LP-based Optimal Path Planning in Acceleration Space

Di Zu\textsuperscript{1,2}  
2 Graduate school  
Chinese Academy of Science  
Beijing, China  
zudi@sia.cn

Jianda Han\textsuperscript{1} and Dalong Tan\textsuperscript{1}  
1 Robotics Laboratory, Chinese Academy of Science (CAS)  
Shenyang Institute of Automation  
Shenyang, China  
{jdan & dltan}@sia.cn

Abstract - This paper proposed Acceleration Space Linear Programming (ASLP) method for the path planning of target pursuit and obstacles avoidance (TPOA) problem in dynamic and uncertain environment. The acceleration space for TPOA problem is defined and LP algorithm is introduced into it. LP is a powerful tool for optimization but exhibits weakness when there exist nonlinear constraints or objective function. In the proposed acceleration space, the constraints and objective function of TPOA problems are inherently described as linear inequalities that can be directly solved by LP. Also, the complexity in obstacle contour does not increase the computational burden of the ASLP. Simulations are conducted with respect to the TPOA scenario including one moving target and multiple moving obstacles. The results obtained by ASLP are presented and compared with those of artificial potential guided evolution algorithm (APEA). Significant superior is demonstrated in many aspects by the comparison.

Index Terms - Path planning, linear programming, target pursuit and obstacle avoidance, dynamic environment.

I. INTRODUCTION

Path planning of mobile robot in dynamic and uncertain environments has been the subject of much recent interest. Various reported methods fall into two categories: global planning [1,2,6,7,8] and local planning [3,4,5]. Global planning techniques are conducted off-line and usually handle long-term planning in a known and static environment. This limits their use in dynamic and uncertain environments.

Local planning, on the other hand, can handle in real time both moving goals and obstacles in sensor-effective range. However, optimal performance criterion and other constraints, such as vehicle dynamics and kinematics are not easy to be integrated into the path planner. Artificial potential (AP) is one of the most popular methods for local path planning. It can easily solve the TPOA problem with single moving target and single obstacle in an unknown environment. But it is difficult to integrate either optimal criterion or dynamics constraints into AP-based methods. Therefore, the resulted trajectory may not be optimal or even unfeasible for the vehicle to track. Another disadvantage of AP is that it cannot handle target tracking and obstacle avoidance simultaneously or coordinate. Most recently, APEA was proposed with the purpose to overcome the above problems and achieve a fast and efficient trajectory searching mechanism for real-time installation. It has been used to search a semi-optimal trajectory for TPOA problem [9]. However, the path obtained by APEA is optimal only with respect to the searched area but not the complete space that can be perceived by sensors. Also, it cannot guarantee convergence although it develops much in convergence compared with normal EA method.

Linear programming (LP) is a general tool for optimization due to its modeling ability and also because that powerful LP solvers are available commercially [10,11]. Another advantage of LP is that performance criterion as well as dynamics constraints are easy to be integrated. The path obtained by LP is optimal with respect to the objective function while satisfying all the constraints defined in the problem. M. G. Earl [12] proposed an improved mixed integer LP (MILP) to decrease computational complexity in the static environment. However, it is still difficult for LP to be used as a real time path planner, and this is especially true if there exist multiple moving obstacles with irregular contour. Another difficulty in LP to solve the planning problem in dynamic and uncertain environments is the nonlinear terms in either the dynamics constraints or the objective function since they have to be linearized first to meet the LP requirement [12].

In this paper, an acceleration space is defined and LP method is proposed to be used in it. By using relative states [13] instead of absolute ones in the acceleration space, the nonlinear dynamics constraints and objective function within TPOA problems are inherently described as several linear inequalities that can be directly solved by LP. This method makes it possible that LP-based optimization can be used to solve the TPOA problem in real time, which is typically considered as a NP-hard problem in dynamic and uncertain environment. Also, the complexity in obstacle contour does not increase the computational burden of the ASLP.

II. OBSTACLE AVOIDANCE AND TARGET PURSUIT IN ACCELERATION SPACE

The TPOA problem can be described as: given a vehicle and a moving target, plan a trajectory that will allow the vehicle to catch the target while meeting the pre-designed optimal criterion and avoiding both static and moving obstacles in the environment. In this paper, without losing generality, we restrict our study to the 2-D case. The vehicle is considered as a mass point after expanding the obstacles and the target correspondingly [14].

Let \((X,Y)\) be the global coordinates, and a relative coordinates \((X',Y')\) is attached on the vehicle (Fig.1(a)). Let the X-axis of \((X',Y')\) be parallel to \(V_x\), where \(V_x\) is the velocity of the vehicle in \((X,Y)\). The arbitrary contour located
at O in Fig.1(a) denotes an obstacle with the velocity $\vec{V}_o$ in $(X,Y)$. The arbitrary contour at G in Fig.1(b) denotes the target with velocity $\vec{V}_g$ in $(X,Y)$ which the vehicle needs to catch. Then

$$\vec{V}_{ao} = \vec{V}_g - \vec{V}_o,$$

is the velocity of the vehicle relative to the obstacle, and

$$\vec{V}_{go} = \vec{V}_g - \vec{V}_o,$$

is the relative velocity of the vehicle to the target. All the angles in this paper are restricted to $[-\pi, \pi]$.

A. Obstacle Avoidance in Acceleration Space

With respect to obstacle avoidance issue in TPOA problem, the acceleration space includes vehicle, obstacles and the following five definitions (Fig.1(a)):

O-1) Line-of-Sight: The line between the vehicle and any observable point on the boundary of the obstacle, i.e., $\vec{L}_o$.

O-2) Collision-area: The area formed by all of $\vec{L}_o$.

O-3) Boundary-LOS: The two boundaries of collision area, i.e., $\vec{L}_{AO}$ and $\vec{L}_{AM}$ in Fig.1(a).

O-4) Collision-angle: The angle between $\vec{V}_{ao}$ and $\vec{L}_o$, denoted by $\gamma_{ao}$, the change of $\gamma_{ao}$ is denoted by $\Delta \gamma_{ao}$.

O-5) Avoidance-angle: The angle between $\vec{V}_{ao}$ and $\vec{L}_{AM}$, $\vec{L}_{AO}$, denoted by $\Delta \gamma_{ao_{\text{max}}}$ and $\Delta \gamma_{ao_{\text{min}}}$ respectively, while clockwise direction is assumed to be negative and $\Delta \gamma_{ao_{\text{max}}} \geq \Delta \gamma_{ao_{\text{min}}}$.

Fig.1(a) indicates that the obstacle avoidance issue in TPOA problem can be done by adjusting the collision angle, i.e., $\Delta \gamma_{ao}$. Assuming that there are $n$ obstacles to be avoided in the acceleration space of Fig.1(a) at the moment and $N = \{1,2,...,n\}$ is the set of obstacle index, we have the following obstacle-avoidance theorem.

Theorem-1: For each obstacle $O_i$, $i \in N$, it will be avoided if the collision angle is outside the collision area, i.e., one of the following two is met:

$$-\pi \leq \Delta \gamma_{ao} \leq \Delta \gamma_{ao_{\text{max}}},$$

$$\Delta \gamma_{ao_{\text{max}}} \leq \Delta \gamma_{ao} \leq \pi.$$

and the change of collision angle can be calculate as

$$\Delta \gamma_{ao} = -\frac{\sin \varphi_{ao}}{V_{ao}} \Delta V_a + \frac{V_{ao} \cos \varphi_{ao}}{V_{ao}} \Delta \alpha$$

(5)

where the subscript $i$ indicates the $i$th obstacle, $V_a$ and $V_{ao}$ are respectively, the norms of $\vec{V}_a$ and $\vec{V}_{ao}$, $\Delta V_a$ is the change of $V_a$, $\varphi_{ao}$ is the angle between $\vec{V}_a$ and $\vec{V}_{ao}$, $\Delta \alpha$ is the change of the direction angle of $\vec{V}_a$ in $(X,Y)$.

Proof:

The obstacle avoidance in Fig.1(a) is intuitonistic and needs no more proof. Followings are the proof of (5).

The direction angle of $\vec{V}_o$ and $\vec{L}_o$ are respectively denoted by $\theta_o$ and $\theta_o$. From Fig.1(a), we can see

$$\gamma_{ao} = \frac{V_o \sin (\alpha - \theta_o) - V_o \sin (\alpha - \theta_o)}{V_o \cos (\alpha - \theta_o)}$$

(6)

We assume

$$f(V_a, \alpha, V_o, \beta_o) = V_o \sin (\alpha - \theta_o) - V_o \sin (\alpha - \theta_o)$$

then

$$\gamma_{ao} = \sin^{-1} f(V_a, \alpha, V_o, \beta_o)$$

(7)

The differential of $f(V_a, \alpha, V_o, \beta_o)$ is

$$df = \frac{\partial f}{\partial V_a} dV_a + \frac{\partial f}{\partial \alpha} d\alpha + \frac{\partial f}{\partial V_o} dV_o + \frac{\partial f}{\partial \beta_o} d\beta_o$$

(8)

The velocity of the obstacle is assumed to be uniform in the short period of a step, and then (9) is approximate to

$$df = \frac{\partial f}{\partial V_a} dV_a + \frac{\partial f}{\partial \alpha} d\alpha$$

(10)

From (6) and (7), we can obtain

$$\frac{\partial f}{\partial V_a} = \frac{-V_o \sin (\alpha - \theta_o)}{V_o \cos (\alpha - \theta_o)}$$

(11)

$$\frac{\partial f}{\partial \alpha} = \frac{V_o \cos (\alpha - \theta_o)}{V_o \cos (\alpha - \theta_o)}$$

(12)

Substitute (11) and (12) into (10), we have

$$df = \frac{-V_o \sin (\alpha - \theta_o) + V_o \cos (\alpha - \theta_o)}{V_o \cos (\alpha - \theta_o)} d\alpha$$

(13)

From (8), we can see the differential of $\gamma_{ao}$ is

$$df_{\gamma_{ao}} = \frac{-V_o \sin (\alpha - \theta_o)}{1 + \frac{V_o \cos (\alpha - \theta_o)}{V_o \sin (\alpha - \theta_o)}}$$

(14)

Substitute (6), (7) and (13) into (14), then

$$df_{\gamma_{ao}} = -V_o \sin (\alpha - \beta_o) d\alpha + V_o \left[ V_o - V_o \cos (\alpha - \theta_o) \right] d\alpha$$

(15)

Within each short time-step of dynamic path planning, (15) can be approximated as

$$df_{\gamma_{ao}} = -V_o \sin (\alpha - \beta_o) \Delta \alpha + V_o \left[ V_o - V_o \cos (\alpha - \theta_o) \right] \Delta \alpha$$

(16)

From Fig.2(a), the following relationships are obtained

$$V_o \sin (\alpha - \beta_o) = V_{ao} \sin \varphi_{ao}$$

(17)
\[ V_A - V_O \cos(\alpha - \beta_O) = V_{AO} \cos \theta_{AO} \]  
(18)
\[ V_A^2 + V_O^2 - 2V_A V_O \cos(\alpha - \beta_O) = V_{AO}^2 \]  
(19)
Substitute the relationships above into (16), we can get (5).

**End Proof.**

From (5), we can see that the change of collision angle is dependent on \( \Delta \alpha \) and \( \Delta V_A \), that can be calculated via (3), (4) and (5). The problem left is how to design \( \Delta \alpha \) and \( \Delta V_A \) to optimally meet a criterion besides the constraints of (3) or (4).

Theorem-1 also shows that obstacle avoidance problem has been described as a set of linear inequalities, so it can be used by LP directly. Moreover, (3) and (4) demonstrates that the conditions for avoiding obstacles are only dependent on the collision area in the acceleration spaces and have no relationship with the obstacle contour. So the complexity of obstacle contour will not increase the computational burden of LP-based optimization.

**B. Target Pursuit in Acceleration Space**

Similar to the analyses of obstacle avoidance, there is also a target-pursuit-related acceleration space, which includes the following four definitions (Fig.1(b)):  

**T-1** Line-of-Sight: The line between the vehicle and any observable point on the boundary of the target, i.e., \( \tilde{L}_G \).

**T-2** Pursuit-area: The area formed by all of \( \tilde{L}_G \).

**T-3** Boundary-LOS: The two boundaries of Pursuit area, i.e., \( \tilde{L}_{AY}_G \) and \( \tilde{L}_{AM}_G \) in Fig.1(b).

**T-4** Catch-angle: The angle between \( \tilde{V}_{AG} \) and \( \tilde{L}_G \), denoted by \( \gamma_{AG} \); the change of \( \gamma_{AG} \) is denoted by \( \Delta \gamma_{AG} \).

Giving a target denoted by \( G \), the path planning for target pursuit issue in the TPOA problem can be realized by adjusting the relative velocity \( \tilde{V}_{AG} \), which includes both its norm value \( V_{AO} \) and its direction \( \gamma_{AG} \). For target pursuit, we have the following theorem.

**Theorem-2**: For the target of \( G \) in Fig.1(b), it will be pursued if the catch angle is inside the pursuit area, i.e., the following equation is met:

\[ \Delta \gamma_{AG} \leq \Delta \gamma_{AG_{max}} \leq \Delta \gamma_{AG_{min}} \]  
(20)
and the change of catch angle can be calculated as

\[ \Delta \gamma_{AG} = -\frac{\sin \theta_{AG} \Delta V_A}{V_{AG}} + \frac{V_{AO} \cos \theta_{AG} \Delta \alpha}{V_{AG}} \]  
(21)
where \( V_A \) and \( V_{AO} \) are respectively the norm of \( \tilde{V}_A \) and \( \tilde{V}_{AO} \), \( \Delta V_A \) is the change of \( V_A \), \( \theta_{AG} \) is the angle between \( \tilde{V}_A \) and \( \tilde{V}_{AO} \), \( \Delta \alpha \) is the change of the direction angle of \( \tilde{V}_A \).

**Proof**: Omitted due to the similarity with that of theorem-1.

**III. OPTIMIZATION IN ACCELERATION SPACE**

The optimization of the path planning for target pursuit issue is proposed as two divisions: optimization with respect to pursuit convergence and that with respect to pursuit time.

**A. Optimization with Respect to Target Convergence**

Equation (21) of Theorem-2 shows that target pursuit can be realized by adjusting \( \tilde{V}_{AG} \), i.e., both \( \Delta \alpha \) and \( \Delta V_A \), and it can be rewritten as:

\[ \gamma_{AG_{max}} = \gamma_{AG_{current}} - \frac{\sin \theta_{AG} \Delta V_A + V_{AO} \cos \theta_{AG} \Delta \alpha}{V_{AG}} \]  
(22)

From the definition of (T-4) and Fig.1(b), we can see that \( \gamma_{AV} \) is the angle between \( \tilde{V}_{AG} \) and \( \tilde{L}_G \). For the purpose of target pursuit, it is desired that \( \gamma_{AG_{max}} \) is approaching zero so that the vehicle will always 'point' the target. Therefore, we propose to design the target convergence criterion as:

\[ J_{G_{1}} = \gamma_{AG} - \frac{\sin \theta_{AG} \Delta V_A + V_{AO} \cos \theta_{AG} \Delta \alpha}{V_{AG}} \]  
(23)

The path planning is to minimize \( J_{G_{1}} \) by adjusting the direction of \( \tilde{V}_{AG} \), i.e., \( \Delta \alpha \) and \( \Delta V_A \).

By defining a positive variable \( z \) as [12]

\[ -z \leq \gamma_{AG} - \frac{\sin \theta_{AG} \Delta V_A + V_{AO} \cos \theta_{AG} \Delta \alpha}{V_{AG}} \leq z \]  
(24)
Then, minimizing \( z \) subject to the inequality of (24) is equivalent to minimize \( J_{G_{1}} \). And the objective function of (23) can be written as a linear one

\[ J_{G_{1}} = z \]  
(25)
where, \( z \geq 0 \).

Thus, the optimal path planning for target convergence can be described as: to find \( \Delta \alpha \) and \( \Delta V_A \), while subjecting to the constraint of (24) and minimizing (25).

**B. Optimization with Respect to Pursuit Time**

For the optimization of pursuit time, we have the following theorem.

**Theorem-3**: By assuming that the vehicle velocity relative to the target is pointing to the target, i.e., \( \gamma_{AG_{max}} \) has been minimizing, then the minimum pursuit time is achieved when the vehicle pursues with the maximum acceleration relative to the target, i.e. \( \Delta V_{AO} \rightarrow \Delta V_{AG_{max}} \), and

\[ \Delta V_{AG} = \cos \theta_{AO} \Delta V_A + V_{AO} \sin \theta_{AG} \Delta \alpha \]  
(26)
where \( \Delta V_{AG_{max}} \) is the maximum velocity change of \( \Delta V_{AG} \).

**Proof**: If \( \Delta V_{AG} \rightarrow \Delta V_{AG_{max}} \), then the vehicle-target relative velocity will approach its maximum as quickly as the vehicle can. So, the pursuit time will be minimum.

Let \( \beta_G \) denote the direction of \( \tilde{V}_{AG} \), and

\[ f = V_{AO}^2 = V_A^2 + V_O^2 - 2V_A V_O \cos(\alpha - \beta_G) \]  
(27)
then
\[ \frac{df}{dV_A} = 2V_A - 2V_o \cos(\alpha - \beta_o) \]  
\[ \frac{df}{d\alpha} = 2V_V \sin(\alpha - \beta_o) \]  
By combining (27), (28) and (29), we have:
\[ df = \left[ 2V_A - 2V_o \cos(\alpha - \beta_o) \right] dV_A + 2V_V \sin(\alpha - \beta_o) d\alpha \]  
Differentiating (27) also gives:
\[ dV_A = \frac{1}{2} \frac{df}{dV_A} dV_A + \frac{1}{2} \frac{df}{dV_V} dV_V + \frac{1}{2} \frac{df}{d\alpha} d\alpha \]  
From Fig.2(b), we have
\[ V_A sin(\alpha - \beta_o) = V_{ag} sin(\phi_{ag}) \]  
\[ V_A - V_o \cos(\alpha - \beta_o) = V_{ag} \cos(\phi_{ag}) \]  
\[ V^2_A + V^2_o - 2V_A V_o \cos(\alpha - \beta_o) = V_{ag}^2 \]  
Substitute (32), (33) and (34) into (31), we have
\[ dV_A = V_{ag} \cos(\phi_{ag}) dV_A + V_{ag} sin(\phi_{ag}) d\alpha \]  
\[ \text{End Proof.} \]
From (21) and (26), we have
\[ \Delta V^2_A + \Delta (V_A \Delta \alpha)^2 = (V_{ag} \Delta \alpha_{ag})^2 + \Delta V^2_v \]  
Substitute the maximum of \( \Delta V_A \) and \( \Delta \alpha \) to (36), we can get
\[ \Delta V^2_{ag_{max}} = (\Delta V^2_{max} + \Delta \alpha^2_{ag_{max}})^2 - (V_{ag} \Delta \alpha_{ag})^2 \]  
Therefore, the optimal pursuit time criterion is designed as:
\[ J_{g2} = \frac{\Delta V^2_{ag_{max}}}{V_{ag}} - \frac{\Delta \alpha^2_{ag_{max}}}{V_{ag}} \Delta V_A - \frac{V_{ag}}{V_{ag}} \Delta \alpha \]  
Because \( \Delta V^2_{ag_{max}} \) is the maximum of \( \Delta V^2_{ag} \), it is obviously that \( J_{g2} \geq 0 \).
Thus, the path planning for minimum pursuit time can be described as: to find \( \Delta \alpha \) and \( \Delta V_A \), while minimizing (38).
From above, the target pursuit and obstacle avoidance is actually realized by adjusting \( \Delta V_A \) and \( \Delta \alpha \) in each planning period. As the change in the norm of \( \vec{v}_A \), \( V_A \) can be regarded as the component of the vehicle acceleration in the direction of \( \vec{v}_A \). As the change in the direction of \( \vec{v}_A \), \( V_A \Delta \alpha \) is close to the component of the acceleration perpendicular to \( \vec{v}_A \). Therefore, these two components determine the trajectory of the vehicle in the acceleration space.

IV. LP METHOD IN ACCELERATION SPACE

Linear programming is a powerful tool to solve constrained optimization problems. For the path planning of TPOA in dynamic environment, the planner has to evolutionarily generate optimal trajectory step by step, and within each time-step, the optimization can be done by LP method. The acceleration space linear programming (ASLP) based path planning method can be described as following:

To minimize: objective function
Subject to:
1) Constraints of the dynamic environment including both target pursuit and obstacle avoidance;
2) Constraints of the vehicle including the dynamics and kinematics limitations.

A. Objective function for LP Searching
The objective function for LP searching is proposed as:
\[ J_{LP} = \omega_1 J_{g1} + \omega_2 J_{g2} \]  
where \( J_{g1} \) and \( J_{g2} \) are the criterion of target convergence and pursuit time defined by (23) and (38), and \( \omega_1 \geq 0 \) and \( \omega_2 \geq 0 \) are, respectively, the weight of \( J_{g1} \) and \( J_{g2} \).

B. Constraints of Vehicle
Besides those for obstacle avoidance, the constraints due to vehicle dynamics and kinematics have also to be considered. The dynamics constraints, i.e., the limitation of \( \Delta V_A \), are described for LP method as:
\[ \Delta V_{A_{min}} \leq \Delta V_A \leq \Delta V_{A_{max}} \]  
where \( \Delta V_{A_{min}} \) and \( \Delta V_{A_{max}} \) are the maximum and minimum velocity change of the vehicle within one time step (also known as maximum and minimum acceleration). The kinematics constraints include
\[ V_{A_{min}} \leq V_A \leq V_{A_{max}} \]  
\[ \Delta \alpha_{min} \leq \Delta \alpha \leq \Delta \alpha_{max} \]  
where \( V_{A_{min}} \) and \( V_{A_{max}} \) are the maximum and minimum vehicle velocity. \( \Delta \alpha_{max} \) and \( \Delta \alpha_{min} \) are, respectively, the maximum and the minimum change of the vehicle moving direction within one time step.
Combining (40) and (41), we have
\[ \max(-\Delta V_{A_{max}} V_{A_{max}} - V_A) \leq \Delta V_A \leq \min(\Delta V_{A_{min}} V_{A_{min}} - V_A) \]  
where, the function \( \max(\cdot) \) and \( \min(\cdot) \) respectively return the largest and the smallest element in the bracket.

C. LP Method for TPOA Problem
The TPOA problem is a dynamic process and the states of the target and the obstacles are changing all the time. The path planner is to generate the next step trajectory of the vehicle that is optimal with respect to current snap shot. In other words, only an optimal pair of \( (\Delta V_A, \Delta \alpha) \) is needed in each time step.
We have already defined the objective function of (39), the vehicle-related constraints of (42) and (43), as well as the obstacle avoidance constraint of (3) or (4), which are all in linear form. So the LP method can be used to search an optimal \( (\Delta V_A, \Delta \alpha) \). However, we should note that the obstacle avoidance constraint of (3) or (4) is an ‘OR’ logic instead of a ‘AND’ one, which cannot be handled directly by LP method. To figure out this problem, we propose the following solution.
Let \( S = \{s_1, s_2, s_3, \ldots, s_n\} \) be the indices of all the subset of \( N \), where \( N = \{1, 2, \ldots, n\} \) is the set of obstacle index in case there are \( n \) obstacles in the environment.
For \( j \in S \), let \( N_j \) be one of the subset of \( N \), and \( M_j = N - N_j \) be the complement set of \( N_j \), then the
A. Efficiency

For pursuit of the acceleration relative to the environment, the simulation is defined as:

\[ J_j = \alpha_1 J_{G1} + \alpha_2 J_{G2} \]

Subjecting to

\[ \begin{align*}
    & J_{G1} - \Delta V_j \sin \varphi_{A0} + V_j \Delta \alpha \leq \Delta Y_j \\
    & J_{G2} - \Delta V_j \sin \varphi_{A0} + V_j \Delta \alpha \geq -\Delta Y_j \\
    & \Delta \alpha \geq \pi \\
    & \Delta \alpha \leq \Delta \alpha_{max} \\
    & \Delta V_j \geq \max(-\Delta V_{Amin}, V_{Amin} - V_j) \\
    & \Delta V_j \leq \min(\Delta V_{Amax}, V_{Amax} - V_j) \\
    & -\sin \varphi_{A0} \Delta V_j + \frac{V_j \cos \varphi_{A0}}{V_{A0}} \Delta \alpha \leq \Delta Y_{A0min} i \in N_j \\
    & -\sin \varphi_{A0} \Delta V_j + \frac{V_j \cos \varphi_{A0}}{V_{A0}} \Delta \alpha \geq -\pi \\
    & -\sin \varphi_{A0} \Delta V_j + \frac{V_j \cos \varphi_{A0}}{V_{A0}} \Delta \alpha \geq \pi i \in N_j \\
    & -\sin \varphi_{A0} \Delta V_j + \frac{V_j \cos \varphi_{A0}}{V_{A0}} \Delta \alpha \leq \pi
\end{align*} \]

End of loop

The optimal trajectory is the one of \( \Delta V_j, \Delta \alpha \) that satisfy \( J = \min(J_j) \). The computation complexity of this algorithm is \( 2^n \), where \( n \) is the number of the obstacle that must be considered in one time step. We will show in the next section that for \( n=3 \), the computation time is only 3.3ms, which indicates that it is fast enough for real-time application.

V. SIMULATION RESULTS

The simulations demonstrated here include one vehicle, one moving target and multiple moving obstacles in 2-D environment. The current position and velocity of the vehicle relative to the target and the obstacles are assumed to be known by sensors or some estimation techniques. Any next-step information of the target or the obstacles is totally unknown to the vehicle.

It should be noted that the proposed ASLP algorithm can pursue the target while avoiding multiple obstacles with any contour. But, in order to highlight the efficiency of the algorithm for obstacles with nonlinear boundary, also to compare the method with other path planning algorithms, the obstacles in the simulation are described by circles. More complex contour can be handled in the same way because in the acceleration space, the obstacle constraints are independent on obstacle contour.

A. Efficiency of ASLP

To show the efficiency of this algorithm, we illustrate a TPOA scenario with two moving obstacles and one static obstacle in different size (see Fig.3). The parameters in the simulation are set as follows:

\[
\begin{align*}
    \Delta V_{Amax} &= 2cm; \quad \Delta V_{Ymax} = 2cm; \\
    V_{Amax} &= 99cm/s; \quad V_{Vmax} = 0cm/s; \\
    V_{A(t_x=0)} &= 65cm/s; \quad V_{V(t_x=0)} = 6cm/s; \\
    \alpha_{Amax} &= 0.5rad/s; \quad \alpha_{Vmax} = 0.5rad/s;
\end{align*}
\]

where \( r_{a1}, r_{a2} \) and \( r_o \) are respectively the radius of three obstacles, \( ta \) is the radius of the target. \( V_{A(t_x=0)} \) and \( V_{V(t_x=0)} \) are the initial velocities of the vehicle in \( (X, Y) \).

The algorithm is run on the computer with the CPU of PentiumIV/2.40GHz processor. The LP is solved by the QSopt function library [11], which is embedded in C language. The computation time for path searching within one time step is about 3.3ms and the resulted trajectory is shown as Fig.3. The blue circles represent the obstacle position at the snap shot while the vehicle is trying to avoid collision. The circles, noted by ‘S’ and ‘E’, respectively represent the initial and the final positions of the obstacles and the target. The line is the planned trajectory of the vehicle and the alternative black and magenta segment distinguish every step. From this figure we can see that the vehicle avoids the obstacles and catches the moving target successfully. The optimization performance is demonstrated that the obstacle avoidance procedure is along the boundary of the obstacle with out any waste and, after the avoidance, the vehicle moves to the target along a strict line.

B. Performance Comparison Between ASLP and APEA

The performance of ASLP is compared with that of APEA with the same TOPA scenario. Without losing generality, the trajectories of the target and the obstacles are designed as:

\[
\begin{align*}
    x_{target}(cm) &= 800 + 3 \cdot t_g \\
    y_{target}(cm) &= 800 + 5 \cdot t_g
\end{align*}
\]

where \( x_{target} = x_{target} - 200; \quad y_{target} = y_{target} - 200 \);

\[
\begin{align*}
    x_{obstacle-1}(cm) &= x_{target} - 200; \quad y_{obstacle-1}(cm) = y_{target} - 200; \\
    x_{obstacle-2}(cm) &= x_{target} - 200; \quad y_{obstacle-2}(cm) = y_{target} - 200; \\
    x_{obstacle-3}(cm) &= x_{target} - 200; \quad y_{obstacle-3}(cm) = y_{target} - 200;
\end{align*}
\]

where \( (x, y) \) is the position, \( t_g \) is the index of the time step.

The kinematics and dynamics constraints of the vehicle, target and obstacles are the same as the simulation \( A \), but their sizes are selected as \( r_{a1} = r_{a2} = r_o = 50cm \).

![Fig. 3 Path planning for TPOA by ASLP method](image-url)
The trajectory generated by ASLP is shown in Fig. 4, and Fig. 5 is the comparison between ASLP and APEA. The ASLP method will always give out a unique path but APEA method is dependent on the offspring times of the evolutionary algorithm. So, in Fig. 5, the ASLP result in red is kept unchanged despite the offspring times. The APEA result in blue, on the other hand, is changing with respect to different offspring times.

From Fig. 5, we can see the followings:
1) Convergence (Fig. 5(a)): With the increase of the offspring times, the convergence of APEA is becoming better. Almost all the trails convergence when the offspring time reaches 16. ASLP is convergence as long as the constraint inequalities can be satisfied simultaneously. That is to say, the vehicle will catch the target successfully if there exists a free path to the target.
2) Catching time (Fig. 5(b)): Catching time was reduced about 6–8 steps by ASLP compared with APEA.
3) Path length (Fig. 5(c)): Path length reduced 120–200 cm by ASLP compared with APEA.
4) Minimum obstacle distance (Fig. 5(d)): The ASLP reduces the entire path wasted by APEA for trying to avoid the obstacle. Actually, these two methods adopt completely different strategies to avoid obstacles. ASLP chooses the path close to the obstacle, while APEA choose the path close to the central line between two obstacles.

VI. CONCLUSION

The acceleration space is proposed with respect to TPOA problem. In such an acceleration space, the obstacle constraints and objective function of TOPA problems can be inherently described as linear inequalities. The LP method is therefore introduced into the acceleration to solve the optimal path-planning problem under the linear objective function and constraints. Another advantage of ASLP is that its computation complexity is independent to the contour of obstacle or target. Both analytical proof and simulation demonstrate the validity and effectiveness of the proposed ASLP method. The comparison between ASLP and APEA clearly demonstrates the improvements of ASLP in the issues of convergence, catching time, path length as well as the minimum obstacle distance. The simulation also indicates that the execution time for the ASLP method to solve 3-obstacle TPOA problem is only 3.3 ms, which is a reasonable sample period for real-time implementation on actual system.

REFERENCES

[1] M. G. Earl, R. D’Andrea, “Modeling and control of a multi-agent system using mixed integer linear programming,” Proceedings of the 41st IEEE International Conference on Decision and Control, Las Vegas, Nevada USA, Dec. 2002, pp. 107-111.
[2] L. Kavraki, M. Kiossoulakis and J. Latombe, “Analysis of probabilistic roadmaps for path planning,” IEEE Trans. Robotics and Automation, vol. 14, no. 1, pp. 166-171, 1998.
[3] J. Kim, P. Hsosla, “Real-time obstacle avoidance using harmonic potential functions,” IEEE Trans. Robotics and Automation, vol. 8, no. 3, pp. 338-349, 1992.
[4] H. Feder and J. Slotine “Real-time path planning using harmonic potentials in dynamic environments,” Proc. IEEE Int. Conf. Robotics and Automation, pp. 874-881, 1997.
[5] S. Waydo, R. M. Murry “Vehicle motion planning using stream functions,” Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taiwan, Sep. 2003, pp. 14-19.
[6] J. Xiao, Z. Michalewicz, L. Zhang and K. Trojanowski, “Adaptive evolutionary planner/navigator for mobile robots,” IEEE Trans. Evolutionary Computation, vol. 1, no. 1, pp. 18-28, 1997.
[7] B. Capozzi and J. Vagners, “Evolving (semi)-autonomous vehicles,” AIAA Guidance, Navigation and Control Conference and Exhibit Montreal, Canada, Aug. 2003.
[8] D. Ratham and B. Capozzi, “Evolutionary approaches to path planning through uncertain environments,” AIAA’s 1st Unmanned Aircraft Vehicles, Systems, Technologies and Operations Conference and Workshop, Portsmouth, Maine, May 2002.
[9] J. D. Han and Mark Campbell, “Artificial Potential Guided Evolutionary Path Plan for Target Pursuit and Obstacle Avoidance,” AIAA Guidance, Navigation and Control Conference and Exhibit Austin, Aug. 2003.
[10] ILOG AMPL CPLEX System Version 7.0 User’s Guide, 2000.
[11] http://www2.isye.gatech.edu/~wcook/qsopt/index.html
[12] M. G. Earl and R. D’Andrea, “Solving minimum time problems and obstacle avoidance problems using improved MILP methods”, IEEE Conference on Decision and Control, Dec. 2004.
[13] P. Fiorini and Z. Shihter, “Motion Planning in dynamic environments using the relative velocity paradigm,” Proceedings of the 1993 IEEE International Conference IEEE on Robotics and Automation, Atlanta, May 1993, pp. 550-556.
[14] Feng Zhang, Dulong Tan and Zhenwei Wu, “Multiple Obstacles Avoidance For Mobile Robot In Unstructured Environments,” Proceedings of the 2004 IEEE International Conference on Robotics, Automation and Mechatronics, Singapore, Dec. 2004
[15] Feng Zhang and Dulong Tan, “Obstacle Avoidance for Mobile Robots Based on Relative Coordinates,” Proc. IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing, Changsha, Oct. 2003