Vision-Based Avoidance of Obstacles with Unknown Constant Velocity

Lili Ma

Abstract—This paper discusses vision-based avoidance of obstacles with unknown sizes and unknown constant velocities for four-wheel nonholonomic robots. The robot is equipped with an inertial measurement unit that provides measurements of the robot’s position and velocity, and a monocular camera that detects the obstacles. An analytical scheme is adopted for trajectory generation and vehicle control. Vision sensor is used to obtain the estimates of the obstacles’ unknown sizes and unknown velocities that are used by the trajectory generation scheme. The effectiveness of the proposed technique is demonstrated by numerical simulation examples.

Key Words: Trajectory generation and replanning, obstacle avoidance, vision-based estimation, fast estimator.

I. INTRODUCTION

With the increase of mobile robots in military and commercial applications in recent years, vision sensors have been widely utilized for exploring dynamic environment. Vision sensors have been preferred over range finders, because they are cheaper and provide richer information about the environment. Moreover, they are passive sensors, so they do not interfere with other sensors and do not pose safety concerns in populated environment [1]. A variety of theoretical and experimental results have been reported pertaining to the usage of vision sensors in navigation, servoing, interception (approaching a moving object until collision), and tracking (approaching a moving object while matching its location and velocity), to name but a few [2]–[12].

In this paper, an analytical trajectory generation controller is augmented with an obstacle avoidance algorithm such that the robot avoids dynamic obstacles using visual information collected from an onboard camera. The nonholonomic vehicle is a four-wheel robot that has two identical parallel, non-deformable rear wheels and two steering front wheels. A trajectory generation controller is implemented that analytically solve the problem of real-time trajectory planning and replanning for nonholonomic mobile robots operating in an environment of multiple dynamically moving obstacles [13].

For obstacles moving with known constant velocities, velocity planning can be done using the velocity obstacle concept [14], [15]. It was reported in [16] that the velocity obstacle concept is advantageous in handling dynamic obstacles than other obstacle avoidance schemes, such as the dynamic window approach [17], [18] and the curvature-velocity method [19].

Assume that a conventional pin-hole camera is used and the camera is calibrated beforehand. Further, some image processing algorithms are available to extract the visual measurements. The visual information collected by the camera is processed by an estimator that provides the inputs to the trajectory generation algorithm. These inputs include estimation of the obstacle’s unknown sizes and unknown velocities, obtained via an identifier-based observer (IBO) [20].

The paper is organized as follows. Section II reviews the analytical trajectory generation algorithm with obstacle avoidance for a nonholonomic robot. Estimation of the obstacles’ unknown sizes and unknown velocities is presented in Sec. III. Simulation results are shown in Sec. IV. Finally, Section V concludes the paper.

II. REVIEW OF TRAJECTORY GENERATION AND OBSTACLE AVOIDANCE

Consider a car-like robot in Fig. 1. The two front wheels are steering wheels. Its rear wheels are driving wheels but have a fixed orientation. The distance between the two wheel-axle centers is $l$. The midpoint along the line connecting the axle centers is set to be the guide point. The whole vehicle is physically within a circle of radius and centered at the guide point. Trajectory planning will be done for the guide point [13].

![Fig. 1. A four-wheel car-like robot.](image)

Following the notations in [13], let the generalized coordinates be $q(t) = [x(t), y(t), \theta(t), \phi(t)]^T$, where $(x(t), y(t))$ are the Cartesian coordinates of the guide point, $\theta(t)$ is the
orientation of the robot with respect to the x-axis, and \( \phi(t) \) is the steering angle. Let \( \rho \) be the radius of the (back) driving wheels, \( u_1(t) \) be the angular velocity of the driving wheels, and \( u_2(t) \) be the steering rate of the (front) guiding wheels. The following kinematic model can be obtained for the four-wheel robot [13]:

\[
\begin{bmatrix}
    \dot{x}(t) \\
    \dot{y}(t) \\
    \dot{\theta}(t) \\
    \dot{\phi}(t)
\end{bmatrix} = \begin{bmatrix}
    \rho \cos(\theta(t)) - \frac{\rho}{2} \tan(\phi(t)) \sin(\theta(t)) & 0 \\
    \rho \sin(\theta(t)) + \frac{\rho}{2} \tan(\phi(t)) \cos(\theta(t)) & 0 \\
    \frac{\rho}{2} \tan(\phi(t)) \cos(\theta(t)) & 0 \\
    0 & 1
\end{bmatrix} \begin{bmatrix}
    u_1(t) \\
    u_2(t)
\end{bmatrix}.
\]

(1)

In order to standardize the process of analytically solving the trajectory-planning problem, the kinematic model in (1) was transformed into a canonical form, called the chained form. With the standard results on the chained form, one can obtain the following transformations of coordinates and inputs [13]:

\[
\begin{align*}
    z_1(t) &= x(t) - \frac{1}{2} \cos(\theta(t)), \\
    z_2(t) &= \frac{\tan(\phi(t))}{l \cos^3(\theta(t))}, \\
    z_3(t) &= \tan(\theta(t)), \\
    z_4(t) &= y(t) - \frac{1}{2} \sin(\theta(t)), \\
    u_1(t) &= \frac{v_{c1}(t)}{\rho \cos(\theta(t))}, \\
    u_2(t) &= -\frac{3 \sin(\theta(t))}{l \cos^2(\theta(t))} \sin^2(\phi(t)) v_{c1}(t) + l \cos^3(\theta(t)) \cos^2(\phi(t)) v_{c2}(t).
\end{align*}
\]

(2)

Under the above transformations, the kinematic model in (1) was mapped into the following two-input four-state chained form [13]:

\[
\begin{align*}
    \dot{z}_1(t) &= v_{c1}(t), \\
    \dot{z}_2(t) &= v_{c2}(t), \\
    \dot{z}_3(t) &= \dot{z}_2(t) v_{c1}(t), \\
    \dot{z}_4(t) &= z_3(t) v_{c1}(t).
\end{align*}
\]

(3)

Trajectory generation and vehicle control are with respect to the above chained form and later transformed to the Cartesian coordinates.

Assuming that the obstacles move with unknown but constant velocities, the velocity obstacle concept in [14], [15] can be applied to solve the problem of obstacle avoidance. Consider Fig. 2 where two objects are moving with constant velocities \( v_i = [v_{i,x}, v_{i,y}]^\top \) for \( i = 1, 2 \) and the objects are enlarged by the radius of the robot. For an arbitrary robot velocity \( v_r(t) = [\dot{x}(t), \dot{y}(t)]^\top \), the relative velocities \( v_{r,i}(t) = v_r(t) - v_i \) for \( i = 1, 2 \) can be determined. It can be verified that if the relative velocities do not enter the two velocity cones (that are corresponding to and are pointing toward the objects), collision will never happen. For the scenario shown in Fig. 2, since \( v_{r,1}(t) \) is in the cone of object 1 and \( v_{r,2}(t) \) is not in the cone of object 2, the robot will collide with object 1 but not with object 2, if all velocities are maintained. In summary, velocity planning of \( v_r(t) \) should be done in a way such that \( v_{r,i}(t) \) are not in their corresponding cones [13], [14].

For constant obstacle velocities, the collision avoidance criterion that applies the velocity concept to the chained model in (3) can be:

\[
(z_1 - v_{i,x} \tau - x_i)^2 + (z_4 - v_{i,y} \tau - y_i)^2 \geq (r_i + R + \frac{l}{2})^2,
\]

(4)

for \( \tau \in [0, t_f] \), where \( t_f \) denotes the total amount of time taken to reach the goal location. Please refer to [13] for more details of the trajectory generation algorithm with obstacle avoidance.

### III. Vision-Based Estimation of Obstacle’s Unknown Size and Velocity

The trajectory generation and obstacle avoidance algorithm in [13] requires that the obstacles’ sizes and velocities are known. When these information are unknown, vision sensors can be used to obtain their estimates. In this section, vision-based estimation is formulated in a 2D setting (Fig. 3). It is assumed that a conventional pin-hole model for the camera is used and the camera is calibrated beforehand. Further, the obstacle is always within the vehicle’s sensor range and some image processing algorithms are available to extract the subtended angle \( \alpha(t) \) and the bearing angle \( \beta(t) \) from the image pixel coordinates. The objective is to estimate the obstacle’s unknown size and unknown velocity using the
visual measurements \( \alpha(t) \) and \( \beta(t) \), as well as information from the vehicle’s onboard sensors. In case of multiple obstacles, it is further assumed that the image processing algorithm is able to associate the visual measurements of each obstacle. For simplicity, the index \( i \) for the \( i^{th} \) obstacle is dropped.

\[ \hat{\eta}(t) = \begin{bmatrix} \hat{\eta}_1(t) \\ \hat{\eta}_2(t) \\ \hat{\eta}_3(t) \end{bmatrix}. \tag{14} \]

Similar to the estimation formulation in [12], consider the motion of the vehicle and an obstacle in a 2D Cartesian space \((X, Y)\). Let \( z(t) = [z_x(t), z_y(t)]^\top \) be the vector of relative distance between the vehicle and the obstacle. Notice that the variable \( z(t) \) is different from those variables in equations (2) and (3) for the transformed coordinates. In the kinematic setting, the relative dynamics can be described by:

\[ \begin{bmatrix} \ddot{z}_x(t) \\ \ddot{z}_y(t) \end{bmatrix} = \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} - \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix}, \tag{5} \]

where \( z_x(t), z_y(t) \) are the Cartesian coordinates of the relative vehicle-obstacle dynamics, \( v_x(t), v_y(t) \) are the components of the velocity vector of the obstacle in the \( X \) and \( Y \) axis, respectively, and \( x(t), y(t) \) are the Cartesian coordinates of the vehicle. The vehicle has no knowledge of the obstacle, including its velocity components \( v_x(t), v_y(t) \) and its characteristic length \( L \), except maybe for some conservative upper and lower bounds. The quantities \( z_x(t), z_y(t), v_x(t), v_y(t), x(t), y(t) \) are related to the inertial frame.

Assume that the bearing angle \( \beta(t) \) and the subtended angle \( \alpha(t) \) can be obtained by some image processing algorithm

\[ \begin{align*} \beta(t) &= \theta(t) - \arctan \left( \frac{z_y(t)}{z_x(t)} \right), \\
\alpha(t) &= 2\arctan \left( \frac{L}{2d(t)} \right), \end{align*} \tag{6} \]

where \( d(t) \) is the range between the two objects

\[ d(t) = \sqrt{z_x^2(t) + z_y^2(t)}. \tag{7} \]

With equations (6) and (7), the coordinates of the relative motion can be expressed as

\[ \begin{align*} z_x(t) &= \frac{L\cos(\theta(t) - \beta(t))}{2\tan(\alpha(t)/2)}, \\
z_y(t) &= \frac{L\sin(\theta(t) - \beta(t))}{2\tan(\alpha(t)/2)}. \tag{8} \end{align*} \]

It follows from equations (5) and (8) that

\[ \begin{align*} \frac{\partial}{\partial \alpha(t)} \alpha(t) + \frac{\partial}{\partial \beta(t)} \beta(t) + \frac{\partial}{\partial \theta(t)} \theta(t) &= v_x(t) - \dot{x}(t), \\
\frac{\partial}{\partial \alpha(t)} \dot{x}(t) + \frac{\partial}{\partial \beta(t)} \dot{y}(t) + \frac{\partial}{\partial \theta(t)} \dot{\theta}(t) &= v_y(t) - \dot{y}(t). \tag{9} \end{align*} \]

Let the vector of unknown parameters be

\[ \eta(t) = \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \eta_3(t) \end{bmatrix} = \frac{1}{L} \begin{bmatrix} 1 \\ v_x \\ v_y \end{bmatrix}, \tag{10} \]

where both the obstacle’s characteristic length \( L \) and its unknown velocity are assumed to be constant. Solving (9) for \( \dot{\alpha}(t) \) and \( \dot{\beta}(t) \) leads to

\[ \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} = F(t) \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \eta(t) + \begin{bmatrix} 0 \\ \dot{\theta}(t) \end{bmatrix}, \tag{11} \]

where

\[ F(t) = \begin{bmatrix} 4\sin^2(\alpha/2)\cos(\theta - \beta) & 4\sin^2(\alpha/2)\sin(\theta - \beta) \\ -2\tan(\alpha/2)\sin(\theta - \beta) & 2\tan(\alpha/2)\cos(\theta - \beta) \end{bmatrix}. \tag{12} \]

The estimation objective is to estimate the unknown parameter \( \eta(t) \) using visual measurements \( \alpha(t), \beta(t) \) and the vehicle’s own control \( \dot{x}(t), \dot{y}(t), \) and \( \dot{\theta}(t) \).

It can be noticed that the matrix \( w(t) \) and the vector \( g(t) \) in (11) are known. As a result, the system in (11) exhibits the structure to which an existing nonlinear observer, the IBO in [20], can be applied to estimate \( \eta(t) \). Other observers including the observers in [21], [22] can also be applied. The IBO is selected here for its easy implementation.

To apply the IBO, the following assumption is needed:

**Assumption 1**: Let \( \nu_i(\tau) \) denote the \( i^{th} \) column of \( w^\top(t) \) in (11). There are no nontrivial constants \( \kappa_i \) (for \( i = 1, 2, 3 \)) such that

\[ \sum_{i=1}^{3} \kappa_i \nu_i(\tau) = 0, \tag{13} \]

for all \( \tau \in [t, t + \mu] \), where \( \mu > 0 \) is a sufficiently small constant.

Let \( \hat{\eta}(t) \) be the estimate of \( \eta(t) \)

\[ \hat{\eta}(t) = \begin{bmatrix} \hat{\eta}_1(t) \\ \hat{\eta}_2(t) \\ \hat{\eta}_3(t) \end{bmatrix}. \tag{14} \]
The following observer can be designed for the system (11):
\[
\begin{aligned}
\dot{\hat{\alpha}}(t) &= GA_m \begin{bmatrix}
\hat{\alpha}(t) - \alpha(t) \\
\hat{\beta}(t) - \beta(t)
\end{bmatrix} + w^T(t)\dot{\eta}(t) + q(t), \\
\dot{\eta}(t) &= -G^2w^T(t)P \begin{bmatrix}
\hat{\alpha}(t) - \alpha(t) \\
\hat{\beta}(t) - \beta(t)
\end{bmatrix},
\end{aligned}
\]
where $G$ is a scalar constant and $A_m$ is a $2 \times 2$ Hurwitz matrix. The matrix $P$ is the positive definite solution of the Lyapunov equation $A_m^TP + PA_m = -Q$ where $Q$ is a positive-definite symmetric matrix.

According to Theorem 2.3 in [20] (page 65), there exists a positive constant $G_0$ such that choosing $G > G_0$ ensures the estimation errors $\dot{\eta}(t) - \eta(t)$ converge to zero exponentially.

IV. Simulation Results

Trajectory generation with vision-based obstacle avoidance is implemented in Matlab. The following settings are used in the simulations:

1) Robot parameters: $R = 1$, $l = 0.8$, $\rho = 0.2$.
2) Boundary conditions:
   \[
   q^0 = [0, 0, \pi/4, 0]^T, \ t_0 = 0 \text{ s},
   \]
   \[
   q^f = [17, 10, -\pi/4, 0]^T, \ t_f = 40 \text{ s}.
   \]
3) Moving obstacles: let $O_i(t)$ be the center of the $i^{th}$ obstacle at time instance $t$ and choose
   \[
   O_1(t_0) = [15, 6]^T,
   \]
   \[
   O_2(t_0) = [5, 12]^T,
   \]
   \[
   O_3(t_0) = [18, 10]^T,
   \]
   \[
   r_i = 0.5, \text{ for } i = 1, 2, 3.
   \]
4) Velocities of the obstacles:
   \[
   v_1 = [-0.3, 0.2]^T,
   \]
   \[
   v_2 = [0.4, -0.3]^T,
   \]
   \[
   v_3 = [-0.2, 0.1]^T.
   \]
5) Estimation parameters and initial conditions:
   \[
   A_m = -I_2, \ P = I_2/2,
   \]
   \[
   \dot{\eta}_1(0) = [0.9, -0.1, 0.1]^T,
   \]
   \[
   \dot{\eta}_2(0) = [0.9, -0.1, 0.2]^T,
   \]
   \[
   \dot{\eta}_3(0) = [0.9, -0.1, 0]^T.
   \]

All quantities conform to a given unit system, for instance, meter, meter per second, and etc. In the simulations, the visual measurements are corrupted by 1% uniform noise. The vehicle’s own state information are assumed to be perfectly known.

The free-space path of the robot is shown in Fig. 4, together with the trajectories of the three obstacles. The initial and final positions of the robot and the obstacles are indicated by triangles and squares, respectively. The moving directions of the obstacles are also indicated by arrows pointing to their final positions. It can be seen from Fig. 4(a) that the robot collides with obstacle_1 and obstacle_3 at two different locations, shown by bold circles at the collision time instances. Figure 4(a) should be interpreted as an illustration of how the vehicle’s free-path trajectory collides with the obstacles instead of how the vehicle will behave. In an actual situation, the vehicle might fail to proceed after colliding with the first obstacle so that collision with the next obstacle might not occur. Figure 4(b) shows the corresponding $u_1(t)$ and $u_2(t)$ for the vehicle control.

When the obstacles’ sizes and velocities are known, implementation of the analytical trajectory generation and replanning algorithm in [13] gives the vehicle’s path in Fig. 5(a), where the vehicle avoids colliding with obstacle_1 and obstacle_3. Figure 5(b) shows the corresponding vehicle control signals.

When the obstacles’ sizes and velocities are unknown, these unknown information are estimated via the vehicle’s onboard vision sensor. Utilizing these estimated information,
the vehicle’s trajectory and control are shown in Fig. 6. Vision-based estimation of the obstacles’ velocities is given in Fig. 7, where in each subfigure the two solid lines denote the components of the velocity vector \((v_x, v_y)\) of the obstacle. The dashed lines are the estimates approaching their corresponding true values.

It can be seen from Figs. 5 and 6 that the vehicle takes different trajectories to reach the goal position while avoiding all obstacles. This is related to the estimated quantities at the time instance that the path is recalculated. Intuitively the path should be replanned at every time instance that a new estimate is obtained. However, this is very time-consuming. Besides, it has been observed that utilizing the “wrong” estimates in the transient stage might lead the vehicle to a direction that would fail to avoid the obstacles at a later time. Thus, a simple strategy is used in the simulation, where estimates within the first 10 seconds are not used. In other words, the vehicle takes the free-space path during \(t \in [0, 10]\). This is clear by comparing between Figs. 4 and 6. After that, the vehicle checks every 5 seconds to see if the path needs to be replanned using the current estimates. Obviously, this simple strategy will fail if the obstacles are close to the vehicle’s initial position since collision might occur before the estimates converge. Generally speaking, the longer the convergence speed of the estimation scheme, the more likely that the vehicle would collide with the obstacles. Complete analysis of the effect of the estimation scheme on the obstacle avoidance is in need. This remains as our future research interests.

V. Conclusion

This paper considers trajectory generation for avoiding dynamic obstacles moving with unknown constant velocities with vision-based estimation for a nonholonomic robot equipped with a single camera. An existing identifier-based observer is used to estimate the obstacles’ unknown size and unknown velocities. The effectiveness of the proposed
Fig. 7. Vision-based estimation of the obstacles’ velocities. Solid lines denote the true values of the obstacles’ velocity components \((v_x, v_y)\). Dashed lines are the estimates approaching their corresponding true values.

Complete analysis of the effect of the estimation scheme on the path regeneration remains as our future research interests.

REFERENCES


