs to be specified over time (LaValle 2006).

We assume there are m robots in the team, and each robot, $A_i, i = 1, \ldots, m$, is treated as a rigid object capable of moving in a workspace that is a bounded subset of $\mathbb{R}^2$ or $\mathbb{R}^3$.



(a) Pre-specified paths of two robots.



(b) Coordination space of two robots.

Fig. 1 Pre-specified paths and coordination space of two robots

Each robot is independently constrained to traverse a pre-specified path, $\tau_i,\ i = 1, \ldots, m$. This path is computed from the initial configuration q_i^{ini} to goal configuration q_i^{goal} of the robot, not taking into account the other robots but considering the static obstacles of the workspace (Choset et al. 2005). For m robots, an m-dimensional state space called a *coordination space* is that schedules the motions of the robots along their paths so that they will not collide (Ghrist et al. 2005a). Mathematically, the coordination space X_τ is defined as

$$X_\tau := (\tau_1 \times \cdots \times \tau_m) - \mathcal{O}, \tag{1}$$

where $\mathcal{O}$ denotes the collision region. This collision region contains the positions on which the collisions will happen between the robots. One important feature is that time will not be explicitly represented in the coordination space. If a trajectory is computed in the coordination space, then the explicit timings can be computed from this trajectory (Ghrist et al. 2005a).

It is clear that following conditions should be satisfied to avoid collisions between any pair of the robots.

$$A_j\big(C(t)\big) \cap A_k\big(C(t)\big) = \emptyset, \quad \forall j \neq k,\ t \in [0, 1], \tag{2}$$

where $A_j(\cdot)$ is the subset of the workspace occupied by robot A_j, and is related to the physical size and shape of the robot.

Example 1 To show the pre-specified paths and corresponding coordination space clearly, we give an example with two robots in Fig. 1. Specifically, Fig. 1(a) shows the pre-specified paths of the robots, and Fig. 1(b) shows the corresponding coordination space. Both robots considered here are squares with sizes 2 m × 2 m and there are intersections between the paths of them. As a result, there are collision regions in the coordination space as shown in Fig. 1(b). This collision region is a set of positions along the paths where the robots will experience collisions/conflicts. It is noted that the lengths of the paths of A_1 and A_2 are 31.241 and 27.785, respectively. This length is consistent with the length of the related axis in Fig. 1(b).

One important feature of the coordination space is that the large sizes of robots will result in large collision regions due to the fact that the large robot will occupy more workspace. For the point masses, there is almost no collision unless the robots are exactly at the same position.

If the rigorous shapes and sizes of the robots are used in constructing the coordination space, and all the robots can follow their paths precisely, then any free trajectory which has no intersection with the collision regions in the coordination space can ensure the collision avoidance between robots. Then considering only the motion time cost, such as the one in Chitsaz et al. (2004), Ghrist et al. (2005a), we can obtain the Pareto-optimal coordination solution to minimize the elapse time of the system. In practice, however, robots may violate their trajectories due to the constraints on the tracking ability of robots. In addition, the physical sizes and shapes of robots considered in the construction of the coordination space may be coarse. We may not able to ensure the collision avoidance between robots considering only the time cost. Therefore, in the coordination of multiple robots, motion safety should be considered as well.

The objective of coordinated planning is to obtain $C : [0, 1] \to X_\tau$ in the coordination space, where the initial state is $\{q_1^{\text{ini}}, \ldots, q_m^{\text{ini}}\}$ and the goal state is $\{q_1^{\text{goal}}, \ldots, q_m^{\text{goal}}\}$. Then the explicit velocity scheduling information for each robot will be computed from the planned trajectory. Let cost functional, $L_i(C),\ i = 1, \ldots, m$, separately measure the cost function of each robot A_i. The goal of safety coordination planning becomes to find the optimal C^* to minimize

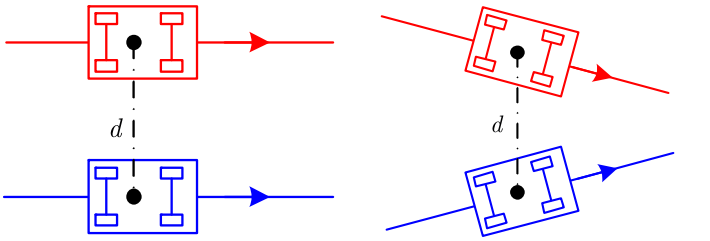


Fig. 2 Two robots with same distance between them

(a) Two robots move with parallel paths. (b) Two robots move with crossing paths.

the $L_i(C)$ over the duration of motion

$$C^* = \arg\min_{C \in U} \{L_i(C),\ i = 1, \ldots, m\}, \tag{3}$$

where U is the set of all the possible coordination results.

While performing tasks, distinct robot possesses distinct goal and/or cost function $L_i(C)$ for evaluating performance, such as finishing its own path in minimum time to maximize its productivity. Then each robot wishes to optimize its cost function independently of the others. However, due to the fact that there are intersections of paths in the workspace between the robots, the motion of the robots will affect others. As a consequence, their objectives are conflicting. Because a multi-objective optimization task can consider several objective functions simultaneously, we treat the coordination planning as a multi-objective optimization problem in this work. Our perspective is to find the Pareto-optimal coordination solution (Parker 1997; Liu and Li 2002; Ghrist et al. 2005b; Chitsaz et al. 2004).

Definition 2 The *Pareto-optimization* is a vector-valued optimization, preserving all cost function data, and widely used in mathematical economics to model individual consumers striving to optimize distinct economic goals. Each Pareto-optimal strategy is one for which there exists no strategy that would be better for all robots (Ghrist et al. 2005b; Chitsaz et al. 2004). In other words, compared with other coordination solution $C' \in U$, the Pareto-optimal solution C^* in this work satisfies

$$\begin{cases} L_i(C^*) \leq L_i(C'), & \forall i = 1, \ldots, m \\ L_i(C^*) < L_i(C'), & \exists i. \end{cases} \tag{4}$$

It is noted that there is usually no single optimal solution for the multi-objective optimization, but a set of alternatives with different trade-offs. Despite the existence of multiple Pareto-optimal solutions, in practice, usually only one of these solutions is to be chosen. Motivated by Chitsaz et al. (2004), Ghrist et al. (2005a), our objective is to plan the velocities of robots along the pre-specified paths to avoid collisions. We use the criterion that minimize the elapse time of the system (the time that the last robot completes its path) and the safety cost to obtain the unique solution.

3 Safety cost map construction

In this section, the safety cost map construction method is introduced. The safety cost is used to measure the collision risk for the robot traveling along the pre-specified path.

3.1 Safety criterion

The most intuitive criterion used to evaluate the collision risk between robots is the distance between robots in the configuration space. For example, one common used artificial potential method is to fill the workspace of the robots with an artificial potential field in which the robot is repulsed away from the obstacles (Ge and Cui 2002a, 2002b). Voronoi graph is another kind of map built using the distance among robots over the configuration space (Mortezaie 1991). However, the distance between robots cannot always reveal the true possibility of the collision between robots. As shown in Fig. 2, the distance d between two robots in Fig. 2(a) is the same as the one in Fig. 2(b). There is no collision risk in case of Fig. 2(a), but the robots in Fig. 2(b) will conflict with each other at the intersection point following the specified paths. Motivated by guidance theory in the aerospace literatures, the collision cone concept was proposed in Chakravarthy and Ghose (1998) for the real-time collision detection and obstacle avoidance of robots. It provides a convenient means to determine whether any two moving objects are on a collision course in the configuration space. It also reduces the engagement between two irregularly shaped objects into an equivalent engagement between a point and a circle.

In this work, we propose a method to estimate the possible collision point for the robot traveling along the pre-specified path. The repulsive potential energy for each point along the path will be computed to evaluate the collision risk and used as the safety cost in the roadmap construction stage. Different from the collision cone approach (Chakravarthy and Ghose 1998), we construct the repulsive potential field in the coordination space. In addition, the physical sizes and shapes of robots have been considered in the construction of the coordination space. The safety cost computed here is dependent on the local path of each robot,

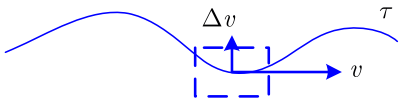
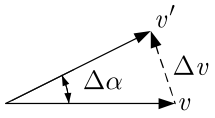


Fig. 3 The velocity and its rate of change

Fig. 4 Relationship between the change of velocity and the angle



and is not dependent on the physical size and shape of the robot.

3.2 Possible collision position

We estimate the possible collision position of the robot by using its local path. When a robot travels along the path $\tau : [0,1] \to Q$, the relationship between the velocity v and its rate of change Δv at point $\tau(s_0)$ can be obtained from the path as shown in Fig. 3. Suppose the robot moves in the 2D plane, then the curvature of the path, $K(\tau(s))$, can be written as

$$K\big(\tau(s)\big) = \frac{|\tau''(s)|}{\sqrt[3]{1+\tau'^2(s)}}. \tag{5}$$

Let $\Delta\ell$ be the displacement of the robot with velocity v in time interval Δt, then $\Delta\ell = v\Delta t$. Define $\Delta\alpha$ as the direction change of the velocity as shown in Fig. 4, then $\Delta v = v\Delta\alpha$.

From (5), we have

$$\lim_{\Delta t\to 0} \frac{|\Delta\alpha(t)|}{|\Delta\ell|} = K\big(\tau(s)\big), \tag{6}$$

and can conclude that

$$\lim_{\Delta t\to 0} \frac{|\Delta v(t)|}{|v(t)|^2\Delta t} = K\big(\tau(s)\big). \tag{7}$$

To compute the potential nearest collision position using current position $\tau(s_0)$, we introduce an estimated path τ_e, which is computed using the local path at current position. Assuming that the velocity and its rate of change keep unchanged in a small time interval, then from (7), we know that the curvature of the estimated path τ_e is constant. Therefore, the estimated path τ_e has a constant curvature K, i.e., the estimated path will be a circle with radius $r = \frac{1}{K(\tau(s))}|_{s=s_0}$.

As shown in Fig. 5, q_{ec} is the possible collision point of the two robots computed from estimated paths τ_{e1} and τ_{e2}. We will use the distance between current position $\tau(s_0)$ and the possible collision point q_{ec} on the estimated path τ_e to

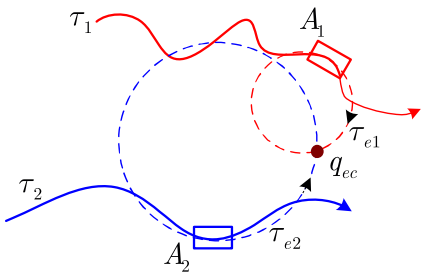


Fig. 5 Possible collision point

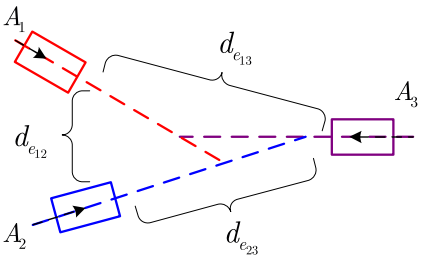


Fig. 6 Estimated distances in multiple robots system

evaluate the collision risk of the robot traveling along the pre-specified path.

3.3 Distance computation in multiple robots system

If there are only two robots, A_i and A_j, in coordination as shown in Fig. 5, then the estimated distance $d_{e_{ij}}$ can be computed using the summation of the distance to the potential collision point of the robots. However, there are always more than two robots in the multiple robots system. For every point x in the coordination space X_τ, it contains the states of all the robots. Paths of all the robots should be taken into account to compute the distance between the robot and the estimated collision position.

Motivated by computing the potential function using sensor-based implementation, we compute all the estimated distances $d_{e_{ij}}$ between each robot pair and select the minimal one as

$$d_e = \min\{d_{e_{ij}}\}, \quad i,j = 1,\ldots,m, \tag{8}$$

as shown in Fig. 6.

3.4 Repulsive potential and cost map

The repulsive potential is computed in accordance with the computed distance d_e, with the purpose to keep the robot away from collision regions in the coordination space. The strength of repulsive potential depends on the proximity to possible collision position. The repulsive potential energy increases when the robot is closer to the collision point. The repulsive potential function used in this work takes the fol-

lowing form (Choset et al. 2005)

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{d_e} - \frac{1}{d^*})^2, & d_e \le d^* \\ 0, & d_e > d^*, \end{cases} \tag{9}$$

where $d^* \in \mathbb{R}$ is the maximum distance that allows to ignore risk of collision sufficiently far away from the collision point, and η is a positive scaling factor. Being consistent with the commonly used potential function, η is used to scale the effect of the repulsive potential, and is usually determined by trial. The large η will result in a large repulsive potential energy.

By utilizing the definition of repulsive potential energy, the cost map X_{cost} can be viewed as function of repulsive potential energy, i.e., $X_{\text{cost}}(U_{\text{rep}}(q))$. In this work, we define the cost map X_{cost} as

$$X_{\text{cost}}(q) = U_{\text{rep}}(q). \tag{10}$$

It is noted that U_{rep} in (9) is used to represent conflict danger for point-vehicles in traditional potential field method for robot path planning. In this paper, U_{rep} is used as the cost map to evaluate the collision risk traveling along the pre-specified path. It is computed from the local path; therefore, it depends on the pre-specified paths, but not depends on the physical sizes and shapes of the robots.

In practice, we can set a maximum limit, $U_{\max}$, on the cost map. If the repulsive potential energy exceeds $U_{\max}$ on some positions along the pre-specified path, these positions will be regarded as being too risky to move. $U_{\max}$ can be determined according to following two factors: (i) The accuracy of the physical sizes and shapes of robots considered in the coordination space construction stage: if it is roughly considered, then we can choose a lower limit, otherwise, we can choose a higher one; and (ii) Tracking abilities of robots: if the tracking error of the robot is larger, then $U_{\max}$ can be selected lower, otherwise, it can be chosen larger. Note that the potential function should be smooth in the traditional potential field method because its gradient will be computed as the potential force. In this work, U_{rep} is not smooth due to $U_{\max}$. This will not affect the computation because we use U_{rep} as the cost map directly, and do not need to compute the potential force, i.e., the gradient of the potential function.

Example 2 Figure 7 gives two examples of the pre-specified paths, corresponding coordination space, and safety cost map under our criterion for two robots. In the examples, both robots are squares with size 10 m × 10 m. We set $U_{\max} = 10$ in both examples.

4 Coordination roadmap

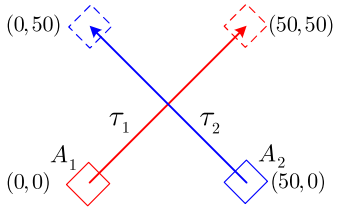
To construct the data structure once and then use that data structure to plan subsequent coordination trajectories more quickly, we build a roadmap in X_τ. This is motivated by the roadmap concept for path planning in configuration space (Choset et al. 2005). We give the definition of the coordination roadmap as follows.

Definition 3 A union of one-dimensional curves is a *coordination roadmap* (CRM) for m robots if for all $\{x_1^{\text{ini}}, \ldots, x_m^{\text{ini}}\}$ and $\{x_1^{\text{goal}}, \ldots, x_m^{\text{goal}}\} \in X_\tau$ that can be connected by a trajectory, the following properties hold: (i) Accessibility: there exists a trajectory from $\{x_1^{\text{ini}}, \ldots, x_m^{\text{ini}}\}$ to some $\{x_1, \ldots, x_m\} \in \text{CRM}$, (ii) Departability: there exists a trajectory from some $\{x_1, \ldots, x_m\} \in \text{CRM}$ to $\{x_1^{\text{ini}}, \ldots, x_m^{\text{ini}}\}$, and (iii) Connectivity: there exists a trajectory in CRM between $\{x_1^{\text{ini}}, \ldots, x_m^{\text{ini}}\}$ and $\{x_1^{\text{goal}}, \ldots, x_m^{\text{goal}}\}$.

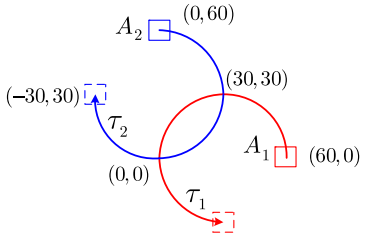
The roadmap is represented by a graph, $G_\tau = (V_\tau, E_\tau)$, where V_τ and E_τ are the vertices set and edges set, respectively. The vertices in V_τ are the points in the coordination space in which there are no collisions between robots. An edge $(x_1, x_2) \in E_\tau$ corresponds to a collision-free trajectory connecting states x_1 and x_2. It is not easy to obtain the coordination solution in X_τ directly. For the path planning problem in configuration space, sampling-based planning was developed for the purpose of reducing planning complexity, of which the computational complexity is related to the number of samples other than the dimension of planning space (Kavraki et al. 1996). Sampling-based planning algorithm can consider at most a countable number of samples. It employs a variety of strategies for generating samples and for connecting the samples with paths to obtain solutions to the path planning problems.

There are several strategies about building the sampling roadmap in configuration space, including probabilistic roadmap planner (PRM), expansive-spaces trees (EST), rapidly-exploring random trees (RRT), and sampling-based roadmap of trees (SRT) (LaValle and Hinrichsen 2001; Hsu 2000; Kuffner and LaValle 2000; Plaku and Kavraki 2005). In the PRM, it creates a roadmap in configuration space, and uses rather coarse sampling to obtain the vertices of the roadmap and very fine sampling to obtain the roadmap edges, which are free paths between node configurations (LaValle and Hinrichsen 2001). Compared to PRM, EST relies on the ability to avoid over sampling any region. Sampling points are chosen biased toward configurations whose neighborhoods are not dense (Hsu 2000). Different from EST, RRT chooses the sampling points nearest to the goal to avoid over sampling (Kuffner and LaValle 2000). In SRT, the vertices of the roadmap are not single configurations but trees and connections between trees are computed by a bidirectional tree algorithm such as EST or RRT (Plaku and Kavraki 2005).

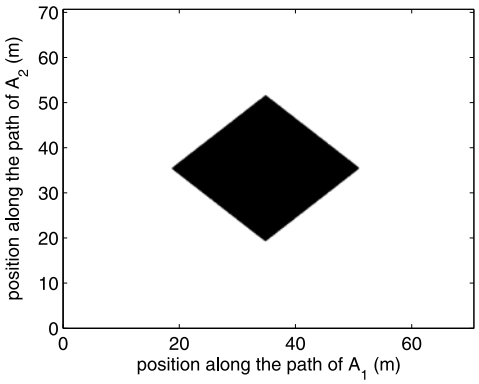
To compute the optimal solution in coordination space, we utilize PRM planner in the coordination space to construct the CRM. The main procedure can be divided into



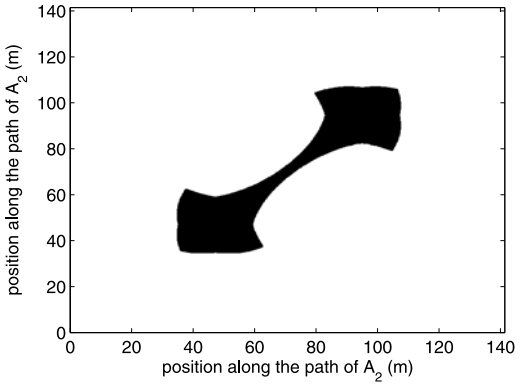
(a) Two robots with straight paths.



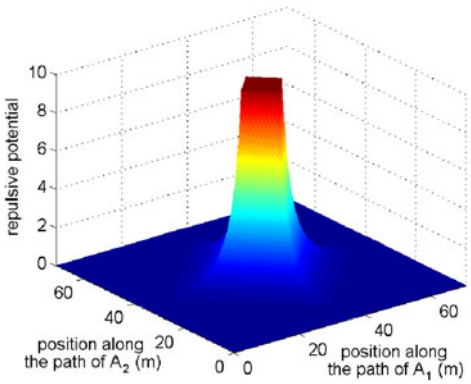
(b) Two robots with turning paths.



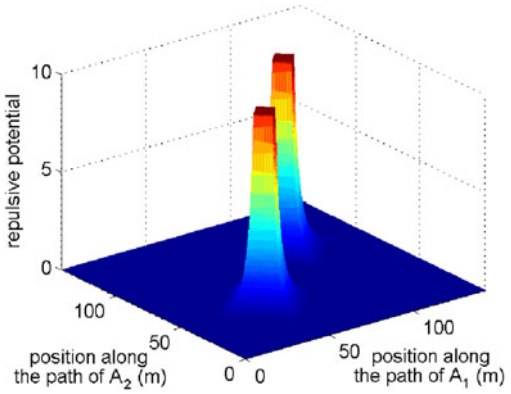
(c) Coordination space with straight paths.



(d) Coordination space with turning paths.



(e) Repulsive potential energy with straight paths.



(f) Repulsive potential energy with turning paths.

Fig. 7 Examples of repulsive potential energy with different paths for two robots

two steps: creating samples in the coordination space and connecting these samples in a suitable way to construct the coordination roadmap.

4.1 Sampling in the coordination space

Sampling means that selecting suitable points in the coordination space to constitute the vertices set V_τ in the roadmap. In this work, assume that the generation of states is done randomly from a uniform distribution. As defined in (1), the coordination space contains all the possible coordination solutions for the robots following their paths. Therefore, a sample in the coordination space corresponds to a combination of robot's particular position on their paths. This property is utilized to create the sampling points. For each robot A_i, we randomly choose a point x_i on its path τ_i. Then the combination of these points, $x = (x_1, \ldots, x_m)$, will be a sample in X_τ.

Since there are collision regions/obstacles in the coordination space, it is clear that the sampling point, x, which corresponds to the position that the robots collide with others, cannot be utilized for planning. The useless point x has the following property.

$$A_j(x_j) \cap A_k(x_k) \neq \emptyset, \quad \exists j, k = 1, \ldots, m, \text{ and } j \neq k. \quad (11)$$

In case of a collision point being selected, we need to increase the connectivity of the roadmap. In this case, we pro-

Algorithm 1: Vertices construction

Input: Paths of all the robots $\{\tau_i | i = 1, \ldots, m\}$
Output: A set of roadmap vertices V_τ
$V_\tau \leftarrow \emptyset$;
while $|V|_\tau < n$ **do**
  $x = (x_1, \ldots, x_m) \leftarrow$ random points set from $\{\tau_i | i = 1, \ldots, m\}$;
  for *All* x_i, x_j **do**
    if $A_i(x_i) \cap A_j(x_j) \neq \emptyset$ **then**
      repeat
        $x_i \leftarrow x_i +$ `step`
      until $A_i(x_i) \cap A_j(x_j) = \emptyset$;
  $V_\tau \leftarrow V_\tau \cup \{x\}$

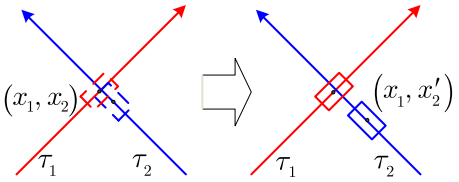


Fig. 8 Collision position transferring

pose a point transferring strategy to transfer the collision point to the feasible one nearby to increase the density of samples in this area. Consequently, it can increase the connectivity of the roadmap. From Ghrist et al. (2005a), we know that the position where several robots collide can be viewed as the collision between different combinations of two robots. Therefore, we can handle the pair-wise conflict resolution first. The proposed vertex construction algorithm for the CRM is shown in Algorithm 1, where `step` is defined as the minimum distance variation of the robot along its path, and n is the number of vertices to put into the coordination roadmap. If at the new position, no collision happens between the robots, then the new state x' can be put into the roadmap. As shown in Fig. 8, $x = (x_1, x_2)$ cannot be put into the roadmap, so we randomly select a robot, A_2, to change its position randomly along the pre-specified path. The position of A_1 is kept unchanged. Then the new state, $x' = (x_1, x_2')$, can be put into the roadmap.

It is noted that the number of samples gives a measure of the work that needs to be done. Hence it can be used as a measure of the complexity of the proposed sampling based planning algorithm. In this work, the number of the sampling points should ensure the three properties of CRM tenable first, and the larger number of sampling points can obtain better solution while increasing the computational complexity of the algorithm.

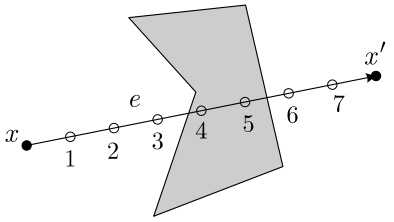


Fig. 9 Sampling along the *straight line* between two states x and x' in coordination space. The *numbers* correspond to the order in which the strategy checks the samples for collision

4.2 Concatenation of the sampling points

To construct the roadmap, the sampling points should be connected to neighbors to form the edges in the roadmap. In general, two aspects should be considered: (i) the created edge $e \in E_\tau$ should be feasible, which means that the robots will not collide while traveling on the edge; and (ii) the cost on the edge should be computed, which is necessary for the subsequent coordination planning.

Motion safety should be considered in building roadmap. It is necessary to judge whether edge e connected two given states, x and x', is feasible to be put into the CRM. In this work, both collision checking and safety checking are used to do this judgment. Motivated by the collision checking in path planning (Schwarzer et al. 2003), we use the step-size checking algorithm to check whether there are collisions on edge e in coordination space (Kuffner 2004). As shown in Fig. 9, in the collision checking, edge e is discretized into a number of states $(x^1, \ldots, x^\ell)$, where $x = x^1$ and $x' = x^\ell$. The distance between any two consecutive states x^i and x^{i+1} is equal to some positive constant `step_size`. On each step point, we check whether the collision takes place as described in (11). If so, the edge e is infeasible and will not be put into the CRM.

Safety checking is carried out with the collision checking simultaneously, and its purpose is to compare the safety cost on the step with the allowed maximum limit of repulsive potential, $U_{\max}$. If the safety cost is larger than $U_{\max}$, the robots will collide with large probability, then the edge will not be put into the coordination roadmap. On the basis of the methods of sampling and connecting vertices, the edge construction algorithm is proposed as shown in Algorithm 2. Same as other sampling based algorithms, the coordination roadmap constructed in this work is not unique because the sampling points are random selected.

Remark 1 It is noted that the value of `step-size` is problem specific and is defined by the user. In general, it needs to be very small to guarantee that all collisions are found in coordination space. In addition, `step_size` can be determined relating the distance between the robot and collision region in the coordination space to the maximum length of the trajectory traced out by any point on the robot.

Algorithm 2: Edges construction

Input: A set of roadmap vertices V_τ, and paths of robots $\{\tau_i | i = 1, \ldots, m\}$
Output: A set of roadmap edges E_τ
$E_\tau \leftarrow \emptyset$;
for *All* $x \in E_\tau$ **do**
  $N \leftarrow k$ nearest neighbor vertices in V_τ;
  for *All* $x' \in N$ **do**
    if $(x, x') \notin E_\tau$ *and feasible* **then**
      $E_\tau \leftarrow E_\tau \cup (x, x')$;

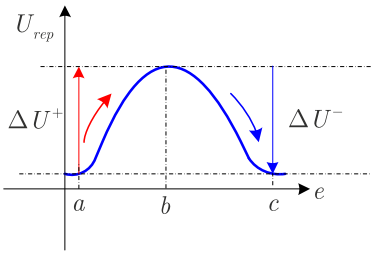


Fig. 10 Different direction of change in safety cost with same quantity

In this paper, we consider both motion cost and safety cost of the edge in the CRM to optimize the coordination planning. The motion cost, L_m, is dependent on the moving distance of the robot, but not dependent on the motion direction. The safety cost is influenced greatly by the edge direction, which is affected by the motion direction as shown in Fig. 5. Different moving direction will result in the different direction of change of the repulsive potential energy. It is clear that only the positive change will harm the motion safety as shown in Fig. 10. Therefore, the cost of the edge can be written as

$$L(e) = L_m(e) + \mu L_s(e), \tag{12}$$

where $L_m(e)$ and $L_s(e) = \int_{dU_{\text{rep}}>0} U_{\text{rep}}\, de$ are the motion cost and safety cost, respectively; μ is an adjustable weight to measure the proportion of the two costs. If we choose a large μ, the safety cost will be considered more in the coordination planning, then more safety trajectory will be find out in the coordination space.

While using the step-size checking algorithm to compute the edge cost, based on the step Δe, (12) can be discretized as

$$L(e) = L_m(e) + L'_s(e), \tag{13}$$

where

$$L'_s(e) = \mu \sum_{\Delta U_{\text{rep}}(i)>0} \left(\frac{U_{\text{rep}}(i) + U_{\text{rep}}(i-1)}{2} \right) \Delta e_i \tag{14}$$

with $\Delta U_{\text{rep}}(i) = U_{\text{rep}}(i) - U_{\text{rep}}(i-1)$.

The discretization transforms the optimization problem to a discrete planning problem in the coordination space. This leads to complete planning approaches, which are guaranteed for a solution when it exists, or correctly report failure if it does not. For discrete planning, it will be important that the set of all the possible states is countable, and this will reduce the computational complexity in practical applications.

By using the proposed algorithms, the coordination roadmaps of Example 1 can be constructed as shown in Fig. 11. It is observed that the constructed roadmap will be closer to the collision regions if ignoring the safety cost.

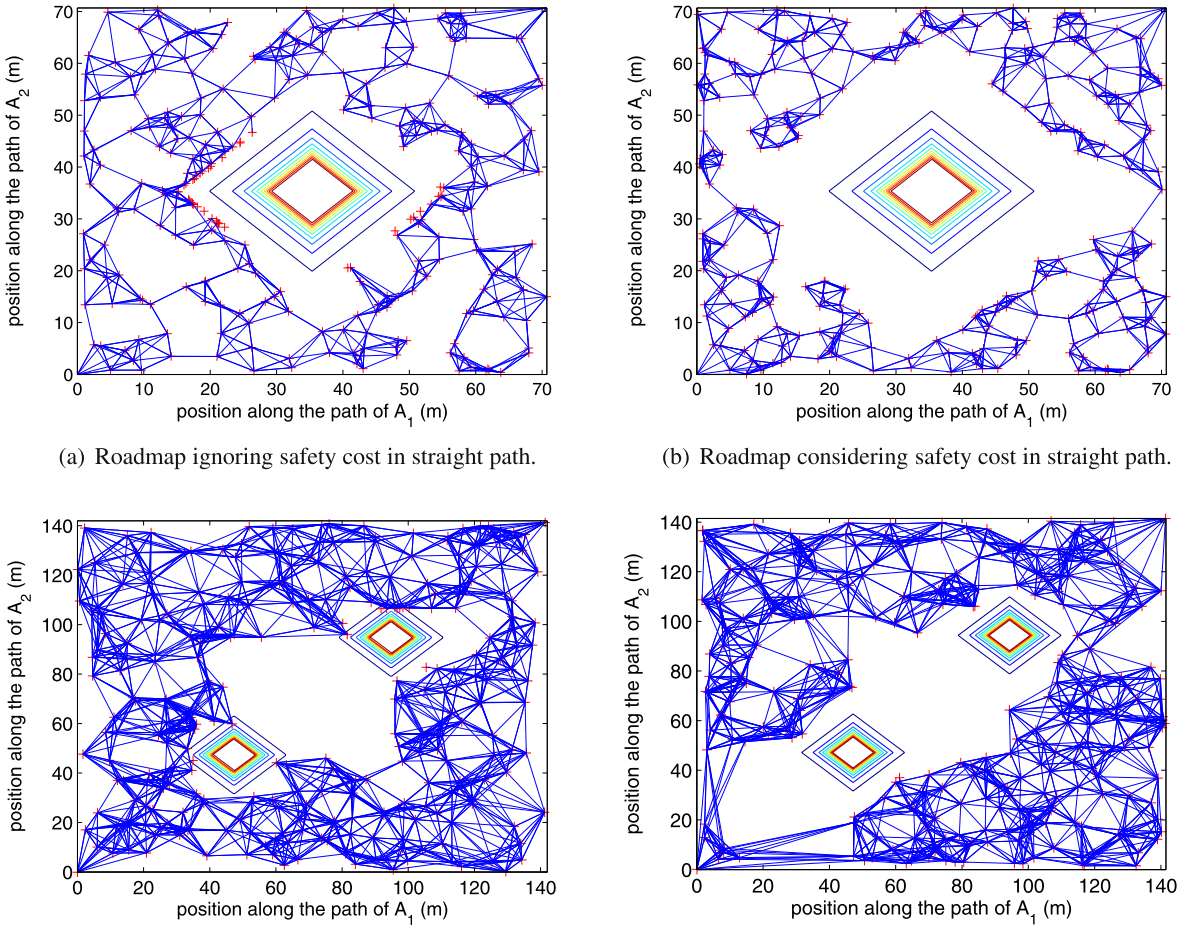
5 Motion cost and searching strategy

Having constructed the coordination roadmap G_τ in X_τ, we can find out the optimized coordination solution C^* through searching the roadmap. Besides of the feasibility of solution, we should also take into account the system efficiency. In this section, we focus on the derivation of the motion cost function on the edge e in the CRM, i.e., to determine $L_m(e)$ in (12).

Several criterions have been proposed to evaluate the motion cost on a path, including minimizing the average time elapsed of system (Hu et al. 2002), and minimizing the performance time of each robot added together (Shin and Zheng 1992). Such approaches are common and may be appropriate in some cases. However, it is important to recognize that scalarization of a vector of m criteria occurs in the process of managing m robots. Each robot has its own cost function, e.g., elapsed time. These m criteria are then converted, often in an arbitrary manner, into a single criterion to be optimized. In this work, we prefer to investigate the optimization problem for multiple robot coordination without scalarizing the vector-valued cost function. This brand of optimization is faithful in the sense that the costs of all the robots are involved by scalarization. A left-greedy searching strategy by introducing the Pareto-optimization to optimize the vector constituted by the efficiency of each robot was proposed in Ghrist et al. (2005a), of which the purpose is to minimize the elapse time of the system. In this work, we propose a new optimal planning algorithm which can obtain the Pareto-optimized solution on the basis of the constructed CRM as well.

As shown in Fig. 12, e is an edge in E_τ of G_τ. Suppose the velocity of A_i is v_i, the angle between the trajectory of A_i and e is θ_i, and the distance of robot A_i traveling along the edge e is e_i. If the robots follow the edge e, the following equation holds

$$\begin{cases} \frac{v_1}{\cos(\theta_1)} = \frac{v_2}{\cos(\theta_2)} = \cdots = \frac{v_m}{\cos(\theta_m)}, & \cos(\theta_i) \neq 0 \\ v_i = 0, & \text{otherwise.} \end{cases} \tag{15}$$



(a) Roadmap ignoring safety cost in straight path.

(b) Roadmap considering safety cost in straight path.

(c) Roadmap ignoring safety cost in turning path.

(d) Roadmap considering safety cost in turning path.

Fig. 11 Examples of coordination roadmaps for two robots. The *parallelograms* are the contours of the repulsive potential energy. Symbols "+" correspond to the vertices of the roadmap. The *straight lines* between "+" correspond to edges of the roadmap

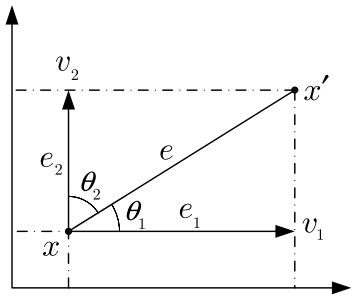


Fig. 12 Edges in the coordination space

Define by t_i the time for robot A_i to complete the motion along edge e, and then the system efficiency can be described as $[t_1, \ldots, t_m]^T$. From (15), we know that on the fixed edge e, the ratio of the velocities between the robots is a constant. As defined in Pareto-optimization, at least one robot travels with its maximum velocity on the edge, i.e.,

$$v_i(t) = v_{i,\max}, \quad \exists i \in \{1, \ldots, m\}, \text{ and } t \in [t_{i0}, t_i], \tag{16}$$

where $v_{i,\max}$ is the maximum speed of A_i, and t_{i0} is the starting time of A_i moving on the edge. Therefore, to obtain the optimal coordination, we can set the time that the last robot takes on this edge as the motion cost, i.e.,

$$L_m(e) = \max_{i=1,\ldots,m} \left\{ \frac{|e_i|}{v_{i,\max}} \right\}. \tag{17}$$

Suppose that we find out $L_m(e) = \frac{|e_k|}{v_{k,\max}}$, robot A_k will travel with maximum speed $v_{k,\max}$ on the edge. Then the velocities of other robots can be computed by (15).

Considering the safety cost in Sect. 4, and combine with (13) and (17), we define the searching heuristic function as

$$H = \min_{e \in E_\tau} \left\{ \sum L(e) \right\}. \tag{18}$$

Based on the heuristic function (18), we can search for the trajectory on the CRM from $(0, \ldots, 0)$ to $(|\tau_1|, \ldots, |\tau_m|)$ by using A^* or Dijkstra algorithm to obtain the optimal solution C^* for multiple robots coordination. As the heuristic func-

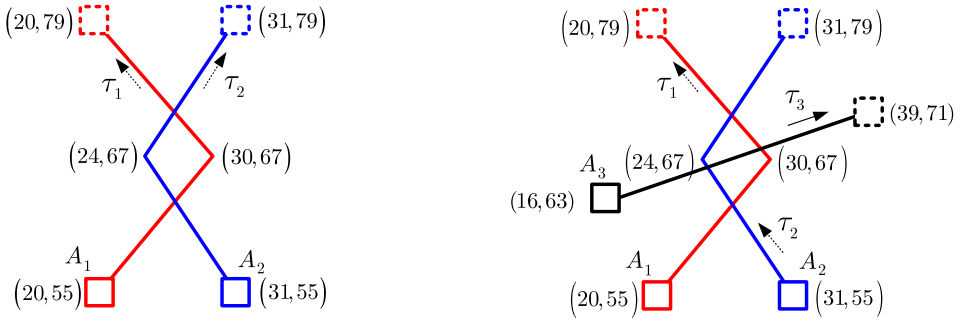


Fig. 13 Pre-specified paths of the robots

(a) Pre-specified paths of two robots.

(b) Pre-specified paths of three robots.

tion (18) is derived taking into account both motion time and motion safety on each edge, the solution C^* could realize the optimal solution in combination of these two costs. In addition, by introducing the cost function (17) and ensuring that there is at least one robot traveling with maximum velocity on each edge, we know that the solution satisfies (4) in the definition of Pareto-optimization. Therefore, the search result C^* is a Pareto-optimal solution for the system.

Remark 2 The proposed algorithm is inherited from the probabilistically completeness PRM planning (Ladd and Kavraki 2002). Therefore, we can always find the optimal solution as long as the number of samples is large enough.

Remark 3 In this work, we considered that the robots objective is to minimize the time to travel a path, and each robot has a maximum velocity constraint. For other applications, suitable class of cost functions can be used by deforming the geometry of the coordination space, and some constraints can be added in selecting feasible edges put into the coordination roadmap. For the fixed wing aircraft, which has a minimum velocity requirement to stay aloft, we should set some constraints in the edge construction stage. Specifically, the edges whose slopes satisfy some requirements can be put into the CRM only. In addition, the algorithm is used for off-line computation when the pre-specified paths of all the robots are available. Then we can get the Pareto-optimal solution for the multiple robots system. If the algorithm is used for the real-time computation when only the local paths of robots are available, we may obtain the local optimal solution, rather than the global Pareto-optimal solution.

6 Simulation results

In the numerical simulation, we consider the systems with both two robots and three robots cases. This is because the dimension of the coordination space agrees with the number of robots in the system. We can show the coordination space in 2D and 3D space intuitively. The pre-specified paths of robots are shown in Figs. 13(a) and 13(b), respectively. In the simulation, the maximum velocity of each robot is given in Table 1. All the robots are squares with sizes 2 m × 2 m.

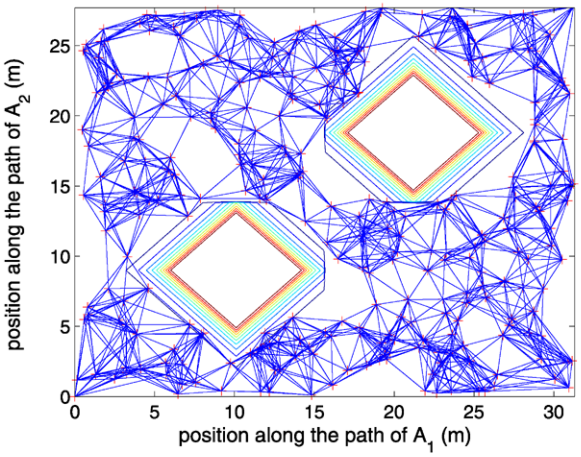
Table 1 Maximum velocities of the robots

Number of robots	$v_{1,\max}$	$v_{2,\max}$	$v_{3,\max}$
2	2 m/s	1 m/s	–
3	2 m/s	1 m/s	1 m/s

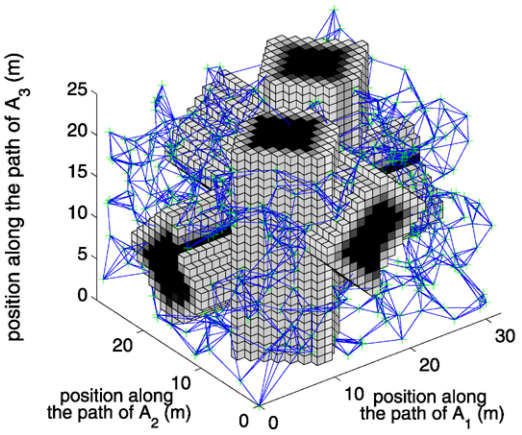
By applying proposed algorithms, the coordination planing results in coordination space are shown in Fig. 14. The CRMs of the systems with two robots and three robots are shown in Figs. 14(a) and 14(b), respectively. Two coordination solutions are computed as shown in Figs. 14(c) and 14(d), where the red lines are the coordination solutions considering time cost only, and the green lines are the coordination solutions considering both time cost and safety cost. For the system with three robots, the coordination planning results in A_1–A_2 profile, and A_1–A_3 profile in the coordination space are shown in Figs. 14(e) and 14(f), respectively.

The motion cost, safety cost and the maximum safety potential value in the coordinated planning are shown in Table 2. In general, one would just indicate a single overall cost for the optimization result. The purpose of the data shown in Table 2 is to compare the results of different cases. The time cost and safety cost are computed by using (17) and (14), respectively. It is noted that the values with brackets are not involved in the coordination planning computation. The value of the safety cost with bracket is computed by (14) from the existing solutions.

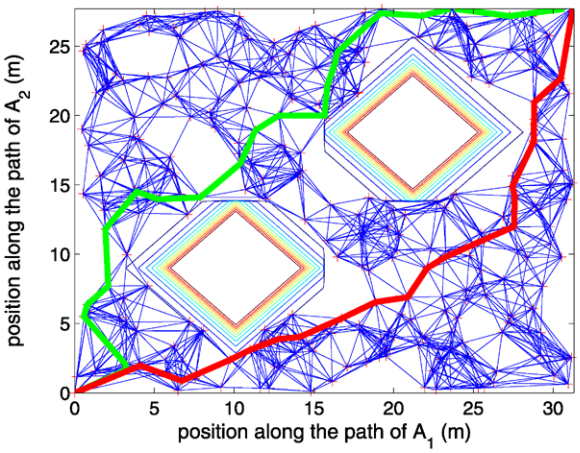
We show the velocity schedule results with respected to time in Fig. 15. It is observed that these velocities have discrete jumps. These planned velocities would be sent to the control systems of the robots as the desired velocities. The actual velocities will be continuous with errors with respect to the desired velocities. Although the control system de-



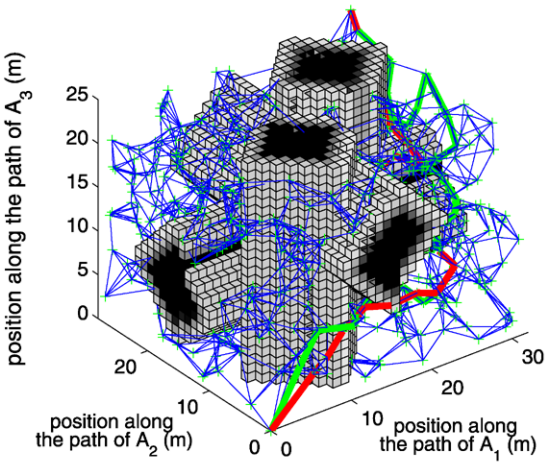
(a) CRM of two robots.



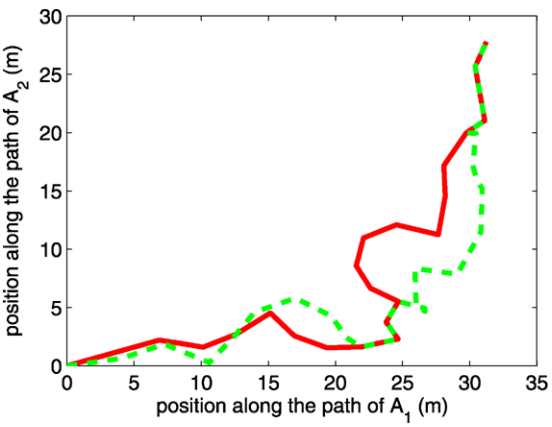
(b) CRM of three robots.



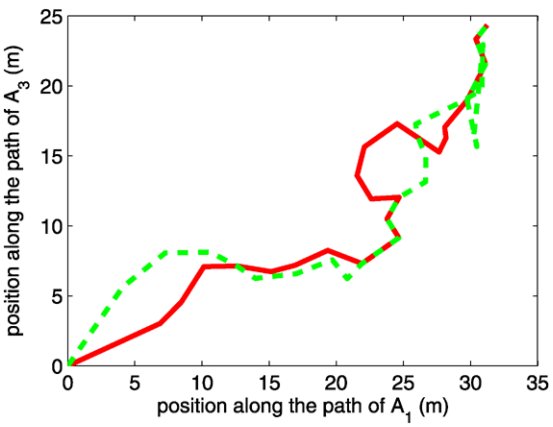
(c) Pareto-optimal solution for two robots.



(d) Pareto-optimal solution for three robots.



(e) Pareto-optimal solution in A_1-A_2 profile.

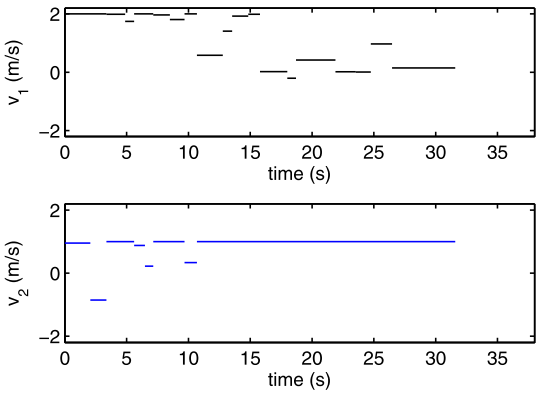


(f) Pareto-optimal solution in A_1-A_3 profile.

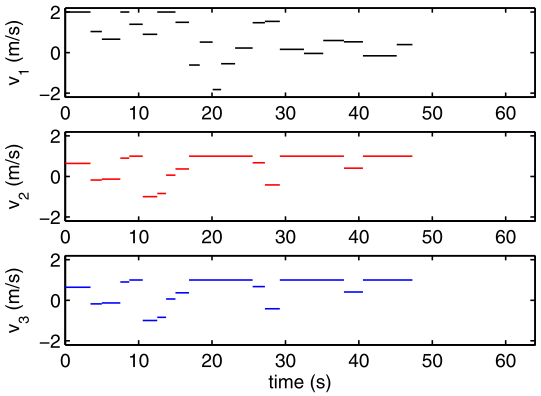
Fig. 14 (Color online) Coordination roadmaps and the Pareto-optimal solutions considering time cost only (*red lines*) and considering both time cost and safety cost (*green lines*)

Table 2 Costs in different cases

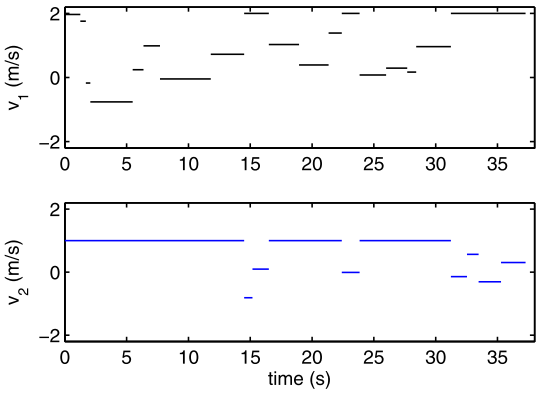
Cases	Time cost	Safety cost	Maximum repulsive potential value
Two robots ignoring safety cost	31.58	(144.24)	9.19
Two robots considering both costs	37.27	66.68	8.48
Three robots ignoring safety cost	47.30	(187.31)	6.71
Three robots considering both costs	63.52	117.04	5.63



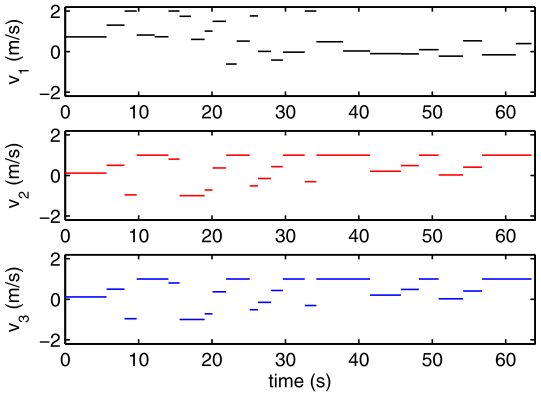
(a) Velocities of two robots ignoring safety cost.



(b) Velocities of three robots ignoring safety cost.



(c) Velocities of two robots considering both costs.



(d) Velocities of three robots considering both costs.

Fig. 15 Velocities schedule results for the robots

sign is out of the scope of this work, we considered that there would be trajectory tracking errors of robots in practice. Hence the safety cost is considered in the coordination planning.

From the simulation results, it is observed that, compared with the case considering both safety cost and time cost, the robots could move with higher speeds in the case ignoring safety cost, as shown in Figs. 15(a) and 15(c). The elapse time of the system will be shorter considering time cost only, as shown in Table 2. However, as shown in Figs. 14(c) and 14(d), the planned path is closer to the collision regions regardless of safety cost. In addition, that the maximum safety potential value is larger in Table 2. Therefore, it is probable that the robots will collide with others which considering time cost only. As such, we can conclude that the coordinated planning solutions considering both costs are more practical and useful in the real applications.

7 Experimental results

7.1 Experiment setup

Two Pioneer3-DX robots are used in the experiment. We use a laptop to connect with the robots via wireless communication as shown in Fig. 16. The coordinated planning results are computed off-line on the laptop, and the velocity commands sent to the robots every 0.1 s. The Pioneer3-DX robot itself has an embedded computer to control its

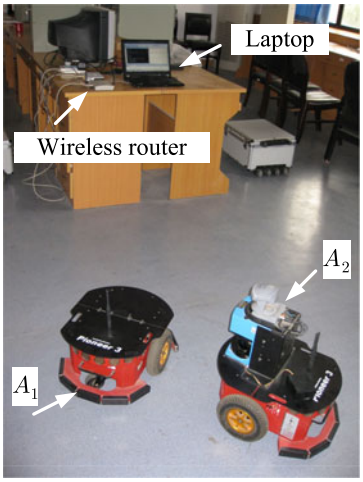


Fig. 16 Experiment setup

motors to follow the desired velocity. In addition, we set the maximum velocities of the robots as $v_{1,\max} = 0.2$ m/s, $v_{2,\max} = 0.1$ m/s, respectively.

7.2 Experimental results and analysis

Two cases in the experiment are considered. In the first case, the robots move on the paths using the scheduled velocity information computed by our algorithm. In the second case, we set that A_1 moves toward its target directly with its maximum speed. A_2 takes A_1 as a moving obstacle, and use *front-avoid* mechanism provided in the Advanced Robotics Interface for Applications (ARIA)[1] software to compute the velocity in real-time (Laffary 2002). The purpose of the *front-avoid* mechanism is to avoid front obstacles through controlling both translation and rotation of the robot. If the distance to the obstacle is within the safety range, the robot will slow down and turn an angle to avoid the obstacle. Users need to set three parameters to use this mechanism, including the safety range d_{avoid} at which to turn on the *front-avoid* mechanism, the angle $\Delta\theta$ to turn relative to current heading, and the speed v_{avoid} at which to go while avoiding an obstacle. If the distance between the closet obstacle and the robot is d_{ro}, the speed of the robot v_{ao} will be adjusted as

$$v_{\text{ao}} = \begin{cases} v_{\text{avoid}} \frac{d_{\text{ro}}}{d_{\text{avoid}}}, & \frac{1}{2} d_{\text{avoid}} < d_{\text{ro}} \leq d_{\text{avoid}} \\ 0, & d_{\text{ro}} \leq \frac{1}{2} d_{\text{avoid}}. \end{cases} \tag{19}$$

In this experiment, we set $d_{\text{avoid}} = 0.6$ m, $\Delta\theta = 15°$, $v_{\text{avoid}} = 0.08$ m/s for A_2. Front sonar are used to detect the distance to the obstacle for the robot.

[1] ARIA is a software provided by MobileRobots, Inc. USA for the Pioneer robots. Users can develop their own algorithms based on the existing classes and interfaces provided by the software.

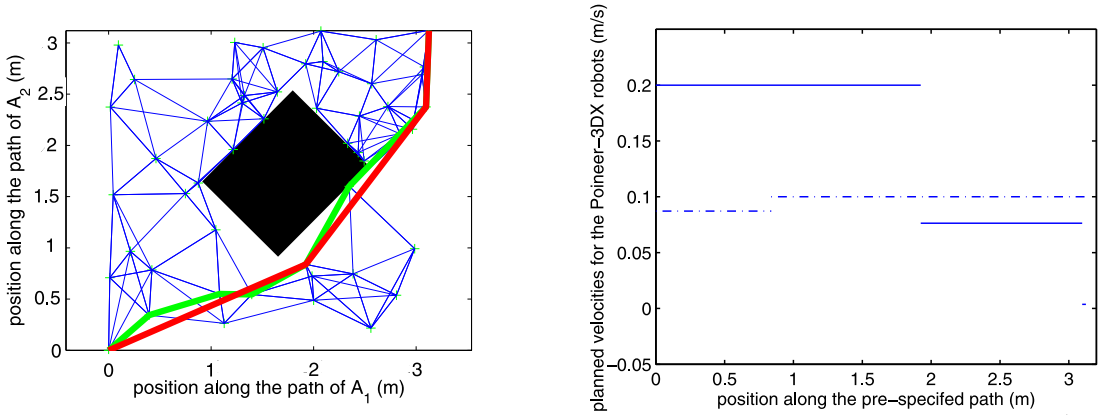
In the first case, the pre-specified path of A_1 is defined as a line from point $(-1.1, -2.4)$ to $(0.9, 0)$, and the pre-specified path of A_2 is a line from point $(1.1, -2.4)$ to $(-0.9, 0)$. Both the lengths of the paths are 3.12 m. It is clear that there are intersections between the paths of the robots. The coordination planning solutions using our algorithms is shown in Fig. 17. From Fig. 17(a), we can see that the path considering safety cost is far away from the collision region in the coordination space. The planed velocities of the robots are shown in Fig. 17(b). The elapse time of the experiment is 27.6 s, and the sequence of the snapshots for the experiment using solutions considering both costs is shown in Fig. 18. The blue areas are the regions where the onboard sonar can detect. A_2 will slow down and let A_1 run first near the collision area as shown in Fig. 18(b). In addition, there is at least one robot moving with maximum speed at any time in the experiment, as shown in Fig. 19(a).

In the second case, A_1 moves with its maximum speed until approaching the target. A_2 will not slow down where near the collision region, but will try to take a detour treating A_1 as an obstacle. It recomputes the velocity and turning angle at each step. This results in large fluctuation in its velocity as shown in Fig. 19(b). Before A_1 approaches the target, A_2 runs on the right of A_2 as shown in Figs. 20(b) and 20(c). After A_1 stops and is not in front of A_2, A_2 moves with maximum speed to the target. The elapse time is 83.3 s for this case, and is longer than the first case. This is because A_2 takes A_1 as a moving obstacle and it takes long tour to the target.

8 Conclusion

In this work, a new method for solving the motion coordination problem for coordination of multiple robots on the pre-specified paths has been proposed. The strategy consists of two parts, building the roadmap in coordination space and searching for the Pareto-optimal solution on the coordination roadmap. The motion safety concept and a computation method for the motion safety based on the local paths of the robots has been presented. Simulation results have been provided for the two robots system and three robots system, and the planning results with and without considering safety cost have been compared. Experimental results for two Pioneer3-DX robots have been presented, and the comparison with the results using the front-avoid mechanism has been illustrated.

The algorithms proposed in this work have the following two characteristics. First, we do not need to explicitly construct the boundaries of the collision regions in coordination space. Second, sampling based algorithms computed in the coordination space are introduced to reduce the computational complexities. The proposed approach has been applied to coordinate the robots traveling on the pre-specified

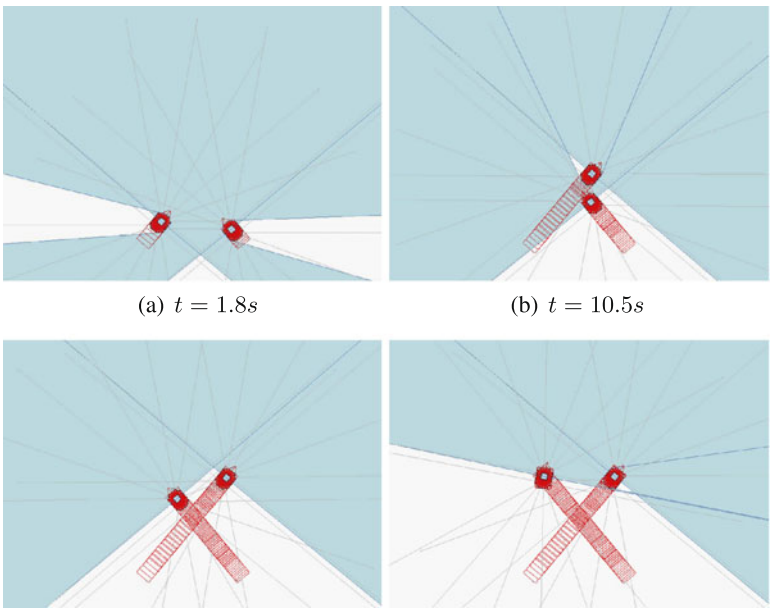


(a) Pareto-optimal solutions ignoring safety cost (*green line*) and considering both costs (*red line*) in the coordination space.

(b) Velocity schedule results for A_1 (*solid*) and A_2 (*dash-dot*).

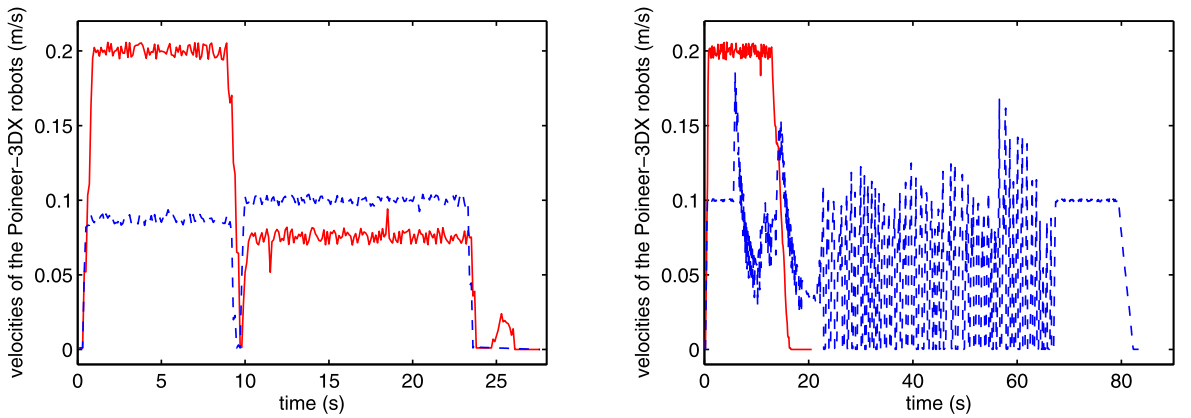
Fig. 17 (Color online) Pareto-optimal solution for two Pioneer-3DX robots

Fig. 18 Snapshots of experimental results of two robots using the proposed methods



(a) $t = 1.8s$ (b) $t = 10.5s$

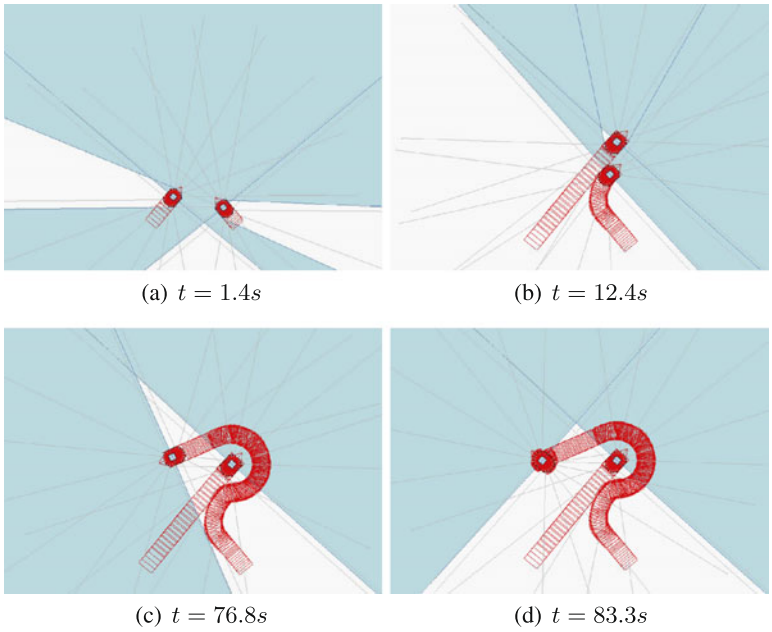
(c) $t = 25.4s$ (d) $t = 27.6s$



(a) Actual velocities using the pre-planned trajectories.

(b) Actual velocities using the front-avoid mechanism.

Fig. 19 Velocities of A_1 (*solid*) and A_2 (*dash-dot*) in the experiment



(a) $t = 1.4s$ (b) $t = 12.4s$

(c) $t = 76.8s$ (d) $t = 83.3s$

Fig. 20 Snapshots of the experimental results using front-avoid mechanism

paths. For the application to general cases using the proposed algorithm, it is possible to reserve more than one valid paths in the individual path planning stage, then search for the Pareto-optimal results in the coordination space taking into account all these paths.

Acknowledgements The authors would like to thank the editor and anonymous reviewers for their constructive comments that helped improve the quality and presentation of this paper. They acknowledge Dr. Bernard How Voon Ee from the National University of Singapore for his help with the language proofreading of the manuscript. This work was supported by the Fundamental Research Program of Northwestern Polytechnical University (NPU) under Grant JC2010-0225, and partly supported by the National Science Foundation of China under Grant 60875071.

References

Barraquand, J., & Latombe, J. (1991). Robot motion planning: a distributed representation approach. *Int. J. Robot. Res.*, *10*(6), 628–649.

Chakravarthy, A., & Ghose, D. (1998). Obstacle avoidance in a dynamic environment: a collision cone approach. *IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum.*, *28*(5), 562–574.

Chitsaz, H., O'Kane, J. M., & LaValle, S. M. (2004). Exact Pareto-optimal coordination of two translating polygonal robots on an acyclic roadmap. In *Proceedings of 2004 IEEE international conference on robotics and automation* (Vol. 4, pp. 3981–3986).

Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., & Thrun, S. (2005). *Principles of robot motion: theory, algorithms and implementation*. Cambridge: MIT Press.

Cui, R., Ge, S. S., How, V. E. B., & Choo, Y. S. (2010). Leader-Follower formation control of underactuated autonomous underwater vehicles. *Ocean Eng.*, *37*(17–18), 1491–1502.

Durfee, E. H. (1999). Distributed continual planning for unmanned ground vehicle teams. *AI Mag.*, *20*(4), 55–61.

Fua, C. H., Ge, S. S., Do, K. D., & Lim, K. W. (2007). Multirobot formations based on the queue-formation scheme with limited communication. *IEEE Trans. Robot.*, *23*(6), 1160–1169.

Ge, S. S., & Cui, Y. J. (2002a). New potential functions for mobile robot path planning. *IEEE Trans. Robot. Autom.*, *16*(5), 615–620.

Ge, S. S., & Cui, Y. J. (2002b). Dynamic motion planning for mobile robots using potential field method. *Auton. Robots*, *13*(3), 207–222.

Ge, S. S., & Fua, C. H. (2004). Queues and artificial potential trenches for multirobot formations. *IEEE Trans. Robot.*, *21*(4), 646–656.

Ghrist, R., O'Kane, J. M., & LaValle, S. M. (2005a). Computing Pareto optimal coordinations on roadmaps. *Int. J. Robot. Res.*, *24*(11), 997–1010.

Ghrist, R., O'Kane, J. M., & LaValle, S. M. (2005b). Pareto optimal coordination on roadmaps. In *Algorithmic foundations of robotics VI* (pp. 171–186).

Hsu, D. (2000). *Randomized single-query motion planning in expansive spaces*. Ph.D. dissertation, Stanford University.

Hu, H., Brady, M., & Probert, P. (2002). Coping with uncertainty in control and planning for a mobile robot. In *Proceedings of IEEE/RSJ international workshop on intelligent robots and systems* (pp. 1025–1030).

Hu, J., Prandini, M., & Sastry, S. (2002). Optimal coordinated maneuvers for three-dimensional aircraft conflict resolution. *J. Guid. Control Dyn.*, *25*(5), 888–900.

Kant, K., & Zucker, S. (1986). Toward efficient trajectory planning: the path velocity decomposition. *Int. J. Robot. Res.*, *5*(3), 72–89.

Kavraki, L. E., Svestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, *12*(4), 566–580.

Kuchar, J. K., & Yang, L. C. (2002). A review of conflict detection and resolution modeling methods. *IEEE Trans. Intell. Transp. Syst.*, *1*(4), 179–189.

Kuffner, J. (2004). Effective sampling and distance metrics for 3d rigid body path planning. In *Proceedings of IEEE international conference on robotics and automation* (Vol. 4, pp. 3993–3998).

Kuffner, J. J., & LaValle, S. M. (2000). Rrt-connect: an efficient approach to single-query path planning. In *Proceedings of IEEE international conference on robotics and automation* (Vol. 2, pp. 995–1001).

Ladd, A., & Kavraki, L. E. (2002). Generalizing the analysis of PRM. In *Proceedings of IEEE international conference on robotics and automation* (Vol. 2, pp. 2120–2125).

Laffary, M. (2002). MobileRobots advanced robotics interface for applications (ARIA) developer's reference manual. http://robots.mobilerobots.com/wiki/ARIA.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.

LaValle, S. M., & Hinrichsen, J. (2001). Visibility-based pursuit-evasion: the case of curved environments. *IEEE Trans. Robot. Autom.*, *17*(2), 196–202.

Liu, G., & Li, Z. (2002). A unified geometric approach to modeling and control of constrained mechanical systems. *IEEE Trans. Robot. Autom.*, *18*(4), 574–587.

Mataric, M. (1995). Issues and approaches in the design of collective autonomous agents. *Robot. Auton. Syst.*, *16*(2), 321–332.

Mortezaie, F. (1991). *Constrained voronoi diagram and its application to autonomous mobile robot path planning*. Ph.D. dissertation, University of California, Irvine.

Parker, L. E. (1997). 'Cooperative motion control for multi-target observation. In *Proceedings of the 1997 IEEE/RSJ international conference on intelligent robots and systems* (Vol. 3, pp. 1591–1597).

Plaku, E., & Kavraki, L. (2005). Distributed sampling-based roadmap of trees for large-scale motion planning. In *IEEE international conference on robotics and automation* (Vol. 4, pp. 3868–3873).

Purwin, O., D'Andrea, R., & Lee, J. W. (2008). Theory and implementation of path planning by negotiation for decentralized agents. *Robot. Auton. Syst.*, *56*(5), 422–436.

Schouwenaars, T., How, J., & Feron, E. (2004). Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees. In *Proceedings of the AIAA guidance, navigation and control conference*.

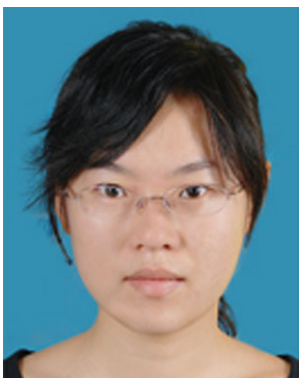
Schwarzer, F., Saha, M., & Latombe, J. C. (2003). Exact collision checking of robot paths. In *Algorithmic foundations of robotics V* (pp. 25–42).

Shanmugavel, M., Tsourdos, A., White, B., & Zbikowski, R. (2009). Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs. *Control Eng. Pract.*, *18*(9), 1084–1092.

Shin, K. G., & Zheng, Q. (1992). Minimum-time collision free trajectory planning for dual robot systems. *IEEE Trans. Robot. Autom.*, *8*(5), 641–644.

Rongxin Cui received the B.Eng. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2003, 2008, respectively. From August 2008 to August 2010, he worked as Research Fellow at the Centre for Offshore Research & Engineering, National University of Singapore, Singapore. He has been with the College of Marine Engineering, Northwestern Polytechnical University, Xi'an, China, since September 2010, where he is currently an Associate Professor. He has been serving as the program committee member of the 5th IEEE CIS&RAM 2011 conference, and the ARMS@AAMAS2011 workshop. He is a regular reviewer for around ten journals in the area of control, robotics, and ocean engineering. His current research interests are control of nonlinear systems, cooperative path planning and control for multiple robots, control and navigation for underwater vehicles, and system development.

Bo Gao received the B.Eng. degree and Master degree from Northwestern Polytechnical University, Xi'an, China, in 2004, 2007, respectively, where he is currently working toward the Ph.D. degree at the College of Marine Engineering. His research interests are in the area of autonomous multirobot collaboration and planning.



Ji Guo received the B.Eng. degree and Master degree from Northwestern Polytechnical University, Xi'an, China, in 2005, 2008, respectively. She is now working as a lecture at the College of Physics and Electrical Engineering, Anyang Normal University, Anyang, China. Her research interests are in the area of robot path planning.