7

Path Planning of Mobile Robot in Relative Velocity Coordinates

Yang Chen[1,2,3], Jianda Han[1] and Liying Yang[1,3]
[1]*State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang*
[2]*Department of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan*
[3]*Graduate School, Chinese Academy of Sciences, Beijing*
China

1. Introduction

Path planning of mobile robot in dynamic environment is one of the most challenging issues. To be more specific, path planning in multi-obstacle avoidance environment is defined as: given a vehicle *A* and a target *G* that are moving, planning a trajectory that will allow the vehicle to catch the target satisfy some specified constrains while avoiding obstacle *O*, and each of the obstacles can be either mobile or immobile in the environment. The corresponding problem is named target-pursuit and obstacles-avoidance (TPOA) and will be researched extensively in this chapter.

The traditional method, such as probability road map, can achieve a successful path in 2D static environments. The planning process using this method generally consists of two phases: a construction and a query phase. In construction stage, the workspace of the robot is sampled randomly for generating candidate waypoints. In the query stage, the waypoints between the start and goal position are connected to be a graph, and the path is obtained by some searching algorithm, such as Dijkstra, A* algorithm and so on. Hraba researched the 3D application of probability road map where A* algorithm is used to find the near-optimal path (Hrabar, 2006). Although probability road map method is provably probabilistically complete (Ladd & Kavraki, 2004), it does not deal with the environment where the information is time-varying. The underlying reason is that this method only focuses on the certain environment. Once some uncertainty appears in the robot workspace, probability road map can not update with the changing environment and plan a valid trajectory for the mobile robot, never an optimal path.

Artificial potential field is another traditional method which is generally used in both 2D and 3D environment. The mechanism that the robot is driven by attractive and repulsive force in a cooperative way is simple and often works efficiently even in dynamic environment. Kitamura et al. construct the path planning model based on the artificial potential field in three-dimensional space which is described by octree (Kitamura et al, 1995). Traditionally, artificial potential field applies in two dimensions extensively. Also some other field concepts are invented. For example, there are harmonic potential functions (Kim & Khosla, 1992; Fahimi et al, 2009; Cocaud et al, 2008; Zhang & Valavanis, 1997), hydrodynamics (Liu et al, 2007),

gradient field (Konolige, 2000), and virtual force field (Oh et al., 2007). Unfortunately, path planning approach based on the function family of potential field cannot obtain the optimal objective function which, in turn, cannot guarantee the desired optimal trajectories. Additionally, some of them, for example, potential panel method and harmonic potential function, can plan a path for an autonomous vehicle, but the computing burdens is huge and real time performance hardly satisfies the practical requirement.

Inspired by biological intelligence, many approaches, such as ant colony optimization (ACO) (Chen et al., 2008), particle swarm optimization (PSO) (Jung et al., 2006), genetic algorithm (GA) (Allaire et al., 2009), evolution algorithm (EA) (Zhao & Murthy, 2007), and their combination, are introduced to solve the path planning problem. They mostly rely on the stochastic searching, known as non-deterministic algorithm. These methods will eventually find an optimal solution, but no estimate on the time of convergence can be given. Thus, it may take long even infinite time to find the best solution. Furthermore, all of them have a great number of parameters to tune and that is never an easy job particularly when users are short of prior knowledge. Some comparisons (Allaire et al., 2009; Krenzke, 2006) show that the expensive calculations limit their real application.

Another kind of method is mathematic programming. Based on the relative velocity coordinates (RVCs), linear programming (LP) and mixed integer linear programming (MILP) can be employed for path planning problem. For example, Wang et al. converted the 2D path planning of an unmanned underwater vehicle to constrained optimization or semi-infinite constrained optimization problem (Wang et al., 2000). Zu et al. discussed the path planning in 2D and an LP method was proposed for the problem of dynamic target pursuit and obstacle avoidance (Zu et al., 2006). This method tried to plan the variables of the linear acceleration and the angular acceleration of a ground vehicle. Schouwenaars et al. proposed a MILP formulation with receding horizon strategy where a minimum velocity and a limited turn rate of aircraft are constrained (Schouwenaars et al., 2004). However, these results still focused on the two dimensions.

In this chapter, we consider the same problem of TPOA but in three dimensions. To our problem, the uncertain environment has one target or some of them, denoted by G, and many obstacles O that are all velocity-changeable with their moving actions. The aim of the vehicle A is to find an optimal path for pursuing the target while, at the same time, avoiding collision threaten from obstacles. The position and velocity of movers, including the target and obstacles, are assumed known or estimated at current time. To be more specific, it is assumed that the noise contained in the data can be eliminated with certain filters. However, this paper has no intend to discuss the filtering algorithm for achieving the feasible data of on-board sensors. The trajectory from a start to a destination location typically needs to be computed gradually over time in the fashion of receding horizon (Schouwenaars et al., 2004) in which a new waypoint of the total path is computed at each time step by solving a linear programming problem. The target and obstacles, probably static or dynamic, are modeled by spheres having a certain radius for their impact area, and the vehicle is modeled by a mass point. In order to optimize the vehicle's acceleration online, we construct a linear programming model in Cartesian orthogonal coordinates based on relative velocity space (Fiorini & Shiller, 1998).

This chapter is organized as follows. First, the relative velocity coordinates are introduced in both two dimensions and three dimensions and then the TPOA principles are introduced, including obstacle-avoidance and target-pursuit principles. The formulation of the standard linear programming is mathematically introduced in Section 3.1, and the path planning model is described in detail in Section 3.2. Simulations of path planning in 2D and 3D are

A, G, O	Vehicle, target, obstacle
τ	Planning period
k	Time step
N	Sum of obstacles
$\mathbf{V}_A$	Vehicle velocity
$V_A = \|\mathbf{V}_A\|$	Magnitude of the vehicle velocity
$\Delta\mathbf{V}_A$	Vehicle acceleration
$\mathbf{V}_O$	Obstacle velocity
$V_O = \|\mathbf{V}_O\|$	Magnitude of the obstacle velocity
$\mathbf{V}_G$	Target velocity
$V_G = \|\mathbf{V}_G\|$	Magnitude of the target velocity
$\mathbf{V}_{AO} = \mathbf{V}_A - \mathbf{V}_O$	Vehicle's relative velocity to the obstacle
$V = \|\mathbf{V}_{AO}\|$	Magnitude of the vehicle velocity relative to the obstacle
$\Delta\mathbf{V}_{AO}$	Vehicle acceleration relative to the obstacle
$\mathbf{V}_{AG} = \mathbf{V}_A - \mathbf{V}_G$	Vehicle velocity relative to the target G
$V_{AG} = \|\mathbf{V}_{AG}\|$	Magnitude of the vehicle's velocity relative to the target
$\Delta\mathbf{V}_{AG}$	Vehicle acceleration relative to the target
$\mathbf{L}_{AO}$	Vehicle position relative to the obstacle
$L = \|\mathbf{L}_{AO}\|$	Magnitude of the vehicle position relative to the obstacle
$\mathbf{L}_{AG}$	Vehicle position relative to target
$L_{AG} = \|\mathbf{L}_{AG}\|$	Magnitude of the vehicle position relative to the target
$P = \mathbf{V}_{AO}^{\mathrm{T}}\mathbf{L}_{AO}$	Temporal variable
$P_{AG} = \mathbf{V}_{AG}^{\mathrm{T}}\mathbf{L}_{AG}$	Temporal variable

Table 1. Nomenclatures used in this chapter (see Fig.1 and Fig. 2).

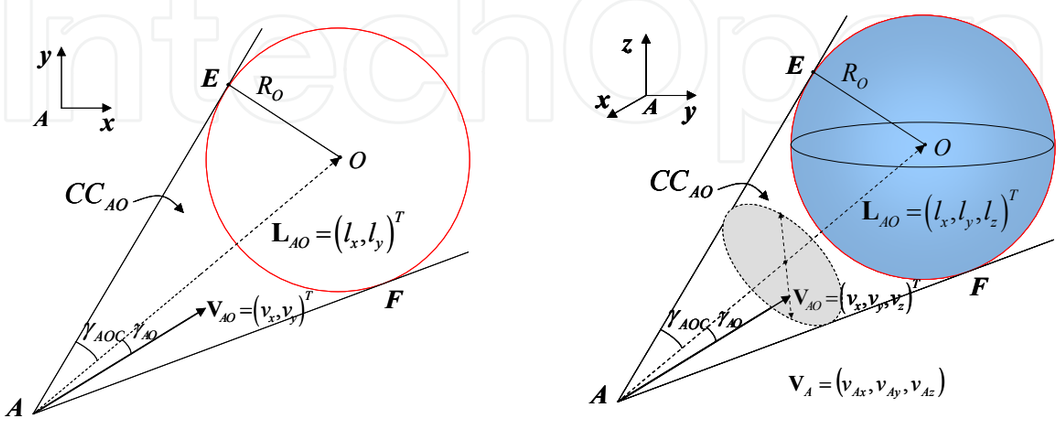
shown in Section 4.1 and 4.2, respectively. In Section 5, an application about the multiple task assignment (MTA) is used to verify the method proposed in this chapter. Finally, a brief conclusion is drawn in Section 6.

2. Relative velocity coordinates and the TPOA principles

2.1 Relative velocity coordinates

RVCs are constructed on the mass point of the vehicle as shown in Fig.1 and Fig.2. Fig.1 shows the relative coordinates when the obstacle-avoiding is considered while Fig.2 shows the scenario where the target-pursuing is considered. In these figures, the target G and obstacle O, which are 2D or 3D movers, are assumed as circles or spheres having

certain radiuses which are denoted by R_Os. As mentioned above, the relative parameters are measurable or estimable for the current planning time, by using certain on-board sensors. For the convenience of description, the relative velocity and relative position between the vehicle A and the target G are denoted by V_{AG} and L_{AG}. Similarly, parameters about the obstacle O are denoted by V_{AO} and L_{AO}. Some other nomenclatures are listed in Table 1.



(a) Obstacle avoidance in two dimensions

(b) Obstacle avoidance in three dimensions

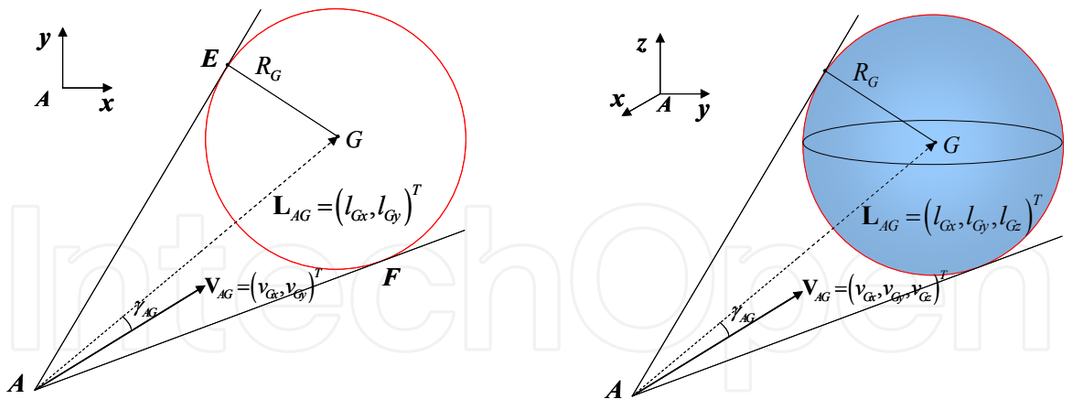
Fig. 1. Geometrical representation of the relative parameters in the relative coordinates when the obstacle-avoiding problem is considered. (a) and (b) show the definition of *relative obstacle angle* in 2D scenario and 3D scenario, respectively.

We define the *relative obstacle angle*, γ_{AO}, as the angle between $\mathbf{V}_{AO}$ and $\mathbf{L}_{AO}$. $\gamma_{AO}\in[0,\pi]$. *Collision cone* is defined as an area in which the vehicle will collide with an obstacle by current velocity $\mathbf{V}_{AO}$, and it is denoted by CC_{AO}. Generally, CC_{AO} is in the cone AEF, as shown in Fig.1. The half cone angle of CC_{AO}, γ_{AOC}, is defined as *collision region angle*, i.e.

$$\gamma_{AOC}=\arcsin\left(\frac{R_O}{L_{AO}}\right) \tag{1}$$

Similar to the definition in the obstacle-avoiding scenario in Fig.1, the angle between the relative velocity $\mathbf{V}_{AG}$ and relative distance $\mathbf{L}_{AG}$ is defined as *relative target angle*, denoted by $\gamma_{AG}\in[0,\pi]$, as shown in Fig.2. The *pursuit cone*, as well as the *pursuit region angle*, is defined in the same as obstacle avoidance scenario.

Some assumptions should be declared as the prerequisites of the planning principles. First, in order to guarantee the robot to catch the target successfully, the maximum velocity of the robot is assumed to be larger than that of the target and the obstacles. Due to this precondition, the vehicle A has the ability to catch the target, as well as avoid the collision with obstacles. Second, the target and the obstacles are supposed to keep their speed as constants during the planning period, τ. In fact, this hypothesis is usually accepted because τ is short enough for a real vehicle. Owe to the mechanism of numerical approximation, the path planning problem can be solved and optimized with a receding horizon fashion in



(a) Target pursuit in two dimensions (b) Target pursuit in three dimensions

Fig. 2. Geometrical representation of the relative parameters in the relative coordinates when the target-pursuing problem is considered. (a) and (b) show the definition of *relative target angle* in 2D scenario and 3D scenario, respectively.

which a new waypoint of the total path is computed gradually over time. Here, we obtain the following equivalence.

$$\Delta\mathbf{V}_{AO} = \Delta\mathbf{V}_{AG} = \Delta\mathbf{V}_{A} \tag{2}$$

2.2 Obstacle-avoidance principle

For each obstacle O, under the assumption that its velocity is constant in τ, it will be avoided if the γ_{AO} is large enough to make the $\mathbf{V}_{AO}$ out of the CC_{AO} over the interval τ when the vehicle moves from time step k to k+1. This fact suggests that the obstacle-avoiding principle hold the following inequality. That is

$$\gamma_{AOC(k)} \leq \gamma_{AO(k+1)} \leq \pi \tag{3}$$

where $\gamma_{AOC(k)}$ is the *collision region angle* in time step k, which is shown in Fig.1. If there are multi obstacles, Eq. (3) changes to

$$\gamma_{AOCi(k)} \leq \gamma_{AOi(k+1)} \leq \pi \tag{4}$$

where the subscript i denotes the label of the obstacles. i=1, 2… N. N stands for the number of obstacles.

2.3 Target-pursuit principle

The $\mathbf{V}_{AG}$ can be resolved into a pair of orthogonal components, as shown in Fig. 3. The vehicle is expected to tune its velocity to the optimum. Only when the velocity direction of the vehicle is identical to $\mathbf{L}_{AG}$, it will not lose its target. In the meanwhile, the vehicle should better minimize the magnitude of $\mathbf{V}_{\mathrm{T}}$. Consequently, the policy for the optimal velocity is obvious in that $\mathbf{V}_{\mathrm{T}}$ should be minimized while $\mathbf{V}_{\mathrm{C}}$ maximized. We formulate these principles as two cost functions. They are

$$\min : V_T = \|\mathbf{V}_{AG}\| \sin \gamma_{AG} \tag{5}$$

and

$$\max : V_C = \|\mathbf{V}_{AG}\| \cos \gamma_{AG} \tag{6}$$

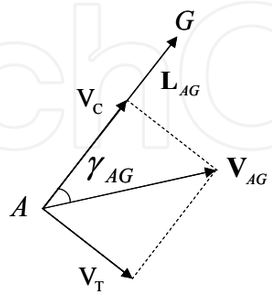


Fig. 3. The resolution of $\mathbf{V}_{AG}$. The velocity of the vehicle relative to the target is resolved into two orthogonal components, $\mathbf{V}_C$ and $\mathbf{V}_T$. One component, $\mathbf{V}_C$, is in the direction of $\mathbf{L}_{AG}$ and pointed to the target. The other, $\mathbf{V}_T$, is orthogonal to $\mathbf{L}_{AG}$.

3. Linear programming model

In this section, we first introduce some knowledge about the mathematical description of the standard linear programming model. Then, the mentioned principles for path planning are decomposed and linearized according to the linear prototype of the standard model.

3.1 Standard linear programming model

The standard Linear Programming is composed of a cost function and some constrains, all of which need to be affine and linear (Boyd & Vandenberghe, 2004). The following is the standard and inequality form.

$$\begin{aligned} &\min_{\mathbf{x}} : \mathbf{c}^T\mathbf{x} \\ &\text{subject to:} \mathbf{D}\mathbf{x} \le \mathbf{h} \end{aligned} \tag{7}$$

where $\mathbf{D} \in R^{m \times n}$. The vector $\mathbf{x} \in R^{n \times 1}$ denotes the variables. In the standard linear programming formulation, the vector $\mathbf{c} \in R^{n \times 1}$, representing the coefficients, the vector $\mathbf{h} \in R^{m \times 1}$, and the matrix $\mathbf{D}$ are all known when the model is going to be solved by some method.

During the decades, several methods have been invented to solve the optimization as shown in (7). From those methods, simplex algorithm and interior-point method are two most famous algorithms. More information about these algorithms can be found in (Boyd & Vandenberghe, 2004). In this chapter, an open library named Qsopt is used where the primal and dual simplex algorithms are imbedded. For the corresponding procedures, the user can specify the particular variant of the algorithm which is applied (David et al., 2011).

3.2 Linear programming model for path planning problem

In this subsection, the problem of obstacle-avoidance is formulated and we obtain some inequality constraints after we introduce two slack variables. Additionally, the formulation of

the target-pursuit problem is transferred to some cost functions. The final linear programming model consists of a weighted sum of the cost functions and a set of linear constraints. It is

$$\min : J = \sum_j \omega_j d_j + \omega_{v1} q_1 + \omega_{v2} q_2 \tag{8}$$

satisfying

$$\sum_j \omega_j + \omega_{v1} + \omega_{v2} = 1 \tag{9}$$

where ω_j, ω_{v1}, $\omega_{v2} \geq 0$. The letter j denotes the subscript of the components of $\mathbf{V}_{AG}$. If the path planning is operated in two-dimensional environment, j belongs to a two-element-set, i.e., $j \in \{x, y\}$. And if in three-dimensional environment, j belongs to a three-element-set, i.e., $j \in \{x, y, z\}$. The variables d_j, q_1, q_2 will be explained later.

3.2.1 Linearization of the avoidance constrain

From Fig.1, we get the *relative obstacle angle* function γ_{AO}.

$$\gamma_{AO} = \arccos\left(\frac{\mathbf{V}_{AO} \cdot \mathbf{L}_{AO}}{\|\mathbf{V}_{AO}\| \cdot \|\mathbf{L}_{AO}\|} \right) = \arccos\left(\frac{P}{VL} \right) \tag{10}$$

The definitions of V, L, and P refer to Table 1. Here, Eq. (10) is linearized by Taylor's theorem and we obtain the following equation.

$$\begin{aligned} \gamma_{AO(k+1)} &= \gamma_{AO(k)} + \Delta\gamma_{AO(k)} \\ &= \gamma_{AO(k)} + \tau \left(\frac{d\gamma_{AO}}{d\mathbf{V}_{AO}} \right)_{(k)} \frac{d\mathbf{V}_{AO}}{dt} + o\left(\|\Delta\mathbf{V}_{AO}\| \right) \end{aligned} \tag{11}$$

$$\Delta\gamma_{AO(k)} = -\frac{\tau}{\sqrt{V^2L^2 - P^2}\Big|_{(k)}} \left(\mathbf{L}_{AO} - \frac{P}{V^2}\mathbf{V}_{AO} \right)_{(k)} \Delta\mathbf{V}_{AO} \tag{12}$$

Let $\gamma_{AO(k+1)}$ represent the *relative obstacle angle* in time step k+1 after vehicle's movement. The variables, $\Delta\mathbf{V}_{AO}$, in (11), are that we are trying to plan in step k for step k+1. If there are multiple obstacles, Eq. (12) changes to Eq. (13).

$$\Delta\gamma_{AOi(k)} = -\frac{\tau}{\sqrt{V^2L^2 - P^2}\Big|_{i,(k)}} \left(\mathbf{L}_{AO} - \frac{P}{V^2}\mathbf{V}_{AO} \right)_{i,(k)} \Delta\mathbf{V}_{AO} \tag{13}$$

where

$$\gamma_{AOCi} = \arcsin\left(\frac{R_{Oi}}{L_i} \right) \tag{14}$$

3.2.2 Minimize the relative distance

In this subsection, the relative distance between the robot and the target-objective is minimized. By tuning the velocity of the robot, the relative distance between the robot

and target will became smaller through each step time. This distance is resolved into two or three elements by the axis number. That is to minimize the elements of the vector $\mathbf{L}_{AG}$, i.e.,

$$\min : \left\| l_{Gj} - \left(v_{Gj}\tau + \Delta v_{Gj}\tau^2 \right) \right\| \tag{15}$$

where l_{Gj} is the element of $\mathbf{L}_{AG}$ and v_{Gj} is the element of $\mathbf{V}_{AG}$. See Fig.2. Δv_{Gj} denotes the acceleration of v_{Gj}. $j \in \{x, y\}$ is for two-dimensional environment and $j \in \{x, y, z\}$ is for three-dimensional environment. The inherited objective function is derived from Eq. (15).

$$\begin{aligned} &\min : d_j \\ &\text{subject to} : -d_j \le l_{Gj} - \left(v_{Gj}\tau + \Delta v_{Gj}\tau^2 \right) \le d_j \end{aligned} \tag{16}$$

where Δv_{Gj} is the variable that we hope to compute in the linear programming model, as stated in Eq. (2). In a manner of preserving convexity of the problem, $d_j \ge 0$ is the newly introduced slack variable associated with each element (Boyd & Vandenberghe, 2004).

3.2.3 Optimize the relative velocity

The relative velocity between the robot and the target-objective needs to optimize on the consideration of target-pursuit principle. We respectively discuss the optimization of the pair component, $\mathbf{V}_T$ and $\mathbf{V}_C$, as shown in Fig. 3.

(1) Minimize the magnitude of the component $\mathbf{V}_T$. We compute $\mathbf{V}_T$ with

$$V_T = \sqrt{V_{AG}^2 - \frac{P_{AG}^2}{L_{AG}^2}} \tag{17}$$

So the optimization of (5) is equal to

$$\min : g\left(\vec{V}_{AG}\right) = V_T^2 = V_{AG}^2 - \frac{P_{AG}^2}{L_{AG}^2} \tag{18}$$

Using Taylor's theorem and introducing slack variable, q_1, we get the new formulation for optimization.

$$\begin{aligned} &\min : q_1 \\ &\text{subject to:} \quad 0 \le g\left(\mathbf{V}_{AG}\right) + \nabla g \Delta \mathbf{V}_{AG}^{\mathrm{T}} \le q_1 \\ &\qquad\qquad 0 \le q_1 \le V_{AG\max}^2 \end{aligned} \tag{19}$$

where $\Delta \mathbf{V}_{AG}$ is the variable vector that we hope to include in the linear programming model. q_1 is the slack variable. ∇g represents the grades of the function g(.). It is computed with

$$\nabla g = 2\tau \left(\mathbf{V}_{AG} - \frac{P_{AG}}{L_{AG}^2} \mathbf{L}_{AG} \right) \tag{20}$$

$V_{AG\max}$ is the maximum of the relative velocity between the robot and the target. We estimate it by $V_{AG\max} = 2V_{A\max}$. $V_{A\max}$ denotes the upper boundary of V_A.

(2) Maximize the magnitude of the component $\mathbf{V}_C$.
Since we can maximize $\mathbf{V}_C$ by minimizing -$\mathbf{V}_C$, we refer to a minimize problem with affine cost function and constraint functions as a linear programming. Consequently, the problem described by (6) can be rewritten as

$$\min : -\mathrm{V_C} = -\left\|\mathbf{V}_{AG}\right\|\cos\gamma_{AG} \tag{21}$$

Linearized using Taylor's theorem, Eq. (21) changes to a standard linear programming problem. That is

$$\begin{aligned} &\min : q_2 \\ &\text{subject to: } -V_C - \nabla V_C \Delta \mathbf{V}_{AG}^{\mathrm{T}} \le q_2 \\ &\qquad -V_{AG\max} \le q_2 \le V_{AG\max} \end{aligned} \tag{22}$$

where $\Delta\mathbf{V}_{AG}$ is the variable that we are trying to include in our model. q_2 is also a slack variable. Here, the grades is computed with $\nabla\mathbf{V}_C = \tau\,\mathbf{L}_{AG}/\mathrm{L}_{AG}$.

3.3 Other constraints

One of the advantages of the LP method is that various constraints can easily be added in the opening constraint set. Here we provide some demonstrations of constraints that are transformed from the dynamics, sensor data, and searching boundaries.

3.3.1 The constraints from the kinematics and dynamics

The kinematics and dynamics are extremely simplified in the path planning model where only the bound of the velocity and acceleration are added in the feasible set. These bound can be described mathematically as

$$\begin{cases} -V_{A\max} \le v_{Aj} \le V_{A\max} \\ -\Delta_{\max} \le \Delta v_j \le \Delta_{\max} \end{cases} \tag{23}$$

where v_{Aj} denotes the components of $\mathbf{V}_A$, as shown in Fig. 1. $\Delta_{\max}$ denotes the max magnitude of the changing of v_{Aj} in each period τ.

$$-V_{A\max} \le v_{Aj} + \Delta v_j \tau \le V_{A\max} \tag{24}$$

We get the net constraints (25) from the simultaneous equations (23) and (24).

$$\max\left\{-\Delta_{\max}, -\frac{1}{\tau}\left(V_{A\max} + v_{Aj}\right)\right\} \le \Delta v_j \le \min\left\{\Delta_{\max}, \frac{1}{\tau}\left(V_{A\max} - v_{Aj}\right)\right\} \tag{25}$$

3.3.2 The constraints from the limit of the sensor

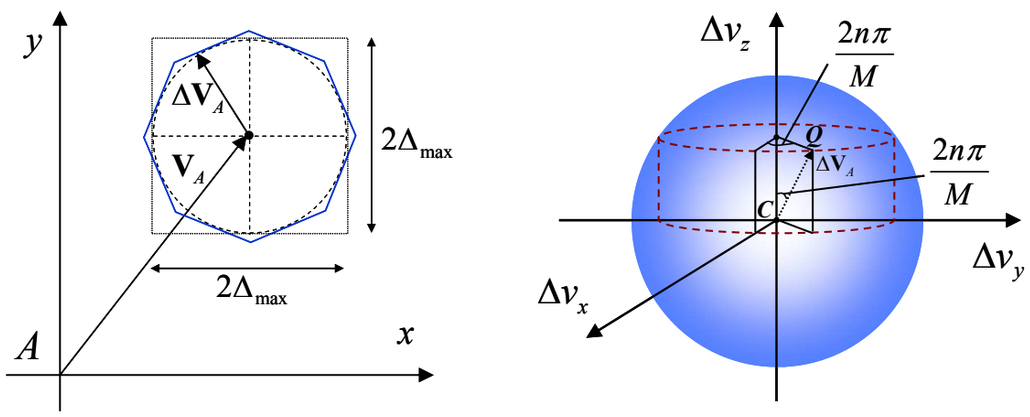
All sensors can not detect the area that is beyond their capability. In real application, the obstacle that has been avoided by the vehicle will possibly disappear from the sensor's operating region. Similarly, any new obstacle from long distance is possible entering the detecting area. On this point, we evaluate the threat of each obstacle in the environment and propose a threat factor for each of them to estimate the performance. More specially, if the relative distance between the vehicle and the obstacle O_i satisfies $\mathrm{L}_{AOi} \ge \mathrm{L}_{\min i}$, this

obstacle is assigned λ_i=0. L_{min} is the threshold. On the contrary, O_i is assigned λ_i=1 if $L_{AOi}<L_{min}$.
In addition, the *relative obstacle angle* has impact on the vehicle. If the velocity vector of the vehicle satisfies $\gamma_{AOi} \geq \gamma_{AOCi} + \Delta\gamma_{AOimax}$, the vehicle can be looked as moving from the obstacle O_i, λ_i will be assigned zero.

3.3.3 The constraints from the searching region

The domain of the variables is generally computed in a square in 2D or a cube in 3D for the interceptive magnitude limit of the acceleration on each axis. In fact, the acceleration on each axis is coupled. So if the domain is treated as a circle in 2D (Richards & How, 2002) or a sphere in 3D, the algorithm will lower its conservation. See Fig. 4. We denote the domain as *dom*. That is

$$\Delta\mathbf{V}_A \in dom \tag{26}$$



(a) Searching region in 2D; (b) Searching region in 3D

Fig. 4. Searching region for acceleration. (a) The domain of the acceleration will be approximated by multiple lines in 2D. (b) The domain of the acceleration will be approximated by multiple planes in 3D.

We try to approximate the *dom* with multiple lines in 2D, or multiple planes in 3D. That is

$$\sin\frac{2m\pi}{M}\Delta v_x + \cos\frac{2m\pi}{M}\Delta v_y \leq \Delta_{\max} \tag{27}$$

where m=0,1,2,…, M-1. Here, M represents the quantity of the line used for approximation. In three-dimensional environment,

$$\sin\frac{2m\pi}{M}\cos\frac{2n\pi}{M}\Delta v_x + \sin\frac{2m\pi}{M}\sin\frac{2n\pi}{M}\Delta v_y + \cos\frac{2m\pi}{M}\Delta v_z \leq \Delta_{\max} \tag{28}$$

where m, n=0,1,2,…,M-1. Here, M represents the quantity of the plane used for approximation. At this point, the linear programming model is formulized. The objective function is (8). All constrains are recapitulated as the following.

$$\begin{cases} \lambda_i \gamma_{AOCi} \le \lambda_i \left(\gamma_{AOi} + \Delta\gamma_{AOi} \right) \le \pi \\ -d_j \le l_j - \left(v_j \tau + \Delta v_j \tau^2 \right) \le d_j \\ 0 \le g\left(\mathbf{V}_{AG} \right) + \nabla g \cdot \Delta \mathbf{V}_{AG}^T \le q_1 \\ -\mathrm{V}_C - \nabla \mathbf{V}_C \Delta \mathbf{V}_{AG}^T \le q_2 \\ \max\left\{ -\Delta_{\max}, -\frac{1}{\tau}\left(\mathrm{V}_{A\max} + v_{Aj} \right) \right\} \le \Delta v_j \le \min\left\{ \Delta_{\max}, \frac{1}{\tau}\left(\mathrm{V}_{A\max} - v_{Aj} \right) \right\} \\ \Delta \mathbf{V}_A \in dom \end{cases} \tag{29}$$

4. Path planning simulation for TPOA simulation

The approach proposed in this chapter is simulated with three obstacles and the results are given scenarios of in 2D and 3D, respectively. See Fig.5 and Fig.6. All the simulations run on the platform of WinXP/Pentium IV 2.53 GHz/2G RAM. A linear programming solver, named QSopt, is called from the library (David et al., 2011). We run the examples with three obstacles and give the results in two-dimensional environment and three-dimensional environment, respectively.

4.1 Path planning in 2D

According to the LP model, as shown with Eq. (29), all initial parameters are listed in Table 2. Assuming that the maximal velocity of the vehicle in 2D is 50cm/s, and the maximal acceleration of the vehicle is 350cm/s^2. The panning period is τ=20ms. The three parameters will be kept the same in the following simulation.

Fig. 5(a)~(d) show the key scenarios when avoiding the three obstacles. Fig. 5(a) shows the situation when the robot avoiding SO. It is obvious that the robot can rapidly adjust its velocity to the most favorable one and go on pursuing the target. The planner calls LP algorithm about 13 times in this stage. Fig. 5(b) shows the case while the robot has to avoid the moving obstacle MO1. At this time, the robot turns left to avoid MO1 because the over speeding is accessible for the robot. Fig. 5(c) shows the different decision that the robot selected to avoid MO2 from its back. Two conclusions can be drawn from this simulation. First, the robot is indeed qualified the avoiding and pursuing ability in uncertain environment. Second, the robot can adjust its velocity autonomously, including the magnitude and the direction, and adapt the optimal decision to avoid the obstacle and to catch the target. The whole pursuing process of this example lasts about 640ms and each period of calculation of our algorithm takes only about 0.469ms which is a low time complexity for real-time application.

	Initial Position (cm)	Initial Velocity (cm/s)	Radius (cm)
Robot (A)	(0,0)	(10,0)	N/A
Target (G)	(1000,1000)	(-12,0)	50
Static Obstacle (SO)	(300,300)	N/A	100
Moving Obstacle (MO1)	(800,600)	(-20,0)	50
Moving Obstacle (MO2)	(300,900)	(6,-3)	50

Table 2. The initial parameters for simulation in 2D.

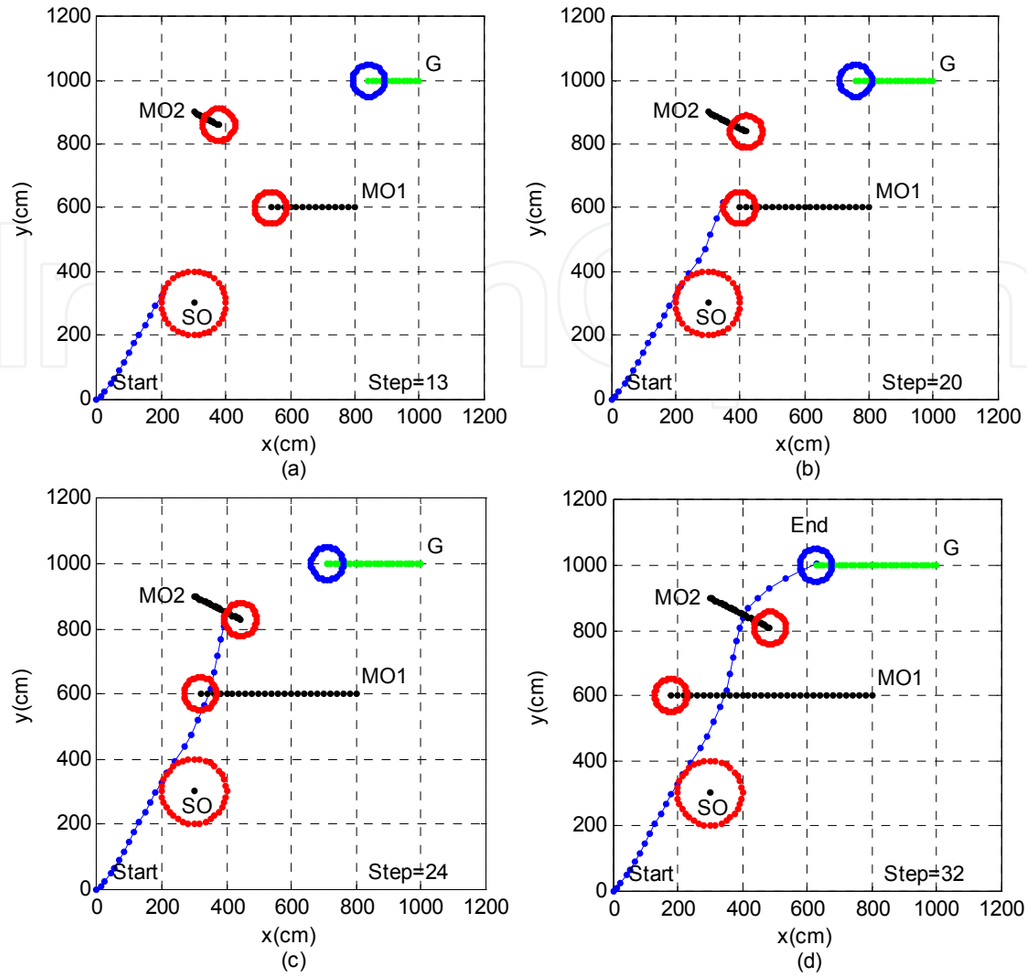


Fig. 5. Simulation in dynamic environment of 2D. The dot-line denotes the trajectories of the robot, the obstacles, and the target. And the circles with dot-line edge are the shape profile of any mover. SO represents static obstacle. MO1 and MO2 represent moving obstacles. *G* represents the moving target.

4.2 Path planning in 3D

For the specific 3D environment, assuming that there is one static obstacle SO1, and two moving obstacles, MO1 and MO2. All the initial parameters are listed in Table 3.

	Initial Position (cm)	Initial Velocity (cm/s)	Radius (cm)
Robot (A)	(0,1000,0)	(15,-15,0)	N/A
Target (G)	(1000,0,1000)	(-5,5,0)	50
Static Obstacle (SO)	(300,700,300)	N/A	100
Moving Obstacle (MO1)	(450,500,900)	(0,0,-10)	80
Moving Obstacle (MO2)	(450,250,850)	(0,10,0)	50

Table 3. The initial parameters for simulation in 3D.

Fig. 6 shows the results. At the beginning, SO is on the line between the robot and the target, and the robot is blocked. In the following time, the robot avoids MO1 and MO2 at step=19 and step=24, respectively. See Fig. 6(b) and (c). It is evident that the velocity-decision of the robot is optimized online. The time complexities of this simulation are 0.556ms in every period averagely.

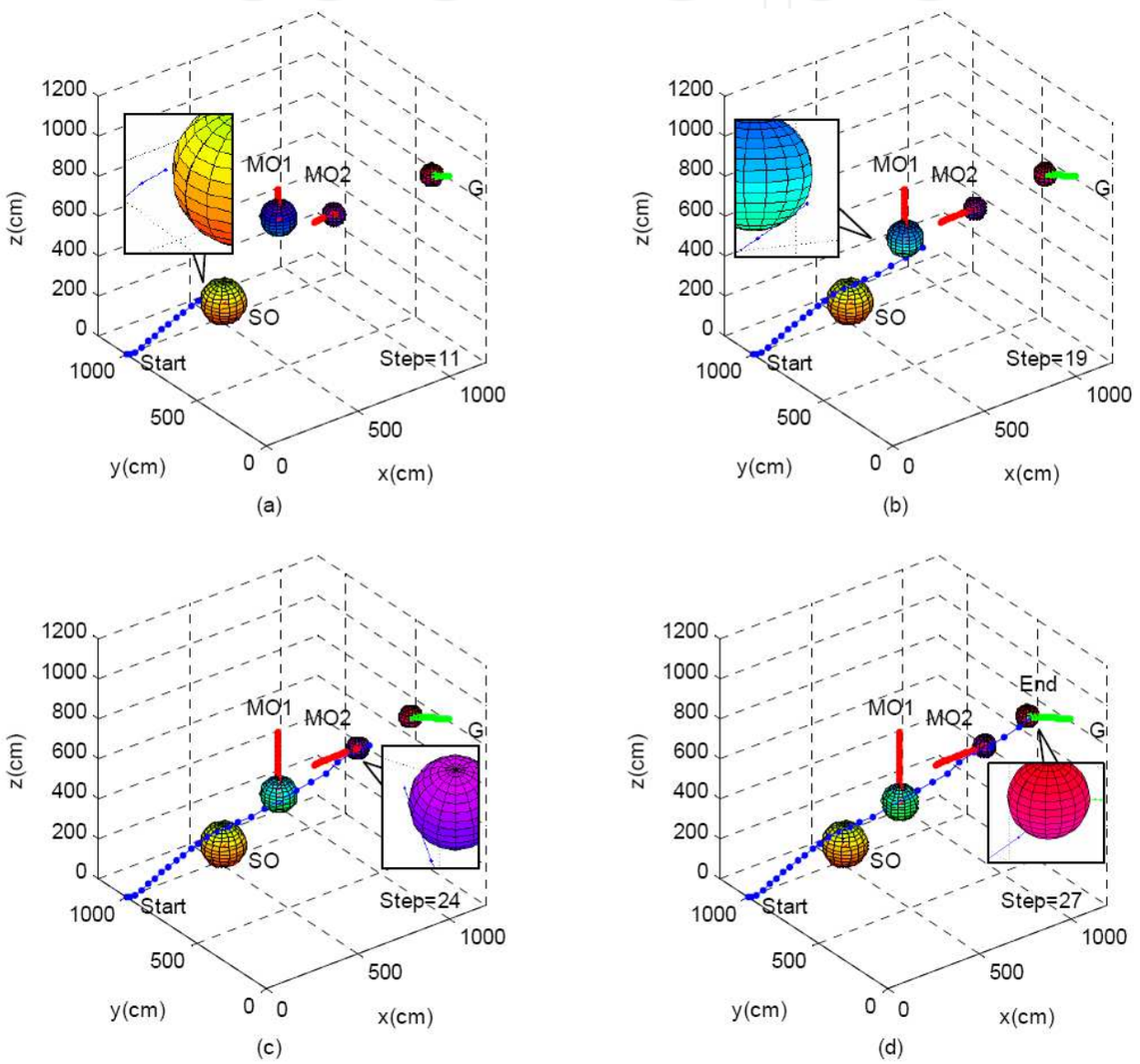


Fig. 6. Simulation in dynamic environment of 3D. The dot-line denotes the trajectories, and the sphere denotes the target and the obstacles.

5. Path planning for multiple tasks planning (MTP): an application

MTP problem is a variant of multiple TPOA. It can be stated that given N targets and N vehicles, let each vehicle can only pursue one target and each target is pursued exactly by one vehicle at any interval. The task is finished until all the targets have been caught. Then, the goal of MTP problem is to assign the tasks to vehicles within each interval online as well as complete the whole mission as fast as possible.

In this section, MTP problem is decomposed into two consecutive models, i.e., the task-assignment model and the path planning model. The path planning model has already been introduced above. With respect to the task-assignment model, a minimax assignment criterion is proposed to direct the task assignment. Under this assignment criterion, the optimal assignment will cost the least time to catch all the targets from the current measurable information.

Some simulations in Fig. 7 are given to show the efficiency of the method. Vehicles pursuing target are assigned according to the minimax assignment. According to the criterion, the current task assignment prefers to finish the whole mission fastest (Yang et al., 2009).

5.1 Assignment model under minimax assignment criterion

In the assignment problem, the payment in each vehicle for a target is assumed to be known. The problem is how to assign the target for each particular vehicle that the total pursuit mission can be completed as fast as possible.

Let x_{ij} be the n^2 0-1 decision variables, where $x_{ij}=1$ represents vehicle i for target j; otherwise, $x_{ij}=0$. Because the "payment" is understood as time, it is important to minimize the maximal time expended by vehicle for its pursuing process. This task assignment problem may be described as the minimax assignment model mathematically.

$$\begin{aligned} &\min : z = \max_{i,j}\{c_{ij}x_{ij}\} \\ &\text{subject to: } \sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1,...,n) \\ &\qquad\qquad \sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1,...,n) \\ &\qquad\qquad x_{ij}=0 \text{ or } 1, \ \ i,j \in \{1,...,n\} \end{aligned} \tag{30}$$

where c_{ij} represents the payment in vehicle i for target j and will be given in Section 5.3. The elements c_{ij} give an n^2 binary cost matrix. This is a classic integer programming where the objective function is nonlinear and difficult to solve.

A solution method for the above minimax assignment problem named the operations on matrix is proposed. This solution method finds the solution directly from the cost matrix $C_n = (c_{ij})$ and the objective function (Yang et al., 2008).

5.2 Global cost function

We give the cost function of this MTP problem as the following. Let N_v, N_T, and N_o be the number of vehicles, targets and obstacles, respectively. The payment of the vehicle-i to pursue the target-j in the assignment problem is written as:

$$c_{ij} = \frac{d_{ij} + \xi_1 \sum_{k=1,k\neq i}^{N_v+N_o} d_{kij}}{\Delta v_{ij}} + \frac{\xi_2 \left|\Delta\Phi_{ij}\right|}{\omega_i} \tag{31}$$

where d_{ij}, Δv_{ij} and $\Delta\Phi_{ij}$ are, respectively, the distance, velocity difference and heading difference between vehicle-i (V-i) and target-j (T-j). ω_i is the maximum turning rate of V-i; d_{kij} is the "additional distance" due to the obstacles and the vehicles other than V-i itself (with respect to vehicle-i, not only the obstacles, but also the other vehicles are its obstacles). ξ_1 and ξ_2 are positive constants. the c_{ij} in Eq. (33) indicates the time possibly needed by V-i to pursue T-j, and the "extended distance" consists of the linear distance, the angle distance, as well as the possible obstacle avoidance between V-i and T-j (Yang et al., 2009).

5.3 Simulation of MTP in 2D

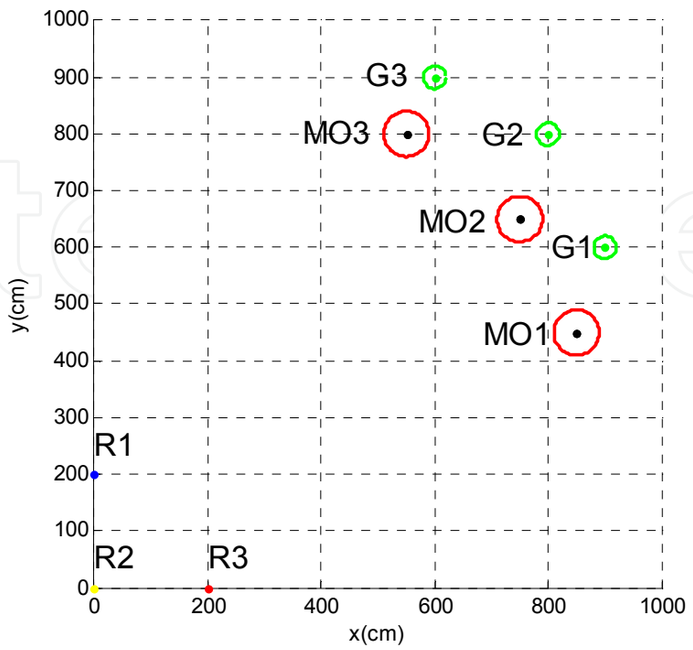
The simulations demonstrated here include three robots, three moving target and three moving obstacles in two-dimensional environment (see Fig. 7). All the initial parameters are listed in Table 4.

	Initial Position (cm)	Initial Velocity (cm/s)	Radius (cm)
Robot (R1)	(0,200)	(17,0)	N/A
Robot (R2)	(0,0)	(17,0)	N/A
Robot (R3)	(200,0)	(17,0)	N/A
Target (G1)	(900,600)	(-11,9)	40
Target (G2)	(800,800)	(-10,10)	40
Target (G3)	(600,900)	(-9,11)	40
Moving Obstacle (MO1)	(850,450)	(-11,9)	60
Moving Obstacle (MO2)	(750,650)	(-10,10)	60
Moving Obstacle (MO3)	(550,750)	(-9,11)	60

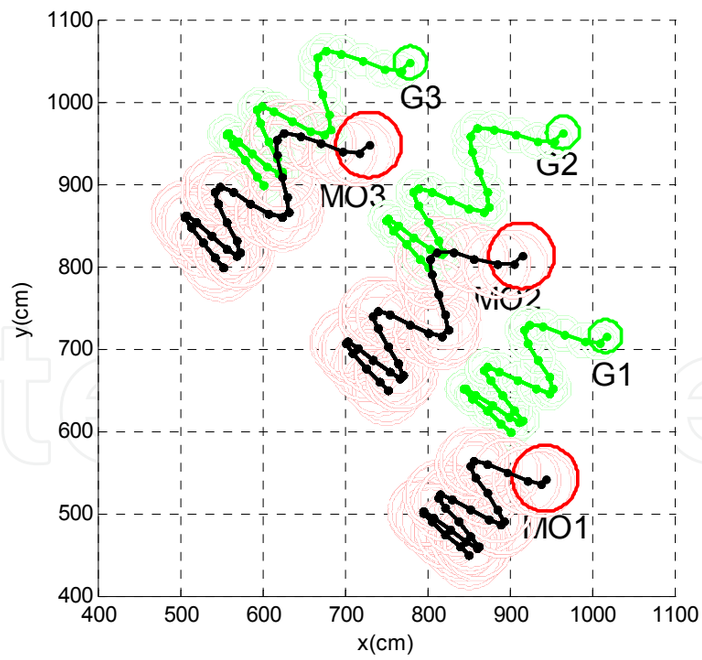
Table 4. The initial parameters for MTP simulation in 2D.

In order to test the performance of the proposed linear programming model in the presence of different-motion vehicles, the following modifications are made during the pursuit process.

① at k=0, the maximal robot velocities are $v_{1\text{-max}}$=30 cm/s, $v_{2\text{-max}}$=68 cm/s, $v_{3\text{-max}}$=45 cm/s;
② at k=16, the maximal robot velocities are $v_{1\text{-max}}$=30 cm/s, $v_{2\text{-max}}$=68 cm/s, $v_{3\text{-max}}$=22.5 cm/s; ③ at k=31, the maximal robot velocities are $v_{1\text{-max}}$=42.5 cm/s, $v_{2\text{-max}}$=68 cm/s, $v_{3\text{-max}}$=22.5 cm/s.

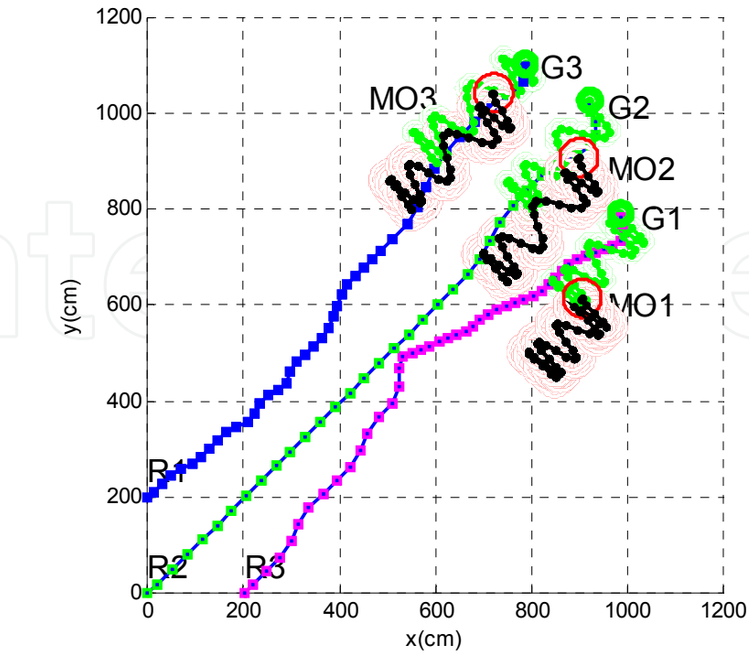


(a)

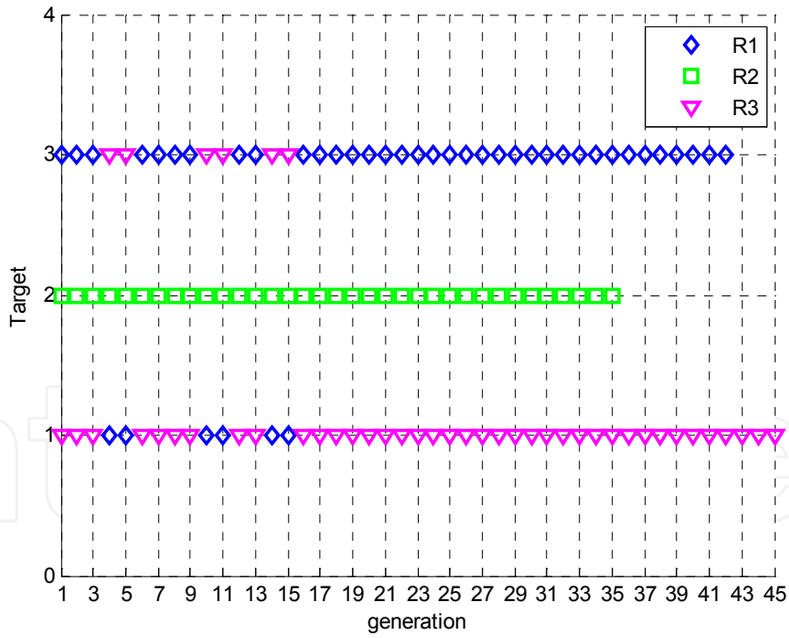


(b)

Fig. 7. Robots, targets and obstacles.
(a) Initial positions at k=0; (b) The target and obstacle trajectories till k=30. The targets and the obstacles are moving in a pso-sine curve.



(a)



(b)

Fig. 8. Simulation of MTP in 2D. (a) Planning trajectories; (b) robot-target assignment pairs.

As Fig. 8 shows, the pair assignment alters between {R1-G3, R2-G2, R3-G1} and {R1-G1, R2-G2, R3-G3} at initial 15 steps according to the minimax assignment computation. At time step 16, assignment is kept for {R1-G3, R2-G2, R3-G1} due to that R3 is the slowest robot after its maximal velocity reducing 50%, and it can only catch G1. R2 is the fastest robot, so it

pursues the fastest target G2. When k=35 and k=42, target G2 and G3 are caught by R2 and R1, respectively. Finally, G1 is caught at k=45 by R3, so the mission is completed successfully. The dot-lines are the planned trajectories of the robots and the colorful dots distinguish every step. From the figures we can see that the robots avoid the obstacles and catch all the moving targets successfully.

6. Conclusion

Path planning is a fundamental problem in robot application. In order to solve the path planning in dynamic environment, this chapter proposes a method based on LP/MILP to plan the acceleration of the robot in relative velocity coordinates. This method has the uniform formulation for 2D and 3D environment and it can give the information of the optimal velocity and acceleration in the view of the special cost function. Multiple constrains, such as the bounds of velocity, acceleration, and sensors, are included in the LP model and the MILP model. We also can add other constrains easily.

A particular application of this method is discussed for the problem of the multi-task planning where several robots are set to pursuit several targets. In the classical cooperation problem, the targets are assumed to be dynamic, similar to the TPOA problem. In the solution of MTP problem, a minimax assignment criterion and a global cost function are proposed to direct the task assignment.

Many simulations about the path planning in 2D/3D and in the multi-task planning requirements are taken to verify this novel method. The results show the low computing load of this method so it is potential to apply in real time manner.

7. Acknowledgment

This work was partially supported by Natural Science Foundation of China (NSFC) under grant #61035005 and #61075087, Hubei Provincial Education Department Foundation of China under grant #Q20111105, and Hubei Provincial Science and Technology Department Foundation of China under grant #2010CDA005.

8. References

Allaire, F. C. J.; Tarbouchi, M.; Labont´e G.; et al. (2009). FPGA implementation of genetic algorithm for UAV real-time path planning. *Journal of Intelligent and Robotic Systems*, Vol.54, No.1-3, pp.495-510, ISSN 09210296

Boyd S. & Vandenberghe L. (2004). Convex Optimization. Cambridge University Press, ISBN: 0521833787, New York

Chen, M.; Wu, Q. X. & Jiang, C. S. (2008). A modified ant optimization algorithm for path planning of UCAV. *Applied Soft Computing Journal*, Vol.8, No.4, pp.1712-1718, ISSN: 15684946

Chen, Y. & Han J. (2010). LP-Based Path Planning for Target Pursuit and Obstacle Avoidance in 3D Relative Coordinates. *Proceeding of American Control Conference*, pp.5394-5399, ISBN-13: 9781424474264, Baltimore, MD, USA, June 30- July 2, 2010

Chen, Yang; Zhao, Xingang; Zhang, Chan; et al. (2010). Relative coordination 3D trajectory generation based on the trimmed ACO. *Proceedings of International Conference on*

Electrical and Control Engineering, pp.1531-1536, ISBN-13: 9780769540313, Wuhan, China, June 26-28, 2010

Cocaud, C.; Jnifene, A. & Kim, B. (2008). Environment mapping using hybrid octree knowledge for UAV trajectory planning. *Canadian Journal of Remote Sensing*, Vol.34, No.4, pp.405-417, ISSN 07038992

David, A.; William, C.; Sanjeeb, D. & Monika, M. (2011). *QSopt Linear Programming Solver* [Online], June 1, 2011, available from: <http://www2.isye.gatech.edu/~wcook/qsopt/index.html>

Fahimi, F.; Nataraj, C & Ashrafiuon, H. (2009). Real-time obstacle avoidance for multiple mobile robots. *Robotica*, Vol.27, No.2, pp.189-198, ISSN 02635747

Fiorini, P. & Shiller, Z. (1998). Motion Planning in Dynamic Environments Using Velocity Obstacles. *International Journal of Robotics Research*, Vol. 17, No.7, pp.760-772. ISSN 02783649

Hrabar, S. E. (2006). Vision-based 3D navigation for an autonomous helicopter, PhD thesis, University of Southern California, United States

Jung, L. F.; Knutzon, J. S.; Oliver, J. H, et al. (2006). Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization, *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pp.992-1001, ISBN-10: 1563478234, Portsmouth, VA, USA, September 6-8, 2006

Kim, J. O. & Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, Vol.8, No.3, pp.338-349, ISSN 1042296X

Kitamura, Y.; Tanaka, T.; Kishino, F.; et al. (1995). 3-D path planning in a dynamic environment using an octree and an artificial potential field, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol.2, pp.474-481, Pittsburgh, PA, USA, August 5-9, 1995

Konolige, K. (2000). A gradient method for realtime robot control, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol.1, pp.639-646, Piscataway, NJ, USA, October 31-November 5, 2000

Krenzke, T. (2006). Ant Colony Optimization for agile motion planning. Master thesis. Massachusetts Institute of Technology, United States

Ladd, A. M. & Kavraki, L. E. (2004). Measure theoretic analysis of probabilistic path planning, *IEEE Transactions on Robotics and Automation*, Vol.20, No.2, pp.229-242, ISSN 1042296X

Liu, C.; Wei, Z. & Liu, C. (2007). A new algorithm for mobile robot obstacle avoidance based on hydrodynamics, *IEEE International Conference on Automation and Logistics*, pp.2310-2313, ISBN-10: 1424415314, Jinan, China, August 18-21, 2007

Nie, Y. Y.; Su, L. J. & Li, C. (2003). An Isometric Surface Method for Integer Linear Programming, *International Journal of Computer Mathematics*, Vol.80, No.7, pp.835-844, ISSN 00207160

Oh, T. S.; Shin, Y. S.; Yun, S. Y., et al. (2007). A feature information based VPH for local path planning with obstacle avoidance of the mobile robot, *4th International Conference on Mechatronics and Information Technology*. Vol.6794. Bellingham, WA, USA: SPIE

Richards, A. & How, J. P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming. *Proceeding of American Control Conference*, pp. 936-1941, ISSN 07431619, Anchorage, AK, USA, May 8-10, 2002

Schouwenaars, T.; How, J. & Feron, E. (2004). Receding Horizon Path Planning with Implicit Safety Guarantees. *Proceeding of the American Control Conference*, pp. 5576-5581, ISSN 07431619, Boston, MA, USA, June 30 - July 2, 2004

Schumacher, C. J.; Chandler, P. R.; Pachter, M.; et al. (2004). Constrained optimization for UAV task assignment, *Proceedings of the AIAA guidance, navigation, and control conference*, AIAA 2004-5352, Providence, RI., 2004

Shima, T.; Rasmussen, S. J.; Sparks, A. G.; et al. (2006). Multiple task assignments for cooperating uninhabited aerial Vehicles using genetic algorithms, *Computers & Operations Research*, Vol.33, No.11, pp.3252-3269, ISSN 03050548

Wang, Y.; Lane, D. M. & Falconer, G. J. (2000). Two novel approaches for unmanned underwater vehicle path planning: constrained optimisation and semi-infinite constrained optimization. *Robotica*, Vol.18, No.2, pp.123-142, ISSN 02635747

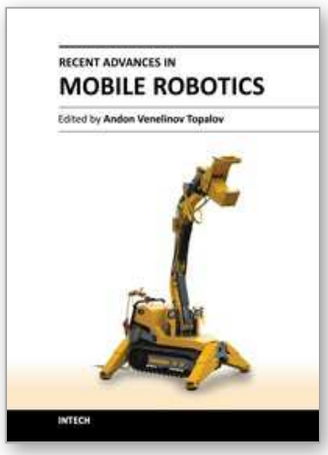
Yang, L. Y.; Nie, M. H.; Wu, Z. W.; et al. (2008). Modeling and Solution for Assignment Problem, *International Journal of Mathematical Models and Methods in Applied Sciences*, Vol. 2, No. 2, pp.205-212, ISSN: 1998-0140

Yang, L. Y.; Wu, C. D.; Han, J. D.; et al. (2009). The task assignment solution for multi-target pursuit problem. *International Conference on Computational Intelligence and Software Engineering*, pp.1–6, ISBN-13 9781424445073, Wuhan, China, December 11-13, 2009

Zhang, Y. & Valavanis K P. (1997). A 3-D potential panel method for robot motion planning. *Robotica*, Vol.15, No.4, pp.421-434, ISSN 02635747

Zhao, L. & Murthy, V. R. (2007). Optimal flight path planner for an unmanned helicopter by evolutionary algorithms, *AIAA Guidance, Navigation and Control Conference,* Vol.4, pp.3716-3739, ISBN-10: 1563479044, Hilton Head, SC, USA, August 20-23, 2007

Zu, D.; Han, J. & Tan, D. (2006). Acceleration Space LP for the Path Planning of Dynamic Target Pursuit and Obstacle Avoidance. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Vol.2, pp.9084-9088, ISBN-10 1424403324, Dalian, China, June 21-23, 2006



Recent Advances in Mobile Robotics

Edited by Dr. Andon Topalov

ISBN 978-953-307-909-7
Hard cover, 452 pages
Publisher InTech
Published online 14, December, 2011
Published in print edition December, 2011

Mobile robots are the focus of a great deal of current research in robotics. Mobile robotics is a young, multidisciplinary field involving knowledge from many areas, including electrical, electronic and mechanical engineering, computer, cognitive and social sciences. Being engaged in the design of automated systems, it lies at the intersection of artificial intelligence, computational vision, and robotics. Thanks to the numerous researchers sharing their goals, visions and results within the community, mobile robotics is becoming a very rich and stimulating area. The book Recent Advances in Mobile Robotics addresses the topic by integrating contributions from many researchers around the globe. It emphasizes the computational methods of programming mobile robots, rather than the methods of constructing the hardware. Its content reflects different complementary aspects of theory and practice, which have recently taken place. We believe that it will serve as a valuable handbook to those who work in research and development of mobile robots.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen