International Journal of Control, Automation and Systems 17(10) (2019) 2610-2623
http://dx.doi.org/10.1007/s12555-018-0176-9
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

PSO-based Minimum-time Motion Planning for Multiple Vehicles Under Acceleration and Velocity Limitations

Anugrah K. Pamosoaji, Mingxu Piao, and Keum-Shik Hong*

Abstract: This paper discusses a particle swarm optimization (PSO)-based motion-planning algorithm in a multiple-vehicle system that minimizes the traveling time of the slowest vehicle by considering, as constraints, the radial and tangential accelerations and maximum linear velocities of all vehicles. A class of continuous-curvature three-degree Bezier curves are selected as the basic shape of the vehicle trajectories to minimize the number of parameters required to express them mathematically. In addition, velocity profile generation using the local minimum of the radial-accelerated linear velocity profile, which reduces the calculation effort, is introduced. A new PSO-based search algorithm, called "particle-group-based PSO," is introduced to find the best combination of trajectories that minimizes the traveling time of the slowest vehicle. A particle group is designed to wrap up a set of particles representing each vehicle. The first and last two control points characterizing a curve are used as the state vector of a particle. Simulation results demonstrating the performance of the proposed method are presented. The main advantage of the proposed method is its minimization of the velocity-profile-generation time, and thereby, its maximization of the search time.

Keywords: Bezier curves, motion planning, multiple-vehicle systems, particle swarm optimization.

1. INTRODUCTION

Collision-free and minimum-time motion planning for multiple vehicles has been studied over several decades. Various solutions, most utilizing geometrical approaches, have been proposed, where significant results based on path analysis theories have been reported [1–11]. These approaches analyzed path planning as a means of connecting two configurations (position and orientation) using spline curve parameters. Issues of path optimality have been investigated as well, for instance, geometric shape-based approach [7–12], sampling-based approach [12], and potential field approach [13–15]. The application of acceleration limits to the generation of tangential velocity along the spline curves as a planned path was considered in several studies [3, 7, 8]. However, most studies addressed only the path generation issue, and not the optimization of the vehicle traveling time.

Research on motion planning is naturally motivated by real-world situations wherein the workspaces are usually irregular and dynamic. Some approaches using control design were revealed in [14–21, 21–27]. The problem of point-to-point control has been addressed in various works: The navigation control approaches proposed in [14–27] rigorously guaranteed that the vehicles would reach their final configurations. To deal with the changes in vehicles' internal parameters, some adaptive control designs [18–22] as well as adaptive control/neural network combinations were introduced [23–26]. Implementation of these approaches probably will depend on the accuracies of new types of sensors, such as visual sensors [24], new control techniques, and brain-signal-based controls [25–27].

In the case of multiple vehicles, the issue of time optimality has been extensively reported [28–32]. Some path generation studies include a potential-based algorithm combined with a genetic algorithm (GA) [28] and a sequential planning algorithm [29–32]. In such reports, the complexity of environments was considered.

However, even though these control approaches have succeeded in simple adversarial and less dynamic workspaces, the derivations of robust navigation functions under more complex workspaces have commonly been proved difficult. Motion-planning paradigms having arisen as a new class of solution, the navigation problem of vehicles has now become that of tracking a given motion plan [16, 20].

Manuscript received March 20, 2018; revised July 23, 2018 and November 15, 2018; accepted December 21, 2018. Recommended by Associate Editor Augie Widyotriatmo. This work was supported by the National Research Foundation of Korea under the auspices of the Ministry of Science and ICT, Republic of Korea (grant no. NRF-2017R1A2A1A17069430).

Anugrah K. Pamosoaji is with the Faculty of Industrial Technology, Universitas Atma Jaya Yogyakarta; Jl. Babarsari no. 44, Yogyakarta 55281, Indonesia (e-mail: kusumo_pamosoaji@mail.uajy.ac.id). Mingxu Piao and Keum-Shik Hong are with the School of Mechanical Engineering, Pusan National University; 2 Busandaehak-ro, Geumjeong-gu, Busan 46241, Korea (e-mails: {piaomx, kshong}@pusan.ac.kr).
* Corresponding author.

Optimal motion planning for multiple-vehicle systems sometimes is unable to find solutions using traditional optimality principles. Various approaches, such as those noted in [28,33,34], involve heuristic-based methods such as GAs, though in their study simplification of velocity generation was not the focus. Notwithstanding, multiple-vehicle systems require reduction in computation effort in velocity generation, as the search process takes additional time to obtain the optimal solution.

This study addresses the problem of slowest-vehicle traveling-time minimization in a multiple-vehicle system, considering the maximum linear velocity and tangential and radial accelerations along with the linear velocity of each vehicle. This issue is important for various applications, such as warehouse operations, as in the real world, the ability to generate minimum-time schedules is necessary for the resolution of crucial problems, such as bottlenecks. To the best of our knowledge, most multiple-vehicle-planning studies have treated path- and velocity-planning problems separately. Even though [28–34] produced convincing results for multiple-vehicle motion planning, they did not consider velocity and acceleration limitations. For instance, Sharma *et al*. [34], having investigated the relation between vehicle size and traveling time of the slowest vehicle, reported significant traveling-time results for a multiple- and omnidirectional-vehicle system in a bounded workspace. However, the results lack tangential acceleration constraints on the vehicles, which is not realistic.

In this study, we apply particle swarm optimization (PSO) to find trajectories whereby the slowest vehicle's traveling time is minimal. We use a class of spline curves, called three-degree Bezier curves, as the basic path shape. Several works [3, 7–9] demonstrated a good performance using such type of curves [3, 7–9]. Some considerations of using such curves include easiness to determine the tangentiality at each point on the path [3], flexibility in reshaping using at least one control point [7]], and trajectory smoothness [8, 9, 11]. Mostly, the path-planning issue in robotics considers the advantage of smoothness [3,7–9,11]. Various optimization studies have employed this search method [33,35–40], which uses particles whose motions are influenced by the set of fittest particles. Some major issues regarding this approach lie in the areas of premature convergence, many local minima, and search effort. Since the introduction of PSO by Clerc and Kennedy [35], it has undergone various modifications. For instance, Cai and Yang [33] reported the use of PSO in concert with potential field functions for cooperative target-searching tasks within the multiple-vehicle realm. Latest results in adaptive PSO were introduced in [36–40].

The contribution of this study is its presentation of a two-step method for generating path and trajectory and finding the minimum traveling time of the slowest vehicle in the multiple-vehicle environment. The first step is the generation of paths (which is highly possible to have intersection points to each other) with a linear velocity profile accommodating the maximum allowable radial and tangential acceleration constraints on vehicles (Subsections 3.1 and 3.2) as well as preventing inter-vehicle collision. We propose a velocity-profile-generation algorithm modified from [3]. This algorithm, to reduce the calculation effort, utilizes path segments to plot the tangential velocity for the path. The second step in the proposed method is minimization of the traveling time of the slowest vehicle through a novel particle-group-based PSO method (Section 4). Using this method, a multi-dimensional search space is split into elementary smaller-dimensional ones. Thereby, each elementary particle of every single particle group can, using different cognitive and social random multipliers, determine the best path for its corresponding vehicle. The proposed method offers a significant advantage of producing a collision-free path together with a time-minimizing velocity profile under maximum linear velocity and tangential- and radial-acceleration limitations.

Another study discussing the same objective is that of Xidias and Azariadis [28]. Here, a group of vehicles must visit all provided stations from their corresponding depots. The number of vehicles allowable to visit a station is no more than one. The combination of the GA and a bump function (to mark the obstacles) is used for the optimization. As shown in this study, compared to the results in [30], our velocity profile generation algorithm is more realistic, as it includes tangential acceleration.

The rest of this paper is organized as follows: Section 2 describes the problem of slowest-vehicle traveling-time minimization in a multiple-vehicle system. Section 3 discusses the motion-planning algorithm employed for linear velocity generation based on the local minimum of the radial-accelerated linear velocity profile. Section 4 introduces the particle-group-based PSO algorithm. Section 5 presents performance simulation results for the proposed method. Finally, Section 6 draws conclusions and discusses future research directions.

2. PROBLEM DESCRIPTION

Consider a group of N_v nonholonomic vehicles. Each vehicle is represented by its configuration (position and orientation), denoted as $(x_i, y_i, \theta_i)^T \in \Re^2 \times (-\pi, \pi]$, $i = 1$, ..., N_v in the planar workspace, as shown in Fig. 1. In this study, we specify the vehicles as those under a kinematic constraint, described as $\dot{x}_i \sin\theta_i = \dot{y}_i \cos\theta_i$. Each vehicle moves from its initial configuration $(x_i^0, y_i^0, \theta_i^0) \in \Re^2 \times (-\pi, \pi]$ to its goal configuration $(x_i^g, y_i^g, \theta_i^g) \in \Re^2 \times (-\pi, \pi]$. The i-th vehicle follows its corresponding planned path $\mathbf{P}_i = \{\mathbf{p}_{i,k}\}$, $k = 1$, ..., N_{sp}, where N_{sp} is the number of sampled points of each vehicle's path, $\mathbf{P}_{i,k} =$

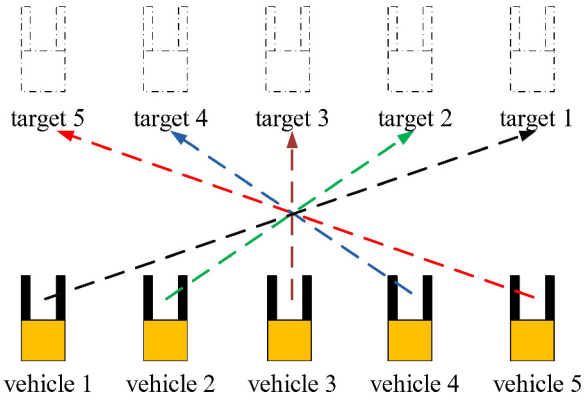

Fig. 1. Multiple-vehicle motion-planning scenario.

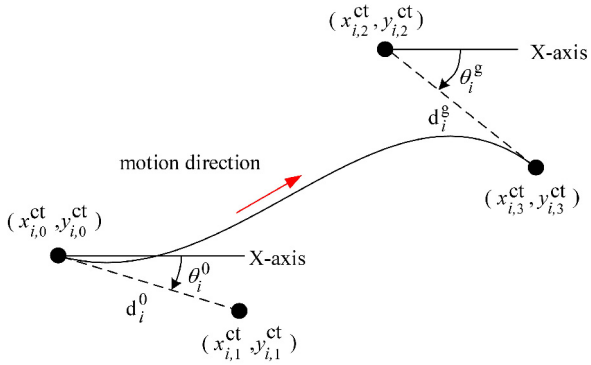

Fig. 2. Three-degree Bezier curve.

$(x_{i,k}, y_{i,k})^{\mathrm{T}} \in \Re^{2\times 1}$ is the k-th sample of $\mathbf{P}_i$ and is parameterized by $\xi_{i,k} = k/N_{\mathrm{sp}} \in [0,1]$. We assume that all vehicles have the same number of sampled points, i.e., N_{sp}.

In this study, we intend to represent $\mathbf{p}_{i,k}$ as a three-degree Bezier curve. To obtain such a curve, we use four control points $(x^{\mathrm{ct}}_{i,h}, y^{\mathrm{ct}}_{i,h})^{\mathrm{T}}$, where $h = 0, ..., 3$, as input.

Remark 1: Some properties of the three-degree Bezier curve employed in this study are described as follows: i) The Bezier curve's shape is invariant under the translation and rotation of the control points $(x^{\mathrm{ct}}_{i,h}, y^{\mathrm{ct}}_{i,h})^{\mathrm{T}}$. By this property, the flexibility of the curve's shape can be guaranteed [7–9]. ii) The control points with $h = 0$ and $h = 3$ are the start and end points of the curve, respectively. The other control points ($h = 1$ and $h = 2$) are used to determine the global shape of the curve [3]. By using such control points, we can represent $\mathbf{p}_{i,k}$ as a point in the three-degree Bezier curve characterized by the control points, as shown in Fig. 2. Furthermore, let us define the following i-th constants obtained by the control points:

$$(\alpha_{i,0}, \beta_{i,0})^{\mathrm{T}} = (x^{\mathrm{ct}}_{i,0}, y^{\mathrm{ct}}_{i,0})^{T}, \tag{1}$$

$$(\alpha_{i,1}, \beta_{i,1})^{T} = (-3x^{\mathrm{ct}}_{i,0} + 3x^{\mathrm{ct}}_{i,1}, -3y^{\mathrm{ct}}_{i,0} + 3y^{\mathrm{ct}}_{i,1})^{T}, \tag{2}$$

$$(\alpha_{i,2}, \beta_{i,2})^{T} = (-3x^{\mathrm{ct}}_{i,0} - 6x^{\mathrm{ct}}_{i,1} + 3x^{\mathrm{ct}}_{i,2}, -3y^{\mathrm{ct}}_{i,0} - 6y^{\mathrm{ct}}_{i,1} + 3y^{\mathrm{ct}}_{i,2})^{T}, \tag{3}$$

$$(\alpha_{i,3}, \beta_{i,3})^{T} = (-x^{\mathrm{ct}}_{i,0} + 3x^{\mathrm{ct}}_{i,1} - 3x^{\mathrm{ct}}_{i,2} + x^{\mathrm{ct}}_{i,3}, -y^{\mathrm{ct}}_{i,0} + 3y^{\mathrm{ct}}_{i,1} - 3y^{\mathrm{ct}}_{i,2} + y^{\mathrm{ct}}_{i,3})^{T}. \tag{4}$$

In addition, we define the following vectors:

$$\mathbf{\Lambda}_{i,k} = [\ \alpha_{i,0}\ \ \alpha_{i,1}\ \ \alpha_{i,2}\ \ \alpha_{i,3}\ ] \in \Re^{1\times 4}, \tag{5}$$

$$\mathbf{B}_{i,k} = [\ \beta_{i,0}\ \ \beta_{i,1}\ \ \beta_{i,2}\ \ \beta_{i,3}\ ] \in \Re^{1\times 4}. \tag{6}$$

We define a vector of parameters as follows:

$$\xi_{i,k} = [\ 1\ \ \xi_{i,k}\ \ \xi^2_{i,k}\ \ \xi^3_{i,k}\ ]^{\mathrm{T}} \in \Re^{4\times 1}. \tag{7}$$

By using (1)-(7), we formulate the path $\mathbf{p}_{i,k}$ as a three-degree Bezier curve as follows:

$$\mathbf{p}_{i,k} = [\ \mathbf{\Lambda}_{i,k}\ \ \mathbf{B}_{i,k}\ ]^{\mathrm{T}} \boldsymbol{\xi}_{i,k} \in \Re^{2\times 1}. \tag{8}$$

Moreover, the control points with $h = 1$ and $h = 2$ can be determined as follows:

$$y^{\mathrm{ct}}_{i,1} = (x^{\mathrm{ct}}_{i,1} - x^{\mathrm{ct}}_{i,0}) \tan\theta^0_i + y^{\mathrm{ct}}_{i,0}, \tag{9}$$

$$y^{\mathrm{ct}}_{i,2} = (x^{\mathrm{ct}}_{i,3} - x^{\mathrm{ct}}_{i,2}) \tan\theta^g_i + y^{\mathrm{ct}}_{i,3}. \tag{10}$$

Because the points with $h = 0$ and $h = 3$ are unchangeable (as a consequence of constant initial and goal configurations), we regard the remaining two control points as variables. Consequently, we now have four unknowns in a path: $x^{\mathrm{ct}}_{i,1}$, $y^{\mathrm{ct}}_{i,1}$, $x^{\mathrm{ct}}_{i,2}$, and $y^{\mathrm{ct}}_{i,2}$. However, the number of unknowns can be reduced to two by defining distances $d^0_i \in \Re^+$ and $d^g_i \in \Re^+$ as follows:

$$d^0_i = \|[x^{\mathrm{ct}}_{i,1}\ \ y^{\mathrm{ct}}_{i,1}]^{\mathrm{T}} - [x^{\mathrm{ct}}_{i,0}\ \ y^{\mathrm{ct}}_{i,0}]^{\mathrm{T}}\| \in \Re^+, \tag{11}$$

$$d^g_i = \|[x^{\mathrm{ct}}_{i,3}\ \ y^{\mathrm{ct}}_{i,3}]^{\mathrm{T}} - [x^{\mathrm{ct}}_{i,2}\ \ y^{\mathrm{ct}}_{i,2}]^{\mathrm{T}}\| \in \Re^+. \tag{12}$$

We can incorporate them into the following equations:

$$(x^{\mathrm{ct}}_{i,1}, y^{\mathrm{ct}}_{i,1})^{\mathrm{T}} = (x^{\mathrm{ct}}_{i,0} + d^0_i \cos\theta^0_i, y^{\mathrm{ct}}_{i,0} + d^0_i \sin\theta^0_i)^{\mathrm{T}}, \tag{13}$$

$$(x^{\mathrm{ct}}_{i,2}, y^{\mathrm{ct}}_{i,2})^{\mathrm{T}} = (x^{\mathrm{ct}}_{i,3} - d^g_i \cos\theta^g_i, y^{\mathrm{ct}}_{i,0} - d^g_i \sin\theta^g_i)^{\mathrm{T}}. \tag{14}$$

Remark 2: From Remark 1, compared to straigth line-arc-based curves, the use of Bezier curves is more efficient. In the line-arc-based curves, we need to modify each elementary curve to change the global shape of the curve, while in the Bezier curves, we only need two control points to alter. In addition, a disadvantage exists since the changing of some control points leads to overall reshaping of the curves. However, since we have no constraint about the curves' shape, this disadvantage can be disregarded in this paper.

Now, let t, t^0_i, and $t^g_i \in \Re^+ \cup \{0\}$ be defined as the time, start time, and completion time of a mission of the i-th vehicle, respectively. Furthermore, let $T_i = t^g_i - t^0_i \in \Re^+ \cup \{0\}$ be the overall traveling time of the i-th vehicle from $\mathbf{p}_{i,0}$ to $\mathbf{p}_{i,N_{\mathrm{sp}}}$ and let $\rho_{i,j}(t) \in \Re^+ \cup \{0\}$ and $\rho^{\min}_{i,j}(t) \in \Re^+ \cup \{0\}$ be the distance and minimum allowable distance between the i-th and j-th vehicles, respectively. Furthermore, let $\mathbf{a}^t_i \in \Re^2$ be the vector of tangential

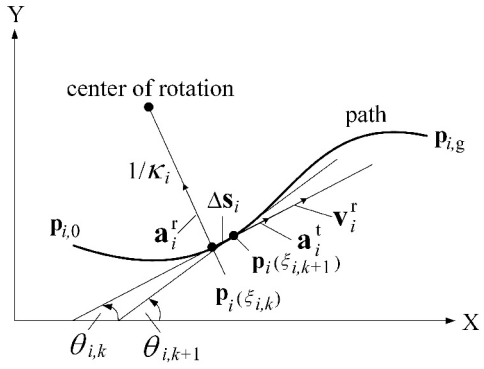

Fig. 3. Radial and tangential accelerations.

acceleration, $a_i^{\mathrm{t,max}} \in \Re^+ \cup \{0\}$ be the maximum allowable magnitude of $\mathbf{a}_i^{\mathrm{t}}$, $\mathbf{a}_i^{\mathrm{r}} \in \Re^2$ be the vector of current radial acceleration, and $a_i^{r,\max} \in \Re^+ \cup \{0\}$ be the maximum allowable magnitude of $\mathbf{a}_i^{\mathrm{r}}$.

Owing to the presence of $\mathbf{a}_i^{\mathrm{t}}$ and $\mathbf{a}_i^{\mathrm{r}}$, as shown in Fig. 3, the linear velocity of the vehicle, denoted by $\mathbf{v}_i \in \Re^2$, can be expressed as a function of either time t or path parameter ξ_i. As a function of time, $\mathbf{v}_i$ can be expressed as

$$\mathbf{v}_i(t) = \mathbf{v}_i(t_i^0) + \int_0^t \mathbf{a}_i^{\mathrm{t}}(\vartheta)d\vartheta. \tag{15}$$

On the other hand, as a function of path parameter ξ_i, $\mathbf{v}_i$ can be expressed as

$$\|\mathbf{v}_i(\xi_i)\| = \left(\frac{\|\mathbf{a}_i^{\mathrm{r}}\|}{\mathrm{sign}(\kappa_i(\xi_i))\kappa_i(\xi_i)}\right)^{1/2}, \tag{16}$$

where $\kappa_i \in \Re$ is the curvature at a point parameterized by ξ_i, and is formulated as

$$\begin{aligned}\kappa_i(\xi_i) =& (x'_i(\xi_i)y''_i(\xi_i) - y'_i(\xi_i)x''_i(\xi_i)) \\ & \times ((x'_i)^2 + (y'_i)^2)^{-3/2}.\end{aligned} \tag{17}$$

The prime and double-prime notations in (17) represent the first and second derivatives with respect to ξ_i, respectively. In addition, according to [1], the positive and negative signs of κ_i correspond to counter-clockwise and clockwise directions of $\mathbf{v}_i$, respectively.

Substitution of the first and second ξ_i-derivatives of x_i and y_i into (17) yields

$$\begin{aligned}\kappa_i(\xi_i) =& (2w_1 + 6w_2\xi_i + 6w_3\xi_i^2) \\ & \times (w_4 + 4w_5\xi_i + (6w_6 + 4w_7)\xi_i^2 \\ & + 12w_8\xi_i^3 + 9w_9\xi_i^4)^{-3/2},\end{aligned} \tag{18}$$

where

$$w_1 = \alpha_{1,i}\beta_{2,i} - \alpha_{2,i}\beta_{1,i}, \tag{19}$$
$$w_2 = \alpha_{1,i}\beta_{3,i} - \alpha_{3,i}\beta_{1,i}, \tag{20}$$
$$w_3 = \alpha_{2,i}\beta_{3,i} - \alpha_{3,i}\beta_{2,i}, \tag{21}$$
$$w_4 = \alpha_{1,i}^2 + \beta_{1,i}^2, \tag{22}$$
$$w_5 = \alpha_{1,i}\alpha_{2,i} + \beta_{1,i}\beta_{2,i}, \tag{23}$$
$$w_6 = \alpha_{1,i}\alpha_{3,i} + \beta_{1,i}\beta_{3,i}, \tag{24}$$
$$w_7 = \alpha_{2,i}^2 + \beta_{2,i}^2, \tag{25}$$
$$w_8 = \alpha_{2,i}\alpha_{3,i} + \beta_{2,i}\beta_{3,i}, \tag{26}$$
$$w_9 = \alpha_{3,i}^2 + \beta_{3,i}^2. \tag{27}$$

Remark 3: The application of $\|\mathbf{v}_i(\xi_i)\|$ in (16) as a constraint is performed under the consideration that the vehicles' turning should not generate an excessive centripetal force.

Therefore, in this study, we apply a maximum allowable radial acceleration $a_i^{\mathrm{r,max}}$. Let $v_i^0 = \|\mathbf{v}_i(t_i^0)\| \in \Re$ and $v_i^g = \|\mathbf{v}_i(t_i^g)\| \in \Re$ be the initial and final linear velocities of the i-th vehicle, respectively. Let $J \in \Re^+$ be the maximum traveling time among the N_{v} vehicles, formulated as

$$J = \max(T_1, T_2, ..., T_{N_{\mathrm{v}}}). \tag{28}$$

The problem can be stated as follows: Consider the initial configuration $(x_i^0, y_i^0, \theta_i^0)^{\mathrm{T}}$ and goal configuration $(x_i^g, y_i^g, \theta_i^g)^{\mathrm{T}}$ of a group of N_{v} vehicles. Generate trajectories (linear velocity profiles) for each vehicle such that J is minimized subject to the following acceleration limitation

$$0 \le \|\mathbf{a}_i^{\mathrm{t}}\| \le \|\mathbf{a}_i^{\mathrm{t,max}}\|, \tag{29}$$

linear velocity limitations

$$0 \le \|\mathbf{v}_i\| \le \left(\frac{\|\mathbf{a}_i^{\mathrm{r,max}}\|}{\mathrm{sign}(\kappa_i(\xi_i))\kappa_i(\xi_i)}\right)^{1/2}, \tag{30}$$

$$0 \le \|\mathbf{v}_i\| \le \left\|\mathbf{v}_i(t_i^0) + \int_0^t \mathbf{a}_i^{\mathrm{t,max}}(\vartheta)d\vartheta\right\|, \tag{31}$$

and boundary conditions

$$v_i^0 = 0 \ \text{ and } \ v_i^g = 0, \tag{32}$$

for all $t \in (t_i^0, t_i^g)$. In addition, the generated velocity profiles guarantee that the vehicle-to-vehicle distances for any pair of i-th and j-th vehicles satisfy

$$\rho_{i\sim j} \ge \rho_{i\sim j}^{\min}. \tag{33}$$

To simplify the problem, we assume that for each vehicle, the only obstacles are the other vehicles.

3. MOTION-PLANNING ALGORITHM

For computational reason, i.e., calculating the linear velocity $\mathbf{v}_i$ in both expressions of (16)-(17), we introduce the terms "tangential-accelerated velocity profiles" (TVPs), "radial-accelerated velocity profiles" (RVPs), and "goal-related velocity profiles" (GVPs).

Definition 1: A linear velocity profile $\mathbf{v}_i$ expressed as a time function (see (15)) is defined as a TVP and is symbolized as $\mathbf{v}_i^{\mathrm{t}} \in \Re^{2\times 1}$.

Definition 2: A linear velocity profile v_i expressed as a path-parameter function (see (16)) is defined as an RVP and is symbolized as $\mathbf{v}_i^{\mathrm{r}} \in \Re^2$.

Definition 3: A TVP that starts from any point $\mathbf{p}_i \in \mathbf{P}_i$ and ends at the goal point $(x_i^g, y_i^g, \theta_i^g)$ is defined as a GVP.

Remark 4: Since the results of the proposed method are linear velocity profiles, the RVPs must be converted into appropriate linear velocity profiles obeying the constraints (29)-(32).

Remark 5: It is highly possible that at point $\mathbf{p}_i$, the applied linear velocity might violate one of the allowable boundaries of constraints (30)-(31) if we apply $\|\mathbf{a}_i^{\mathrm{r}}\| = a_i^{\mathrm{r,max}}$ and $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$, simultaneously. Therefore, as shown later in this section, we only use one type of velocity profile at any point $\mathbf{p}_i \in \mathbf{P}_i$.

We define the boundaries of a TVP as $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}} \in \mathbf{P}_i$ (start point) and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}} \in \mathbf{P}_i$ (end point), where $n_{\mathrm{TVP}} \in \{0, 1, 2, ...\}$ is the index of TVP. For calculating T_i, we divide the path into NSP sampling points and apply the following steps: Generate the path-parameter-based velocity profile (Subsection 3.1), determine the boundaries of TVP (Subsection 3.2), and generate the time-based velocity profile (Subsection 3.3). As a preliminary step, we prepare some propositions and theorems describing necessary properties of time-based linear velocity profiles.

Let us define L_i as the total length of the i-th vehicle's path. As the path is curvilinear, we apply linear interpolation to approximate the value of L_i as

$$L_i = \sum_{k=1}^{N_{\mathrm{SP}}} (L_{i,k} - L_{i,k-1}) + \varepsilon_L, \tag{34}$$

where

$$L_{i,k} = \|\mathbf{p}_{i,k} - \mathbf{p}_{i,0}\| \tag{35}$$

is the length of the k-th segment of the i-th path bounded by $\mathbf{p}_{i,0}$ and $\mathbf{p}_{i,k}$, and ε_L is the approximation error (which is assumed to be sufficiently small). We state Theorems 1 and 2 as the basic theorem for determining tangential acceleration to reduce the traveling time T_i. Note that we use notation "$\cdot$" to represent the dot product operator.

Theorem 1: Suppose that the vehicle moves forward from $\mathbf{p}_i^{\mathrm{m}} \in \mathbf{P}_i$ to $\mathbf{p}_i^{\mathrm{n}} \in \mathbf{P}_i$ with tangential velocity $\mathbf{v}_i^{\mathrm{t}}$, where the initial velocity is $\mathbf{v}_i^{\mathrm{t}} = \mathbf{v}_i^{\mathrm{t,m}}$ and the final velocity is $\mathbf{v}_i^{\mathrm{t}} = \mathbf{v}_i^{\mathrm{t,n}}$, $\|\mathbf{v}_i^{\mathrm{t,m}}\| < \|\mathbf{v}_i^{\mathrm{t,n}}\|$, along the path connecting such points. In addition, suppose that the tangential acceleration $\mathbf{a}_i^{\mathrm{t,m\sim n}}$ with a monotonic fashion $\mathbf{a}_i^{\mathrm{t,m\sim n}} \cdot \mathbf{v}_i^{\mathrm{t}} = \|\mathbf{a}_i^{\mathrm{t,m\sim n}}\|\|\mathbf{v}_i^{\mathrm{t}}\|$ (accelerated motion) is applied along the path. Then, a large $\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\|$ leads to a short traveling time T_i.

Proof: Without loss of generality, assume that $\mathbf{a}_i^{\mathrm{t,m\sim n}}$ is constant. Let $t^{\mathrm{m\sim n}}$ be the traveling time from $\mathbf{p}_i^{\mathrm{m}}$ to $\mathbf{p}_i^{\mathrm{n}}$. Then,

$$L_i^{\mathrm{m\sim n}} \approx \|\mathbf{v}_i^t t^{m\sim n} + 0.5\mathbf{a}_i^{\mathrm{t,m\sim n}} (t^{m\sim n})^2\|. \tag{36}$$

Since $L_i^{\mathrm{m\sim n}}$ is a constant, it can be proved that $t^{\mathrm{m\sim n}}$ is shorter if $\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\|$ increases. □

Theorem 2: Suppose that the vehicle moves forward from $\mathbf{p}_i^{\mathrm{m}}$ to $\mathbf{p}_i^{\mathrm{n}} \in \mathbf{P}_i$ with tangential velocity $\mathbf{v}_i^{\mathrm{t}}$, where the initial velocity is $\mathbf{v}_i^{\mathrm{t}} = \mathbf{v}_i^{\mathrm{t,m}}$ and the final velocity is $\mathbf{v}_i^{\mathrm{t}} = \mathbf{v}_i^{\mathrm{t,n}}$, $\|\mathbf{v}_i^{\mathrm{t,m}}\| > \|\mathbf{v}_i^{\mathrm{t,n}}\|$, along the path connecting such points. A tangential acceleration $\mathbf{a}_i^{\mathrm{t,m\sim n}}$ with a monotonic fashion $\mathbf{a}_i^{\mathrm{t,m\sim n}} \cdot \mathbf{v}_i^{\mathrm{t}} = -\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\|\|\mathbf{v}_i^{\mathrm{t}}\|$ (decelerated motion) is applied along the path. Then, a larger $\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\|$ leads to a shorter traveling time T_i.

Proof: Without loss of generality, we assume that $\mathbf{a}_i^{\mathrm{t,m\sim n}}$ is constant. It is straighforward that

$$\mathbf{v}_i^m = \mathbf{v}_i^n - \mathbf{a}_i^{\mathrm{t,m\sim n}} t_i^{\mathrm{m\sim n}}, \tag{37}$$

$$L_i^{m\sim n} = \|\mathbf{v}_i^m t^{\mathrm{m\sim n}} + 0.5\mathbf{a}_i^{\mathrm{t,m\sim n}} (t^{m\sim n})^2\|. \tag{38}$$

Then, by substituting $\mathbf{v}_i^n$ into (37) to (38), we obtain

$$L_i^{m\sim n} = \|\mathbf{v}_i^n t^{m\sim n} - 0.5\mathbf{a}_i^{\mathrm{t,m\sim n}} (t^{m\sim n})^2\|. \tag{39}$$

Since $L_i^{\mathrm{m\sim n}}$ is constant and $\mathbf{a}_i^{\mathrm{m\sim n}}$ is in the opposite direction to $\mathbf{v}_i^{\mathrm{t}}$, it is proved that a larger $\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\|$ leads to a shorter $t^{\mathrm{m\sim n}}$. □

3.1. Path-parameter-based velocity profile generation

Suppose that the set of control points of a Bezier path (i.e., $x_{i,1}^{\mathrm{ct}}$, $y_{i,1}^{\mathrm{ct}}$, $x_{i,2}^{\mathrm{ct}}$, and $y_{i,2}^{\mathrm{ct}}$) are given. This step aims to generate the maximum allowable radial linear velocity, denoted by $v_{i,k}^{r,max} \in \Re$, $k \in [1, N_{\mathrm{sp}}]$, for each $\mathbf{p}_{i,k}$. We symbolize points at which $\|\mathbf{v}_i\| = v_{i,k}^{\mathrm{r,max}}$, $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$, and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} > 0$ as $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ and those at which $\|\mathbf{v}\| = v_{i,k}^{\mathrm{r,max}}$, $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$, and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} = -\|\mathbf{a}_i^{\mathrm{t}}\|\|\mathbf{v}_i^{\mathrm{t}}\|$ as $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, as shown in Fig. 4. The use of these points will be explained in Subsection 3.2. The procedures for generating parameter-based velocity profiles are provided in Algorithm 1 below:

Algorithm 1 (Generate path)**:**

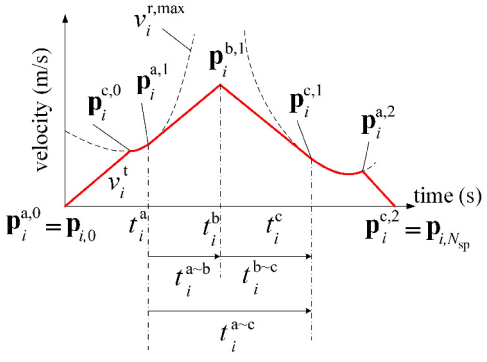

Fig. 4. Example of generated linear velocity profiles.

1) Input: $x_{i,0}^{\mathrm{ct}} = x_i^0$, $y_{i,0}^{\mathrm{ct}} = y_i^0$, $x_{i,3}^{\mathrm{ct}} = x_i^{\mathrm{g}}$, $y_{i,3}^{\mathrm{ct}} = y_i^{\mathrm{g}}$, $x_{i,1}^{\mathrm{ct}}$, $y_{i,1}^{\mathrm{ct}}$, $x_{i,2}^{\mathrm{ct}}$, and $y_{i,2}^{\mathrm{ct}}$.
2) **For** each segment $k \in [1, N_{\mathrm{sp}}]$ on the i-th path,
3) Determine $\mathbf{p}_{i,k}$ using (1)-(8);
4) Calculate $\kappa_{i,k+1}$ using (18).
5) Determine $v_{i,k}^{\mathrm{r,max}}$ by applying (16) with $a_i^{\mathrm{r}} = a_i^{\mathrm{r,max}}$.
6) Determine $L_{i,k}$ using (35).
7) **End**

Proposition 1: A path connecting (x_i^0, y_i^0) and $(x_i^{\mathrm{g}}, y_i^{\mathrm{g}})$ can be performed by applying Algorithm 1.

Proof: Considering the definition of control points in Remark 1 and the expression of Bezier curves in (1)-(8), the proposition is proved. □

3.2. Determination of TVP boundaries

As $\mathbf{v}_i^{\mathrm{t}}$ is a function of time and its magnitude is prevented from exceeding $v_i^{\mathrm{r,max}}$ at all points on $\mathbf{P}_i$, it is difficult to calculate it directly at a single point $\mathbf{p}_i$. Therefore, evaluation must be performed by applying time-based scanning from $\mathbf{p}_i^0$ to $\mathbf{p}_i^{\mathrm{g}}$. On the other hand, under constant $a_i^{\mathrm{r,max}}$, $v_i^{\mathrm{r,max}}$ on a point depends only on the point's curvature κ_i, which is the function of ξ_i. A solution to this difficulty, presented in [8], entails a separate calculation of both velocity profiles starting from each stationary point (the point of local maximum curvature) of the path and taking the minimum values. However, this technique, since the velocity profiles are generated twice (i.e., both TVP and RVP are generated and their minimum values are determined at each sampled point of the final velocity profile), is time-consuming. Additionally, the algorithm is inefficient for multiple-vehicle trajectory planning. Thus, we develop a technique such that the evaluation of $v_i^{\mathrm{t,max}}$ and $v_i^{\mathrm{r,max}}$ can be synchronized through time-based scanning, with no repetitive velocity profile calculation.

Now, let us define $t_i^{\mathrm{a\sim b}}$ as the traveling time from $\mathbf{P}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ to an intermediate point $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$, $t_i^{\mathrm{b\sim c}}$ as the traveling time from $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ to $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, and $t_i^{\mathrm{a\sim c}}$ as the traveling time from $\mathbf{P}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ to $\mathbf{P}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, as shown in Fig. 4. The position of point $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ is unknown and should be determined such that the application of a_i^{t} satisfies constraint (29). Let ξ_i^{a}, ξ_i^{b}, and ξ_i^{c} be the path parameters at $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$, $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$, and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, respectively. To minimize $t_i^{\mathrm{a\sim c}}$, the velocity $\mathbf{v}_i(\xi_i^{\mathrm{b}}) \in \Re^2$ applied in $\mathbf{P}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ must be determined. Therefore, let $L_i^{\mathrm{a\sim b}}$, $L_i^{\mathrm{b\sim c}}$, and $L_i^{\mathrm{a\sim c}}$ be defined as the lengths of the paths connecting $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$, $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, and $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, respectively. It is straightforward that

$$L_i^{\mathrm{a\sim c}} = L_i^{\mathrm{a\sim b}} + L_i^{\mathrm{b\sim c}}. \tag{40}$$

We express the equation for the path length from $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ to $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ symbolized as $L_i^{\mathrm{a\sim b}}$, as

$$L_i^{\mathrm{a\sim b}} = \left\| \mathbf{v}_i(\xi_i^{\mathrm{a}}) t_i^{\mathrm{a\sim b}} + 0.5 \mathbf{a}_i^{\mathrm{t,a\sim b}} (t_i^{\mathrm{a\sim b}})^2 \right\|, \tag{41}$$

where $\mathbf{a}_i^{\mathrm{t,a\sim b}}$ is the tangential acceleration applied to the motion from $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ to $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$. It is straightforward that $\mathbf{v}_i(\xi_i^{\mathrm{b}})$ and $t_i^{\mathrm{b\sim c}}$ have the following relation.

$$\mathbf{v}_i(\xi_i^{\mathrm{b}}) = \mathbf{v}_i(\xi_i^{\mathrm{a}}) + \mathbf{a}_i^{\mathrm{t,a\sim b}} t_i^{\mathrm{a\sim b}}. \tag{42}$$

Furthermore, $\mathbf{v}_i(\xi_i^{\mathrm{c}})$ and $\mathbf{a}_i^{\mathrm{b\sim c}}$ have the relation

$$\mathbf{v}_i(\xi_i^{\mathrm{c}}) = \mathbf{v}_i(\xi_i^{\mathrm{b}}) + \mathbf{a}_i^{\mathrm{t,b\sim c}} t_i^{\mathrm{b\sim c}}, \tag{43}$$

where $\mathbf{a}_i^{\mathrm{b\sim c\ c}}$ is the tangential acceleration applied to the motion from $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ to $\mathbf{P}_i^{\mathrm{c},n_{\mathrm{TVP}}}$. The substitution of (42) into (43) yields

$$t_i^{\mathrm{b\sim c}} = \left\| \mathbf{v}_i(\xi_i^{\mathrm{c}}) - \mathbf{v}_i(\xi_i^{\mathrm{a}}) - \mathbf{a}_i^{\mathrm{t,a\sim b}} t_i^{\mathrm{a\sim b}} \right\| \Big/ \left\| \mathbf{a}_i^{\mathrm{t,b\sim c}} \right\|. \tag{44}$$

Similarly, we can express $L_i^{\mathrm{b\sim c}}$ as

$$L_i^{\mathrm{b\sim c}} = \left\| \mathbf{v}_i(\xi_i^{\mathrm{b}}) t_i^{\mathrm{b\sim c}} + 0.5 \mathbf{a}_i^{\mathrm{t,b\sim c}} (t_i^{\mathrm{b\sim c}})^2 \right\|. \tag{45}$$

We state the following theorem as the fundamental principle for constructing propositions of tangential velocity profile generation.

Proposition 2: Let $\|\mathbf{v}_i\| \le v_i^{\mathrm{t,max}}$ satisfying constraints (29)-(33) be applied to points in a TVP between $\mathbf{p}_i^{\mathrm{m}}$ and $\mathbf{p}_i^{\mathrm{n}}$ of the i-th path, where $\|\mathbf{v}_i(\xi_i^{\mathrm{m}})\| < \|\mathbf{v}_i(\xi_i^{\mathrm{n}})\|$, as shown in Fig. 4. The application of $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t,m\sim n}} \cdot \mathbf{v}_i^{\mathrm{t}} = \|\mathbf{a}_i^{\mathrm{t,m\sim n}}\| \|\mathbf{v}_i^{\mathrm{t}}\|$ between both points minimizes the traveling time $t_i^{\mathrm{m\sim n}}$ without violating (29)-(33).

Proof: This proposition is a particular case of Theorem 1 with $\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\| = a_i^{\mathrm{t,max}}$. □

Proposition 3: Let $\|\mathbf{v}_i\| \le v_i^{\mathrm{t,max}}$ be applied to the points in TVP between $\mathbf{p}_i^{\mathrm{m}}$ and $\mathbf{a}_i^{\mathrm{n}}$ of the i-th path, where $\|\mathbf{v}_i(\xi_i^{\mathrm{m}})\| < \|\mathbf{v}_i(\xi_i^{\mathrm{n}})\|$, as shown in Fig. 4. The application of $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} = -\|\mathbf{a}_i^{\mathrm{t}}\| \|\mathbf{v}_i^{\mathrm{t}}\|$ between both points minimizes the traveling time $t_i^{\mathrm{m\sim n}}$ without violating (29)-(33).

Proof: This proposition is a particular case of Theorem 2 with $\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\| = a_i^{\mathrm{t,max}}$. □

According to Propositions 2 and 3, the linear velocity profile generated between two points $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$ with predetermined initial and final linear velocities, i.e., $\mathbf{v}_i(\xi_i^{\mathrm{a}})$ and $\mathbf{v}_i(\xi_i^{\mathrm{c}})$, respectively, will reach the global minimum $t_i^{\mathrm{a\sim c}}$ if the following steps are applied consecutively:

1) Apply $\|\mathbf{a}_i^{\mathrm{t,a\sim b}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t,m\sim n}} \cdot \mathbf{v}_i^{\mathrm{t}} = \|\mathbf{a}_i^{\mathrm{t,m\sim n}}\| \|\mathbf{v}_i^{\mathrm{t}}\|\}$ to the vehicle from $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ to a midpoint $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$.

2) Apply $\|\mathbf{a}_i^{\mathrm{t,b\sim c}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t,m\sim n}} \cdot \mathbf{v}_i^{\mathrm{t}} = -\|\mathbf{a}_i^{\mathrm{t,m\sim n}}\| \|\mathbf{v}_i^{\mathrm{t}}\|$ from $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ to $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$.

The traveling time from $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ to $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$ (i.e., $t_i^{\mathrm{a\sim c}}$) is derived in the following steps. We have the expression $L_i^{\mathrm{a\sim c}}$, according to (40)-(42) and (45), as follows:

$$\begin{aligned} L_i^{\mathrm{a\sim c}} = & \left\| \mathbf{v}_i(\xi_i^{\mathrm{a}}) t_i^{\mathrm{a\sim b}} \right\| + 0.5 a_i^{\mathrm{t,max}} (t_i^{\mathrm{a\sim b}})^2 \\ & + \left(\|\mathbf{v}_i(\xi_i^{\mathrm{a}})\| + a_i^{\mathrm{t,max}} t_i^{\mathrm{a\sim b}} \right) t_i^{\mathrm{b\sim c}} \\ & - 0.5 a_i^{\mathrm{t,max}} (t_i^{\mathrm{b\sim c}})^2. \end{aligned} \tag{46}$$

Applying $\|\mathbf{a}_i^{\mathrm{t,b\sim c}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t,b\sim c}} \cdot \mathbf{v}_i^{\mathrm{t}} < 0$ to (44) and substituting the resulting $t_i^{\mathrm{b\sim c}}$ into (46) yields

$$\begin{aligned}&\left\|\mathbf{v}_i(\xi_i^{\mathrm{a}})t_i^{\mathrm{a\sim b}}\right\| + 0.5a_i^{\mathrm{t,max}}(t_i^{\mathrm{a\sim b}})^2 + (0.5/a_i^{\mathrm{t,max}}) \\ &\quad\times\left(\|\mathbf{v}_i(\xi_i^{\mathrm{a}})\| + a_i^{\mathrm{t,max}}t^{a\sim b}\right)^2 - \|\mathbf{v}_i(\xi_i^{\mathrm{c}})\|^2 - L_i^{\mathrm{a\sim c}} \\ &= 0. \end{aligned} \tag{47}$$

The solution to (47) is the formulation of $t_i^{\mathrm{a\sim b}}$; that is,

$$t_i^{\mathrm{a\sim b}} = 0.5\left(-\gamma_{1,i} + ((\gamma_{1,i})^2 - 4\gamma_{2,i})^{0.5}\right), \tag{48}$$

where

$$\gamma_{1,i} = -2(\|\mathbf{v}_i(\xi_i^{\mathrm{c}})\| + \|\mathbf{v}_i(\xi_i^{\mathrm{a}})\|)/a_i^{\mathrm{t,max}}, \tag{49}$$

$$\gamma_{2,i} = \frac{3\left(\|\mathbf{v}_i(\xi_i^{\mathrm{a}})\|^2 - 4\|\mathbf{v}_i(\xi_i^{\mathrm{a}})\|\,\|\mathbf{v}_i(\xi_i^{\mathrm{c}})\| + \|\mathbf{v}_i(\xi_i^{\mathrm{c}})\|^2\right)}{(4(a_i^{\mathrm{t,max}})^2) + L_i^{\mathrm{a\sim b}}/a_i^{\mathrm{t,max}}}. \tag{50}$$

We can determine the position of $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ by its length to $\mathbf{p}_i^0$; that is,

$$\begin{aligned}L_i^{\mathrm{a\sim b}} =& 0.25(\gamma_{1,i})^2 a_i^{\mathrm{t,a\sim b}} - 0.5\gamma_{1,i}\|\mathbf{v}_i(\xi_i^{\mathrm{a}})\| - \gamma_{2,i} \\ &+ \left\|0.5\mathbf{v}_i(\xi_i^{\mathrm{a}}) + 0.25\mathbf{a}_i^{\mathrm{t,a\sim b}}\gamma_{1,i}\right\|\left((\gamma_{1,i})^2 - 4\gamma_{2,i}\right)^{0.5}. \end{aligned} \tag{51}$$

The mechanism for generating TVP boundaries is shown in Algorithm 2.

Algorithm 2 (Determine the TVP and RVP boundaries)**:**

1) Input: $\mathbf{p}_{i,k}$, $\kappa_{i,k+1}$, $v_{i,k}^{\mathrm{r,max}}$, and $L_{i,k}$, for all $k \in [1, N_{\mathrm{sp}}]$.

2) **For** each segment $\mathbf{p}_{i,k}$, $k \in [1, N_{\mathrm{sp}}]$, calculate the tangential acceleration

$$a_i^{\mathrm{d}} = \left((v_{i,k}^{\mathrm{r,max}})^2 - (v_{i,k-1}^{\mathrm{r,max}})^2\right)/(2(L_{i,k} - L_{i,k-1})).$$

3) If $a_i^{\mathrm{d}} > a_i^{\mathrm{t,max}}$, then $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}} = \mathbf{p}_{i,k-1}$.

4) If $a_{i,k}^{\mathrm{t}} > -a_i^{\mathrm{t,max}}$, then $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}} = \mathbf{p}_{i,k-1}$.

5) **End**

3.3. Time-based velocity profile

As shown in the previous subsections, we deal with path-parameter-based and time-based velocity-profile-generation mechanisms. The time-based velocity profile generation steps are described in Algorithms 3-6. As shown in step 3 of Algorithm 3, there are some alternatives of velocity profile generation: TVP (Algorithm 4), RVP (Algorithm 5), or GVP (Algorithm 6). Clearly, the principle used in generating TVP and GVP velocity profiles comes from Propositions 2 and 3 (implicitly, Theorems 1 and 2, respectively). For instance, in steps 2a and 2b of Algorithm 4, the generated velocity profiles yield the minimum traveling time for the accelerated and decelerated motions, respectively. In the generation of RVP, the tangential acceleration used is not $a_i^{\mathrm{t,max}}$, since it will violate (30). However, it will still comply with Theorems 1 and 2, since the applied tangential acceleration is as maximum as possible such that constraints (29)-(31) are not violated.

Algorithm 3 (Generate the velocity profile)**:**

1) Input $\mathbf{p}_{i,k}$, $\kappa_{i,k+1}$, $v_{i,k}^{\mathrm{r,max}}$, and $L_{i,k}$ for all $k \in [1, N_{\mathrm{sp}}]$.

2) For each i-th vehicle, initiate $T_i = 0$.

3) For each segment between $\mathbf{p}_{i,k}$ and $\mathbf{p}_{i,k+1}$, $k \in [1, N_{\mathrm{sp}}]$, perform the following steps.

a) Evaluate the parts' segment based on time sampling $\Delta t_{i,h} = T^{\mathrm{s}}$. If the tangential acceleration needed to reach the goal is $a_{i,h}^{\mathrm{t}} = -a_i^{\mathrm{t,max}}$, then it can be concluded that the velocity profile is GVP. Consequently, execute Algorithm 6. If not, execute Algorithm 4. Afterward, continue to step d).

b) If Algorithm 4 is executed and gives a result that the velocity profile is RVP, then execute Algorithm 5. Subsequently, continue to step d).

c) If the result of Algorithm 4 is that the velocity profile is TVP, then continue to step d).

d) Calculate the traveling time from the initial segment to the current segment:

$$T_i = T_{i-1} + \Delta t_{i,h}.$$

e) Continue to the next segment.

Algorithm 4 (Generate TVP)**:** For a segment between $\mathbf{p}_{i,k}$ and $\mathbf{p}_{i,k+1}$, perform the following steps.

1) Check whether there exists an intermediate point $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ between $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$.

2)**If** $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ exists,

a) if the evaluated segment is located between $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$, then apply $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} > 0$.

b) If the evaluated segment is located between $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ and $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$, apply $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} < 0$.

c) Otherwise, set the velocity profile as RVP.

3) **If** $\mathbf{p}_i^{\mathrm{b},n_{\mathrm{TVP}}}$ does not exist, then

a) apply $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} > 0$ if $v_i^{\mathrm{r,max}}$ at $\mathbf{p}_i^{\mathrm{a},n_{\mathrm{TVP}}}$ is greater than $v_i^{\mathrm{r,max}}$ at $\mathbf{p}_i^{\mathrm{c},n_{\mathrm{TVP}}}$.

b) apply $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} < 0$ otherwise.

4) Return to Algorithm 3.

Algorithm 5 (Generate RVP)**:** For a segment between $\mathbf{p}_{i,k}$ and $\mathbf{p}_{i,k+1}$, perform the following step:

1) Determine all points on the evaluated segment such that the tangential velocity magnitude follows $v_i^{\mathrm{r,max}}$ and the tangential acceleration magnitude needed by the vehicle to travel from a point to the consecutive point in $\Delta t_i = T^{\mathrm{s}}$ does not exceed $a_i^{\mathrm{t,max}}$.

2) Calculate the total traveling time through RVP. The formulation of the traveling time is similar to (44) and (48)-(50) with a_i^{t} might not being equal to $a_i^{\mathrm{t,max}}$.

3) Calculate the elapsed time from the beginning of the RVP segment to the recent time:

$$T_i^{\mathrm{RVP}} = \sum_{i=1}^{\text{current segment}} \Delta t_i.$$

4) Return to Algorithm 3.

Algorithm 6 (Generate GVP)**:** For a segment between $\mathbf{p}_{i,k}$ and $\mathbf{p}_{i,k+1}$, perform the following step:

1) Determine all points on the evaluated segment such that by applying $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ and $\mathbf{a}_i^{\mathrm{t}} \cdot \mathbf{v}_i^{\mathrm{t}} < 0$, the vehicle can travel point-to-point in $\Delta t_i = T^{\mathrm{s}}$.

2) Calculate the elapsed time from the beginning of the GVP segment to the recent time.

3) Return to Algorithm 3.

As the vehicle-to-vehicle distances need to be verified to obey (33), we apply a time-based distance verification method. Suppose that the i-th vehicle initially occupies the k-th point $\mathbf{p}_{i,k}$ of its corresponding path and moves with tangential acceleration $a_i^{\mathrm{t,max}}$. Let T^{s} be the default sampling time. At the next time interval $\Delta t_i = T^{\mathrm{s}}$, the occupied point is possibly not $\mathbf{p}_{i,k+1}$: In fact, it is highly possible that the next occupation point is between $\mathbf{p}_{i,k}$ and $\mathbf{p}_{i,k+1}$ or between $\mathbf{p}_{i,k+1}$ and $\mathbf{p}_{i,k+2}$. To address this issue, we define $\mathbf{v}_i^{\mathrm{t}} = \{\mathbf{v}_{i,h}^{\mathrm{t}}\}$, $\mathbf{a}_i^{\mathrm{t}} = \{\mathbf{a}_{i,h}^{\mathrm{t}}\}$, $\mathbf{L}_i = \{\mathbf{L}_{i,h}\}$, and $\mathbf{\Delta t}_i = \{\Delta t_{i,h}\}$ (where $h \in [1, N_{\mathrm{tsp}}]$) as sets of linear velocity, tangential acceleration, distance from (x_i^0, y_i^0) along the i-th path, and set of elapsed times of the i-th vehicle, respectively. N_{tsp} is the maximum number of time-based samples from (x_i^0, y_i^0) until (x_i^g, y_i^g). This approach leads to the possibility that the number of time-based samples for each vehicle is different.

Let $\mathbf{p}_{i,h+1}$ be a point occupied after moving from $\mathbf{p}_{i,h}$ at initial velocity $\mathbf{v}_{i,h}^{\mathrm{t}}$ and applying $\|\mathbf{a}_i^{\mathrm{t}}\| = a_i^{\mathrm{t,max}}$ within the time interval Δt_i, as shown in Fig. 5. Therefore, the location of $\mathbf{p}_{i,h+1}$ can be expressed as

$$\mathbf{p}_{i,h+1} = \mathbf{p}_{i,h} + \mathbf{v}_{i,h}^{\mathrm{t}} \Delta t_i + 0.5 \mathbf{a}_{i,h}^{\mathrm{t}} (\Delta t_i)^2. \tag{52}$$

Assume that $\mathbf{p}_{i,h+1}$ is located in a line bounded by $\mathbf{p}_{i,k}$ and $\mathbf{p}_{i,k+1}$; in other words, $\mathbf{p}_{i,h+1}$ is defined as

$$\mathbf{p}_{i,h+1} = \lambda \mathbf{p}_{i,k+1} + (1-\lambda)\mathbf{p}_{i,k}. \tag{53}$$

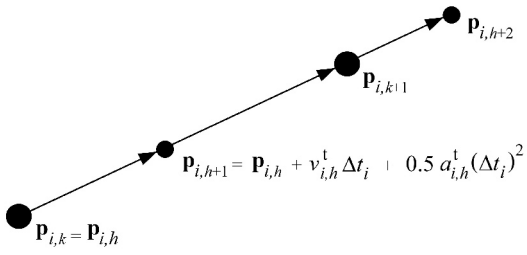

Fig. 5. Positions after the application of time-based velocity v_i^{t} within $\Delta t_i = T^{\mathrm{s}}$.

There are two possibilities for the value of λ. First, $\lambda \in (0, 1]$. In this case, the default sampling time $\Delta t_i = T^{\mathrm{s}}$ needs no revision. The second case is $\lambda \in (0, +\infty]$. Here, Δt_i must be revised so that $\mathbf{p}_{i,h+1} = \mathbf{p}_{i,k}$ (Fig. 5).

4. PARTICLE-GROUP-BASED PSO SEARCH ALGORITHM

The problem defined in Section 2 is a combination of planning and scheduling. The work is not only planning paths for all vehicles but also scheduling the exact times of each vehicle to pass through the resulting interception points such that any single collision to other vehicles can be prevented. As shown in [28], this type of problem can be categorized as NP-hard, which needs large computational effort if we use a conventional method such as dynamic programming. Therefore, meta-heuristic techniques are typically considered to solve such types of problems.

In the present study, we utilize a population-based PSO search algorithm to search the collision-free and minimum-time motion plans. The main rationale for this technique is the fact that the gradient of the cost function J is not required to find the global minimum; simply, this cost function is evaluated by reference only to the particle positions.

We design a PSO-based planner as shown in Fig. 6, which consists of three parts: path generator, velocity profile generator, and optimizer. The path generator executes the parameter-based path construction as shown in Algorithm 1. The velocity profile generator performs velocity profile generation as shown in Algorithms 2-6. The last

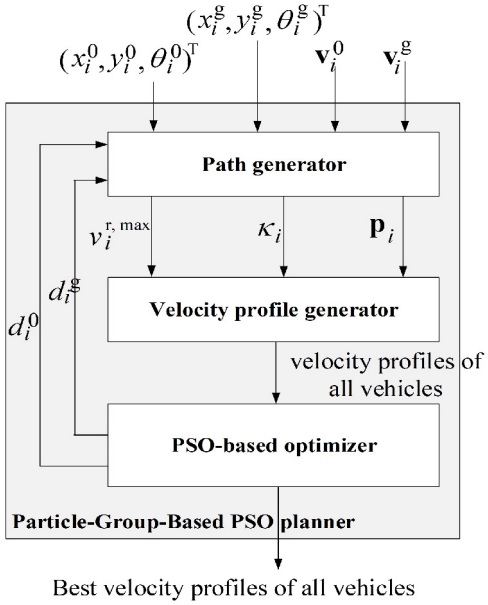

Fig. 6. PSO-based planner.

process is an optimizer that executes the searching mechanism described in Algorithm 7.

In this planner, we apply a new type of PSO algorithm, named "particle-group-based PSO," which uses groups of particles called "particle groups" rather than single ones. In principle, a particle group is designated as a wrapped set of particles representing each vehicle. A particle group $\mathbf{Q}_{k,l}$ is denoted by

$$\mathbf{Q}_{k,l} = [\ \mathbf{q}_{1,k,l}\ \ \mathbf{q}_{2,k,l}\ \ \cdots\ \ \mathbf{q}_{N_\mathrm{v},k,l}\ ]^\mathrm{T}, \tag{54}$$

where $\mathbf{q}_{i,k,l} = [\ d^0_{i,k,l}\ \ d^\mathrm{g}_{i,k,l}\ ]$, $i \in [1, N_\mathrm{v}]$, $k \in [1, N_\mathrm{iter}]$, and $l \in [1, N_\mathrm{pg}]$ indicate the particles of the i-th vehicle's trajectory parameter vector $[\ d^0_{i,k,l}\ \ d^\mathrm{g}_{i,k,l}\ ]^\mathrm{T}$ in the l-th group evaluated at the k-th iteration. $N_\mathrm{iter} > 0$ and $N_\mathrm{pg} > 0$ are the numbers of iterations and particle groups, respectively. The distances $d^0_{i,k,l}$ and $d^\mathrm{g}_{i,k,l}$ are defined as in (11)-(12) but with additional indexes k and l indicating the k-th iteration and l-th particle group.

For the l-th iteration, each particle of the particle group, i.e., $\mathbf{q}_{i,k+1,l}$, corresponds to $T_{i,k,l}$ (the time T_i that is calculated at the k-th iteration for the l-th particle group), which is calculated in Algorithm 3.

We define the locally best l-th particle group $\mathbf{Q}^\mathrm{lbest}_{k,l}$ at the k-th iteration as

$$\begin{aligned}\mathbf{Q}^\mathrm{lbest}_{k,l} =& [\ \mathbf{q}^\mathrm{lbest}_{1,k,l}\ \ \mathbf{q}^\mathrm{lbest}_{2,k,l}\ \ \cdots\ \ \mathbf{q}^\mathrm{lbest}_{N_\mathrm{v},k,l}\ ]^\mathrm{T} \\ =& \mathbf{Q}^\mathrm{lbest}_{k,l}\Big|_{J=\min\left(\max_i\{T_{i,k,l}\},\max_i\{T_{i,k-1,l}\},\ldots,\max_i\{T_{i,1,l}\}\right)}.\end{aligned} \tag{55}$$

In addition, we define the globally best particle group $\mathbf{Q}^\mathrm{gbest}$ until the l-th iteration as

$$\begin{aligned}\mathbf{Q}^\mathrm{qbest} =& [\ \mathbf{q}^\mathrm{gbest}_{1,k}\ \ \mathbf{q}^\mathrm{gbest}_{2,k}\ \ \cdots\ \ \mathbf{q}^\mathrm{qbest}_{N_\mathrm{v},k}\ ]^\mathrm{T} \\ =& \mathbf{Q}_{k,l}\Big|_{J=\min\left(\max_i\{T_{i,k,l}\},\max_i\{T_{i,k,l-1}\},\ldots,\max_i\{T_{i,k,1}\}\right)}.\end{aligned} \tag{56}$$

The resulting trajectories with minimum time, or $\min J$, are those performed by $\mathbf{Q}^\mathrm{gbest}$ for all iterations $k \in [1, N_\mathrm{iter}]$ and all particle groups $l \in [1, N_\mathrm{pg}]$. In other words, the traveling time of the slowest vehicle in the $\mathbf{Q}^\mathrm{gbest}$-generated trajectories after all iterations $k \in [1, N_{iter}]$ is the shortest traveling time among other trajectories in the search space.

Let us define a search space, denoted as $D_i \in \Re^+ \times \Re^+$, for the i-th vehicle's trajectory. The algorithm proceeds in three steps. First, the particles in all groups (i.e., $\mathbf{q}_{i,k,l}$) explore D_i to find the parameter vector $\mathbf{q}_{i,k,l} \in D_i$ such that the i-th trajectory has the minimum traveling time T_i. The updating rule $\Delta\mathbf{q}_{i,k+1,l}$ is defined as

$$\begin{aligned}\Delta\mathbf{q}_{i,k+1,l} =& c^\mathrm{in}_i \Delta\mathbf{q}_{i,k,l} + c^\mathrm{cg}_i r^\mathrm{cg}_i (\mathbf{q}^\mathrm{lbest}_{i,k,l} - \mathbf{q}_{i,k,l}) \\ &+ c^\mathrm{sc}_i r^\mathrm{sc}_i (\mathbf{q}^\mathrm{gbest}_{i,k} - \mathbf{q}_{i,k,l}),\end{aligned} \tag{57}$$

where $\mathbf{q}^\mathrm{best}_{i,k,l}$ represents the best position achieved by the i-th particle in the l-th group until the k-th iteration; $\mathbf{q}^\mathrm{best}_{i,k}$ represents the best parameter vector $\mathbf{q}_{i,k,l}$ among all groups $l \in [1, N_\mathrm{pg}]$ at the k-th iteration; $\Delta\mathbf{q}_{i,k,l}$ is the changing rate of vector $\mathbf{q}_{i,k,l}$ at the k-th iteration; c^in_i, c^cg_i, and c^sc_i are constant inertial, cognitive, and social scaling factors, respectively; and $c^\mathrm{cg}_i \in [0,1]$ and $c^\mathrm{sc}_i \in [0,1]$ are cognitive and social random multipliers. Therefore, the next position of $\mathbf{q}_{i,k+1,l}$ is expressed as

$$\mathbf{q}_{i,k+1,l} = \mathbf{q}_{i,k,l} + \Delta\mathbf{q}_{i,k+1,l}. \tag{58}$$

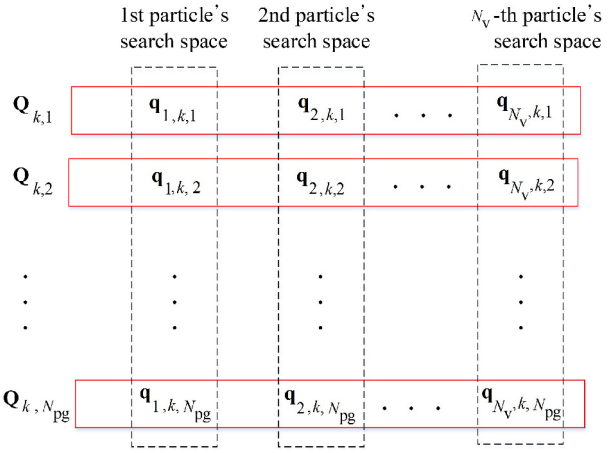

Fig. 7. Each particle group provides particles whose types represent respective vehicles.

As shown in Fig. 7, at the k-th iteration, each particle group $\mathbf{Q}_{k,l}$ consists of N_v types of particles, where the i-th particle represents the i-th vehicle with its own search space. As shown in Algorithm 7, each iteration entails the search for appropriate $\mathbf{q}_{i,k,l}$ for all vehicles. Second, each particle group performs a search for the collective maximum traveling time among N_v particles. The third and final step is to find $\min J$.

This method simplifies the number of iterations; that is, only a 2-dimensional search space, rather than a $2N_\mathrm{v}$-dimensional space for each particle representing all vehicles, is needed. Using this method, the cognitive term of each particle, or in other words, the second term of (57), is based on the best-experienced position of the represented vehicle. This is different from the situation wherein a particle is in a $2N_\mathrm{v}$-dimensional space. In this case, the cognitive term is based on the best-experienced position of all vehicles, which is not realistic, as each vehicle naturally remembers its experience, not the remaining vehicles' experiences.

Now, define t^el_i and $t^{\mathrm{el},0}_i$ as the general time frame and the lower bound of the partial time frame, respectively, used for collision-checking purposes in line 3d of Algorithm 7. For each time frame $[t^{\mathrm{el},0}_i, t^{\mathrm{el},0}_i + \Delta t_i]$, all $\mathbf{q}_{i,k,l}$ are tested to determine if they satisfy (24) or not. If a collision is detected between any two pairs of $\mathbf{q}_{i,k,l}$ and $\mathbf{q}_{j,k,l}$ and $i \neq j$, we set a high $\Delta T_{i,l}$.

Two issues in PSO-based searching are convergence and stability. The term "convergence" relates to the driving of a particle into a minimum local state, while "stability" refers to the prevention of an explosion of particle velocity. In the proposed particle-group-based PSO, the convergence and stability proofs follow [35].

Proposition 4: The following condition guarantees the convergence and stability of particle-group-based PSO.

$$c_i^{\mathrm{cg}} r_i^{\mathrm{cg}} + c_i^{\mathrm{sc}} r_i^{\mathrm{sc}} < 4 c_i^{\mathrm{in}}. \tag{59}$$

Proof: Convergence and stability in this study are determined using the steps used in [35]. Let $\varphi_1 = c_i^{\mathrm{cg}} r_i^{\mathrm{cg}} / c_i^{\mathrm{in}}$, $\varphi_2 = c_i^{\mathrm{sc}} r_i^{\mathrm{sc}} / c_i^{\mathrm{in}}$, $\varphi = \varphi_1 + \varphi_2$, and χ be a constriction coefficient formulated as

$$\chi = \begin{cases} \left(\dfrac{2\psi}{\varphi - 2 + (\varphi^2 - 4\varphi)^{0.5}} \right)^{0.5}, & \text{if } \varphi > 4, \\ \psi^{0.5}, & \text{otherwise}, \end{cases} \tag{60}$$

where $\psi \in (0,1)$. The particle velocity expression in (57) can be rewritten as

$$\begin{aligned} \Delta \mathbf{q}_{i,k+1,l} =& \chi c_i^{\mathrm{in}} (\Delta \mathbf{q}_{i,k,l} + \varphi_1 (\mathbf{q}_{i,k,l}^{\mathrm{best}} - \mathbf{q}_{i,k,l})) \\ &+ \varphi_2 (\mathbf{q}_{i,k}^{\mathrm{best}} - \mathbf{q}_{i,k,l}). \end{aligned} \tag{61}$$

Expression (61) is similar to constriction type 1 in [35]. According to this constriction type, the requirement for convergence and stability is $\varphi \in (0,4)$; thus (59) must be met. □

Algorithm 7 (Particle-Group-Based PSO)**:**

1) **For** all iterations, $k \in [1, N_{\mathrm{iter}}]$,
2) **For** all particle groups, $l \in [1, N_{\mathrm{pg}}]$
3) **For** all vehicles, $i \in [1, N_{\mathrm{v}}]$ in each particle, and calculate $\mathbf{q}_{i,k,l}$ by executing the following steps:
   a) Generate the path-parameter-based velocities of all vehicles using Algorithm 1.
   b) Calculate TVP and RVP boundaries using Algorithm 2.
   c) Apply Algorithm 3 to generate velocity profiles and calculate the traveling time $T_{i,j}$ of each vehicle.
   d) Check the vehicle-to-vehicle distance $\rho_{i\sim j}$. If $\rho_{i\sim j} < \rho_{i\sim j}^{\max}$ (the i-th and j-th vehicles collide), then the current particle group is not used in the searching process anymore. Therefore, the evaluation continues to the next particle group. Otherwise, the evaluation continues to the next vehicle.
4) **End**
5) **End**
6) Find locally best particle groups $\mathbf{Q}_l^{\mathrm{lbest}}$ by using (57)-(58).
7) **End**
8) Find the globally best particle group $\mathbf{Q}^{\mathrm{gbest}}$.

5. SIMULATION RESULTS

5.1. Time-based linear velocity generation

The proposed algorithm for generating time-based linear velocity is simulated for a single vehicle. For instance, we use initial and final configurations (30 m, 0 m, $\pi/2$ rad) and (90 m, 120 m, $\pi/2$ rad), respectively. The paths d_i^0 and d_i^{g} are 45 and 170 m, respectively. Both the initial and final velocities are set to 0 m/s. The generated path and linear velocities are depicted in Figs. 8 and 9, respectively. By applying Algorithms 1-7, the traveling time is found to be 28.2 s.

We also simulate the generated velocity profile calculated using the proposed method of [3], as shown in Fig. 10. Compared to [3], the proposed method is more advantageous: the velocity profile is plotted over time, while in [3], the velocity profile was plotted over the path parameter, which is not enough for solving trajectory planning in adversarial workspaces. In addition, the proposed velocity profile generation technique leads to another advantage, which is that the vehicle has a reference for the time required to reach any point on the path.

In a multiple-vehicle system, the advantage of using the proposed method is that, if there exists another vehicle on its associated path, we can determine whether the two vehicles collide with each other, as revealed in Subsection 5.2. This is the missing feature of [3]. Even though the paths intersect at a conflict point, it cannot be concluded that the vehicles will collide there. The study of [3] needs one more step to map the parameter axis to the time axis, specifically by using the relations among linear velocity, tangential acceleration, and the distance between any consecutive parameter samples. However, any pair of consecutive path-parameter samples is highly likely to have a different time gap from that of another pair. This makes the collision-checking process more complicated.

5.2. Group-based PSO search

In this subsection, the simulation result of the scenario depicted in Fig. 1 is presented to demonstrate the performance of the proposed method. The method is applied to five vehicles under the following constants: $a_i^{\mathrm{r,max}} = 1$ ms^{-2}, $a_i^{\mathrm{t,max}} = 2$ ms^{-2}, and $v_i^{\mathrm{t,max}} = 6$ ms^{-1}. The initial conditions for all vehicles are as follows: for the 1st vehicle, the initial position is (0 m, 0 m); for the second vehicle, it is (30 m, 0 m); for the third vehicle, it is (60 m, 0 m); for the fourth vehicle, it is (90 m, 0 m); and for the fifth vehicle, it is (120 m, 0 m). The initial orientation of each vehicle is $\pi/2$ rad.

The goal positions to be achieved are as follows: for the first vehicle, (120 m, 120 m); for the second vehicle, (90 m, 120 m); for the third vehicle, (60 m, 120 m); for the fourth vehicle, (30 m, 120 m); and for the fifth vehicle, (0 m, 120 m). The entry orientation in each case is $\pi/2$ rad. Additionally, the minimum vehicle-to-vehicle

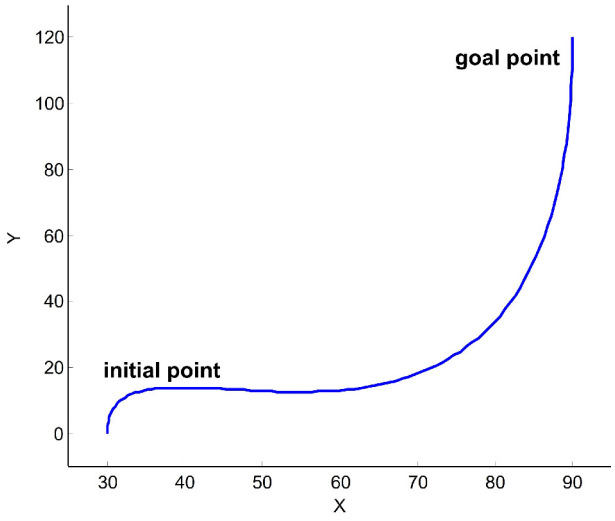

Fig. 8. Generated trajectory for a single vehicle.

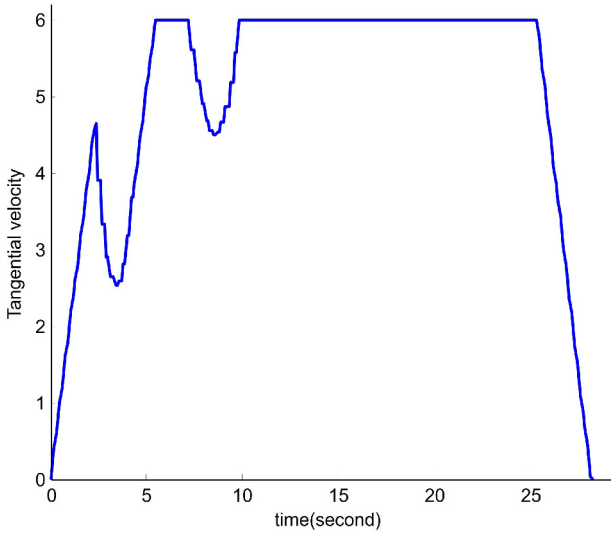

Fig. 9. Velocity plot generated by the proposed method.

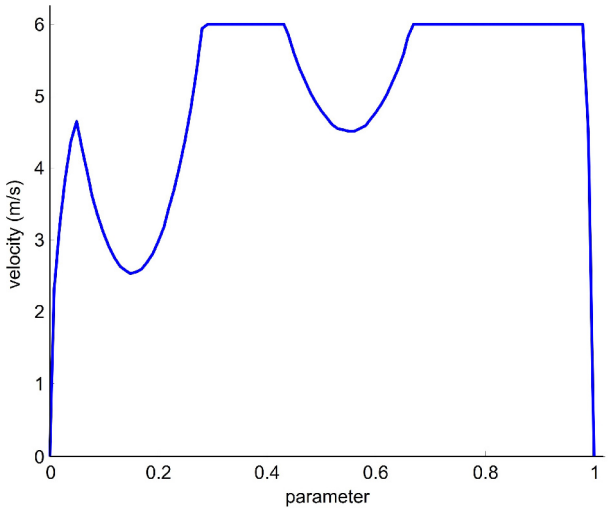

Fig. 10. Velocity plot generated by [3].

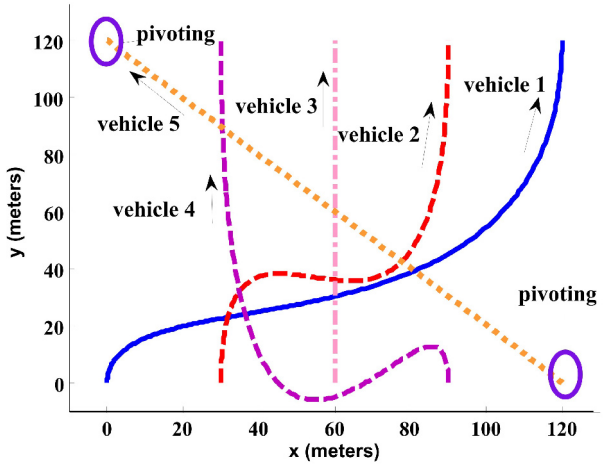

Fig. 11. Generated trajectories for 5 vehicles.

distance $\rho_{i\sim j}^{\min}$, $i \neq j$, is set as 10 m.

To search the best path and trajectories, the proposed PSO algorithm is applied with the following parameters, all set randomly: a constant inertial scaling factor, a cognitive scaling factor, and a social scaling factor of $c_i^{\text{in}} = 1$, $c_i^{\text{cg}} = 0.5$, and $c_i^{\text{sc}} = 0.5$, respectively; r_i^{cg} and r_i^{sc} as random multipliers; as well as a maximum iteration number of 20, a particle group number of 30, and initial $d_{i,k,l}^0 > 0$ and $d_{i,k,l}^{\text{g}} > 0$. The algorithm searches the minimum-time path by which the waypoint can be reached. If there is no significant obstacle to avoid, the planner generates the path to the goal.

Fig. 11 plots the trajectories of the five vehicles. Each trajectory follows the pattern of the three-degree Bezier curves with parameters d_i^0 and d_i^{g}. The resulting $\mathbf{q}_i^{\text{best}}$ is $\mathbf{q}_1^{\text{best}} = (42,27,\ 121.34)^{\text{T}}$ m, $\mathbf{q}_2^{\text{best}} = (96.31,\ 15998)^{\text{T}}$ m, $\mathbf{q}_3^{\text{best}} = (167.05,\ 83.46)^{\text{T}}$ m, $\mathbf{q}_4^{\text{best}} = (57.20,\ 229.76)^{\text{T}}$ m and $\mathbf{q}_5^{\text{best}} = (1.59,\ 1.52)^{\text{T}}$ m.

It can be concluded that the 5-th vehicle makes an almost-pivoting motion at the start and end points as the lengths of d_i^0 and d_i^{g} are extremely short compared to those of vehicles 1-4. On the other hand, the paths of vehicles 1-4 have two turning points with small curvatures (relative to vehicle 5), as their parameters d_i^0 and d_i^{g} are very large. Fig. 12 plots the vehicles' resultant velocity profiles. Vehicles 1, 2, 3, 4, and 5 start with the initial velocities equal to zero and reach their goals in 34.4 s, 27.9 s, 23.8 s, 34.0 s, and 31.2 s, respectively. The simulation successfully maintains the velocities of the vehicles at less than or equal to the maximum allowable velocities. Furthermore, it reduces them to the required velocities at the goal points without violating the velocity and acceleration constraints. Fig. 13 shows the vehicle-to-vehicle distances $\rho_{i\sim j}$, which are more than the pre-determined $\rho_{i\sim j}^{\min}$, that is,

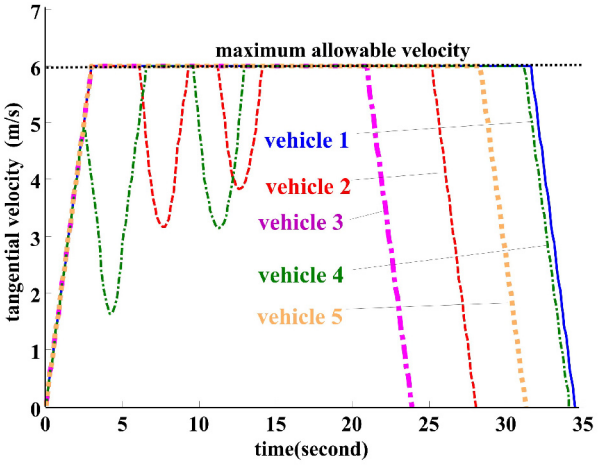

Fig. 12. Generated velocity profiles.

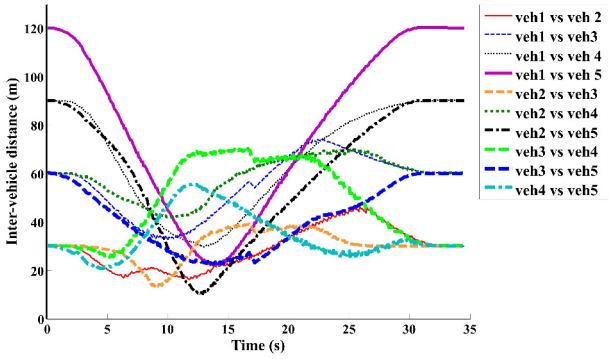

Fig. 13. Vehicle-to-vehicle distances.

10 m.

The performances in five particle-group-based PSO trials for 30 particle groups and 20 iterations are plotted in Fig. 14. Initially, the best traveling time ΔT is set very high. As can be seen, for the first and second iterations, there is no change in ΔT. Then, the search results converge within the range of $\Delta T \in [35, 48]$ s. However, only one trial yields $\Delta T = 48$ s, while the remaining ones fall within the range of $\Delta T \in [35, 40]$ s.

6. CONCLUSIONS

A PSO-based motion-planning algorithm that minimizes the traveling time of the slowest vehicle in a multiple-vehicle system by considering the radial and tangential velocities and maximum linear velocity as constraints was discussed. Three-degree Bezier curves were utilized as the basic shapes of the trajectories of the vehicles to minimize the number of parameters required to express them mathematically. Moreover, a velocity-profile-generation procedure utilizing the local minimum of the radial-accelerated linear velocity profile, which reduces

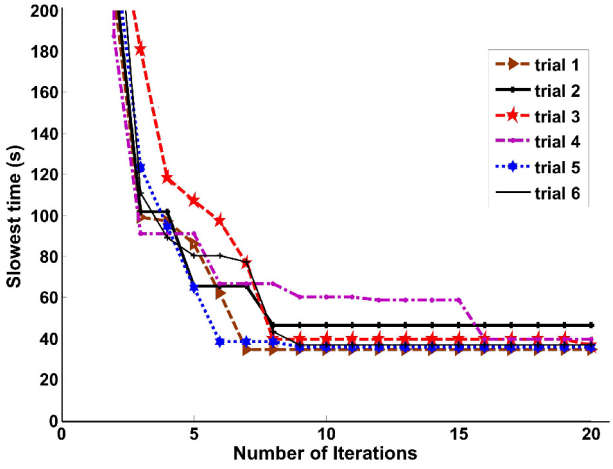

Fig. 14. Performance of the proposed PSO-based search.

the calculation effort, was introduced herein. The particle-group-based PSO search algorithm effectively minimized the traveling time of the slowest vehicle by searching for and finding the optimal combination of trajectories. Indeed, the simulation results revealed that the proposed method yielded satisfactory results: the traveling time of the slowest vehicle was minimized and the trajectories of the vehicles were collision-free. Future work will focus on more complex scenarios, for instance, the addition of non-vehicle obstacles. Moreover, the design of a control law for the tracking of generated motion plans will be explored.

REFERENCES

[1] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025-1035, December 2004.

[2] Y. J. Kanayama and B. I. Hartman, "Smooth local-path planning for autonomous vehicles," *International Journal of Robotics Research*, vol. 16, no. 3, pp. 263-284, 1997.

[3] K. G. Jolly, R. S. Kumar, and R. Vijayakumar, "A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 23-33, 2009.

[4] C. G. Lo Bianco and O. Gerelli, "Generation of paths with minimum curvature derivative with η^3-splines," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 249-256, 2010.

[5] M. Haddad, T. Chettibi, S. Hanchi, and H. E. Lehtihet, "A random-profile approach for trajectory planning of wheeled mobile robot," *European Journal of Mechanics A/Solid*, vol. 26, no. 3, pp. 519-540, 2007.

[6] M. Haddad, W. Khalil, and H. E. Lehtihet, "Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 954-962, 2010.

[7] G. Klancar and I. Skrjanc, "A case study of the collision-avoidance problem based on Bernstein-Bezier path tracking for multiple robots with known constraints," *Journal of Intelligent & Robot Systems*, vol. 60, no. 2, pp. 317-337, 2010.

[8] M. Lepetic, G. Klancar, I. Skrjanc, D. Matko, and B. Potocnik, "Time optimal path planning considering acceleration limits," *Robotics and Autonomous Systems*, vol. 45, no. 3-4, pp. 199-210, 2003.

[9] W. Yu, W. Shuo, W. Rui, and T. Min, "Generation of spatial-temporal Bezier curve for simultaneous arrival of multiple unmanned vehicles," *Information Sciences*, vol. 418-419, pp. 34-45, 2017.

[10] Q. C. Nguyen, Y. Kim, and H. Kwon, "Optimization of layout and path planning of surgical robotic system," *International Journal of Control, Automation and System*, vol. 15, no. 1, pp. 375-384, February 2017.

[11] J. Faigl and P. Vana, "Surveillance planning with Bezier curve," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 750-757, April 2018.

[12] S. M. LaValle and J. J. Kuffner, Jr., "Randomized Kinodynamic Planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378-400, May 2001.

[13] A. K. Pamosoaji and K.-S. Hong, "A path-planning algorithm using vector potential functions in triangular regions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 832-842, 2013.

[14] A. Widyotriatmo and K.-S. Hong, "A navigation function-based control of multiple wheeled robots," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 4, pp. 722-732, 2011.

[15] A. Widyotriatmo and K.-S. Hong, "Switching algorithm for robust configuration control of a wheeled vehicle," *Control Engineering Practice*, vol. 20, no. 3, pp. 315-325, 2012.

[16] Y.-S. Kim and K.-S. Hong, "A tracking algorithm for autonomous navigation of AGVs in an automated container terminal," *Journal of Mechanical Science and Technology*, vol. 19, no. 1, pp. 72-86, January 2005.

[17] A. Widyotriatmo, B. Hong, and K.-S. Hong, "Predictive navigation of an autonomous vehicle with nonholonomic and minimum turning radius constraints," *Journal of Mechanical Science and Technology*, vol. 23, no. 2, pp. 381-388, February 2009.

[18] K.-S. Hong and J. Bentsman, "Direct adaptive control of parabolic systems: Algorithm synthesis, and convergence and stability analysis," *IEEE Transactions on Automatic Control*, vol. 39, no. 10, pp. 2018-2033, October 1994.

[19] K.-S. Hong, "An open-loop control for underactuated manipulators using oscillatory inputs: steering capability of an unactuated joint," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 469-480, August 2002.

[20] K.-S. Hong, T. A. Tamba, and J.-B. Song, "Mobile robot control architecture for reflexive avoidance of moving obstacles," *Advanced Robotics*, vol. 22, pp. 1397-1420, April 2008.

[21] A. K. Pamosoaji, P. T. Cat, and K.-S. Hong, "Sliding mode and proportional-derivative-type motion control with radial basis function neural basis function neural network based estimators for wheeled vehicles," *International Journal of Systems and Science*, vol. 45, no. 12, pp. 2515-2528, July 2014.

[22] S. Islam, P. X. Liu, and A. E. Saddik., "Nonlinear robust adaptive sliding mode control design for miniature unmanned multirotor aerial vehicle," *International Journal of Control, Automation and System*, vol. 15, no. 4, pp. 1661-1668, August 2017.

[23] B. V. Galbraith, F. H. Guenther, and M. Versace, "A neural network-based exploratory learning and motor planning system for co-robots," *Frontiers in Neurorobotics*, vol. 9, no. 7, pp. 1-14, July 2015.

[24] T. T. Q. Bui and K.-S. Hong, "Evaluating a color-based active basis model for object recognition," *Computer Vision and Image Understanding*, vol. 116, no. 11, pp. 1111-1120, November 2012.

[25] E. Grinke, C. Tetzlaff, F. Wörgötter, and P. Manoonpong, "Synaptic plasticity in a recurrent neural network for versatile and adaptive behaviors of a walking robot," *Frontiers in Neurorobotics*, vol. 9, no. 11, pp. 1-15, October 2015.

[26] M. Frank, J. Leitner, M. Stollenga, A. Förster, and J. Schmidhuber, "Curiosity driven reinforcement learning for motion planning on humanoids," *Frontiers in Neurorobotics*, vol. 7, pp. 1-15, January 2014.

[27] K.-S. Hong and N. Nasser, "Reduction of delay in detecting initial dips from functional near-infrared spectroscopy signals using vector-based phase analysis," *International Journal of Neural Systems*, vol. 26, no. 3 (article number: 1640012), pp. 1-16, January 2016.

[28] E. K. Xidias and P. N. Azariadis, "Mission design for a group of autonomous guided vehicles," *Robotics and Autonomous Systems*, vol. 59, no. 1, pp. 34-43, January 2011.

[29] J. Banfi, N. Basilico, and F. Amigoni, "Intractability of time-optimal multirobot path planning on 2D grid graph with holes," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1941-1947, June 2017.

[30] X. C. Ding, A. R. Rahmani, and M. Egerstedt, "Multi-UAV convoy protection: an optimal approach to path planning and coordination," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 256-268, March 2010.

[31] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *International Journal of Robotics Research*, vol. 24, no. 4, pp. 295-310, April 2005.

[32] K. E. Bekris, D. K. Grady, M. Moll, and L. E. Kavraki, "Safe distributed motion coordination for second-order systems with different planning cycles," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 129-150, February 2012.

[33] Y. Cai and S. X. Yang, "An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments," *International Journal of Control*, vol. 86, no. 10, pp. 1720-1732, May 2013.

[34] V. Sharma, M. Savchenko, E. Frazzoli, and P. G. Voulgaris, "Transfer time complexity of conflict-free vehicle routing with no communications," *International Journal of Robotics Research*, vol. 26, no. 3, pp. 255-271, March 2007.

[35] M. Clerc and J. Kennedy, "The particle swarm – explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, February 2002.

[36] P. Yang, R. Gao, X. Pan, and T. Li, "Study on sliding mode fault tolerant predictive control based on multi agent particle swarm optimization," *International Journal of Control, Automation and Systems*, vol. 15, no. 5, pp. 2034-2042, October 2017.

[37] W. Ye, W. Feng, and S. Fan, "A novel multi-swarm particle swarm optimization with dynamic learning strategy," *Applied Soft Computing*, vol. 61, pp. 832-843, December 2017.

[38] F. Wang, H. Zhang, K. Li, Z, Lin, J. Yang, and X.-L. Shen, "A hybrid particle swarm optimization algorithm using adaptive learning strategy," *Information Sciences*, vol. 436-437, pp. 162-177, April 2018.

[39] Y. Chen and D. Wang, "Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms," *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2950-2958, December 2017.

[40] H. Freire, P. B. M. Oliveira, and E. J. S. Pires, "From single to many-objective PID controller design using particle swarm optimization," *International Journal of Control, Automation and Systems*, vol. 15, no. 2, pp. 918-932, April 2017.

Anugrah K. Pamosoaji received his B. Eng. and M. Eng. degrees in Electrical Engineering from the Institut Teknologi Bandung (ITB), Indonesia, in 2003 and 2006, respectively. He is a Ph.D. student in the School of Mechanical Engineering, Pusan National University. Recently, he is with Department of Industrial Engineering, Faculty of Industrial Technology, Universitas Atma Jaya Yogyakarta (UAJY), Indonesia. His research interests include robotics, path and trajectory planning, robotics control systems, mobile robots, and multiple-vehicle systems.

Mingxu Piao received the B.S. degree in Mechanical Engineering Department from Shanghai Maritime University, China, in 2010. He is currently an integrated Master and Ph.D. student in the School of Mechanical Engineering, Pusan National University, Korea. His research interests include sliding mode control, adaptive control, robotics, port automation, and crane control.

Keum-Shik Hong Please see vol. 13, no. 2, p. 425, April, 2015 of this journal.