

# KINEMATICS-BASED CHARACTERIZATION OF THE COLLISION COURSE

F. Belkhouche\* and B. Belkhouche\*\*

## Abstract

The problem of collision course between a mobile robot and a moving object is modeled in polar coordinates using the kinematics equations. A model of the relative motion of the moving object as seen by the robot is then derived. This model consists of the relative velocities along and across the visibility line, and gives the range rate and the turning rate of the moving object with respect to the robot. The conditions for the collision course are derived in terms of the robot's and the moving object's states. We define two types of collision course: the exact collision course and the weak collision course. The exact collision course always results in a collision, and is clearly characterized by a given set of equations. The weak collision course may become an exact collision course near collision and allows an early detection of the collision course in various scenarios. Several examples and scenarios illustrating the theory are shown using simulation.

## Key Words

Dynamic environment, collision course, relative kinematics equations

## 1. Introduction

The ability to avoid collision with moving obstacles is necessary for various applications of mobile robots. This ability significantly increases the mobility of a robot for navigation, and thus allows the robot to perform other tasks. Recently, research on robotics navigation has shifted from static environments to dynamic environments [1–5]. Planning collision-free paths in dynamic environments requires a dynamic collision avoidance system. Path planning in dynamic environments is more difficult, where algorithmic complexity issues are widely studied [6, 7]. Collision detection problems are discussed in different areas such as mobile robotics, computer graphics and simulation, intelligent transportation, and aerospace engineering. In general, collision detection deals with the problem of checking whether two objects overlap in space, or whether their boundaries intersect as they move. Collision detection may have dif-

ferent aspects; therefore, the same collision detection problems are seen from different perspectives in different fields. For example, probability approaches are widely used in aerospace engineering [8, 9]; however, these techniques are not used in computer graphics. Various methods such as spatio-temporal intersection, swept volume, and proximity and distance queries are discussed in the literature. Several computer modeling techniques employ the swept volume method, where the aim is to determine the overlap between the swept volume and the moving obstacle. These techniques are quite efficient for determining whether two models overlap. However, false collision can be detected if the time dimension is not added. Also, exact computation of such volumes is time consuming, especially when objects undergo a rotational motion.

Several researchers in robotics combine the collision detection problem with path planning and navigation in dynamic environments. Some of the earliest methods can be found in [10–12], where collision detection with an obstacle that moves with constant speed at fixed direction is accomplished by computing a collision front, which is a number of curve and line segments characterizing collision points at a certain speed. Collision detection based on the prediction of the future path of the obstacle using the sensory system is discussed in [1, 5, 13]. In [13], the task is achieved by a learning process. The training data consist of trajectories of the obstacles obtained by means of video cameras.

The velocity obstacle approach for collision detection is suggested in [14]. This technique is among the most important techniques for collision detection between a wheeled mobile robot and a circular moving object. The velocity obstacle approach maps the dynamic environment into the velocity space. The velocity obstacle is a first order approximation of the robot velocities that would cause collision. The drawback of this method is that collision prediction may be inaccurate if the obstacle does not move in a straight line. A collision cone approach is suggested in [15] for collision detection and avoidance, where the moving objects have arbitrary shapes. The collision cone approach is based on the relative velocities of the robot and the moving object. Important results are presented and proven rigorously; however, the authors focus on the non-accelerating case, which is relatively simple. Another important work can be found in [16]. The method is based on spatial decomposition of the relative motion (relative

\* Texas A&M International University; e-mail: fbelkhouche@tamiu.edu

\*\* UAE University; e-mail: B.Belkhouche@uaeu.ac.ae

Recommended by Prof. Simon Yang  
(paper no. 206-3123)

position, velocity, and acceleration), from which an urgency function is derived and used. Algebraic expressions are used in [17], and non-linear programming is used to solve the problem in an optimal way.

Collision detection using the notion of a distance function is widely discussed [18–23]. The aim is to detect the pair of the closest features between two objects. Different algorithms for distance computation are discussed in the literature. Two families of methods for distance computation are widely used [16]: computational geometry methods and iterative methods. For distance computation between non-overlapping convex polytopes, the algorithms suggested by Lin and Canny [21], and Gilbert *et al.* [22] are among the most known algorithms. These algorithms have been extended and improved in different ways by other researchers [24–27]. Typically, distance and proximity collision detection methods are time consuming. Thus, improving computation time is an important issue. This problem is considered in [28, 29]. Other techniques for collision detection based on distance functions use range sensors [19]. These techniques are quite successful in practice. However, they may lead to wrong conclusion or false alarm, since two approaching objects are not always in a collision course. In [30], the authors used the smallest polar angle criterion instead of the minimum distance for the relative distance. Some of the previous methods can be augmented by adding the element of time. This approach is called the time configuration space [31–33]. In [34], collision detection is performed using tactical sensors.

Note that many of the suggested methods [5, 10–12, 32] are limited the simple particular case of linear or piecewise linear motions, or constant speeds.

Our goal in this paper is to elaborate a new approach for collision course detection for robotics applications, without any assumptions on the motion of the robot and the moving object. Our contribution consists of the use of the relative kinematics equations, expressed in terms of the velocity along and across the visibility line between the robot and the moving object. We define two collision types: the exact collision course and the weak collision course. Here, we do not assume linear or piecewise linear motions, or constant speeds. The main application of the method is in robot navigation in dynamic environments, where obstacles are moving with unknown motion.

This paper is organized as follows. In Section 2, the problem is formulated. Definitions concerning the geometry and kinematics quantities are introduced in Section 3. The relative kinematics models are also derived in this section. In Section 4, collision with a stationary object is discussed briefly. In Section 5, collision conditions between two geometric points are derived. The weak collision course is then discussed in Section 6. These results are extended to the case of circular objects in Section 7. Finally, an extensive simulation is used to illustrate our theory in Section 8.

## 2. Problem Formulation

Let  $R$  denote a mobile robot moving in a two-dimensional workspace  $W$ . We assume that the dynamics of the robot

can be described by a differential equation of the form  $\dot{s}_r = f(s_r, u_r)$ , where  $s_r \in S_r$  is the state of the robot and  $u_r \in U_r$  is its control;  $S_r$  and  $U_r$  represent the state and control spaces, respectively.  $B$  is an object moving in the workspace. The motion of  $B$  is characterized by a differential equation such as  $\dot{s}_b = f(s_b, u_b)$ , where  $s_b \in S_b$  is the state of the moving object, and  $S_b$  is the state space.  $u_b$  may be a control or command input of  $B$ . Let  $P_r(t) = (x_r(t), y_r(t))$  denote the trajectory of the robot, where  $x_r$  and  $y_r$  are the robot's coordinates in the inertial frame of reference. In a similar way  $P_b(t) = (x_b(t), y_b(t))$  denotes the trajectory of the moving object, and  $x_b$  and  $y_b$  are the moving object's coordinates in the inertial frame of reference. Both  $P_b(t)$  and  $P_r(t)$  are assumed to be continuous functions. Given the motion dynamics of the robot and the moving object, our aim is to find the conditions of the collision course between the robot and the moving object as a function of the state variables  $s_r$  and  $s_b$ . We assume that:

1. The robot does not have a priori knowledge regarding the motion of the moving object. However, the moving object's velocity and position are assumed to be measurable in real time by the robot's sensory system.
2. Both the robot and the moving object have circular shapes. In order to simplify the analysis, the robot is reduced to its reference point, and the moving object is enlarged by the robot's radius. This approach is widely used in robotic navigation [1, 14].
3. The moving object is a rigid body, where all its points move at the same velocity.

The robot moves in the horizontal plane according to the following kinematics equations:

$$\begin{aligned}\dot{x}_r &= v_r \cos \theta_r \\ \dot{y}_r &= v_r \sin \theta_r \\ \dot{\theta}_r &= \omega_r \\ \dot{v}_r &= a_r\end{aligned}\tag{1}$$

Under system (1), the robot state is given by  $s_r = [x_r, y_r, \theta_r, v_r]^T$ , where  $v_r$  is the robot's linear velocity,  $a_r$  is its linear acceleration,  $\theta_r$  is the robot orientation angle, and  $\omega_r$  is the robot angular velocity. The control input for the robot is given by  $(\omega_r, a_r)$ . The moving object moves according to the following kinematics equations:

$$\begin{aligned}\dot{x}_b &= v_b \cos \theta_b \\ \dot{y}_b &= v_b \sin \theta_b \\ \dot{\theta}_b &= \omega_b \\ \dot{v}_b &= a_b\end{aligned}\tag{2}$$

The state space of the moving object is given by  $s_b = [x_b, y_b, \theta_b, v_b]^T$ , where  $v_b$  is its linear velocity,  $a_b$  is its linear acceleration,  $\theta_b$  is its orientation angle, and  $\omega_b$  is its angular velocity. A relative position function is defined as

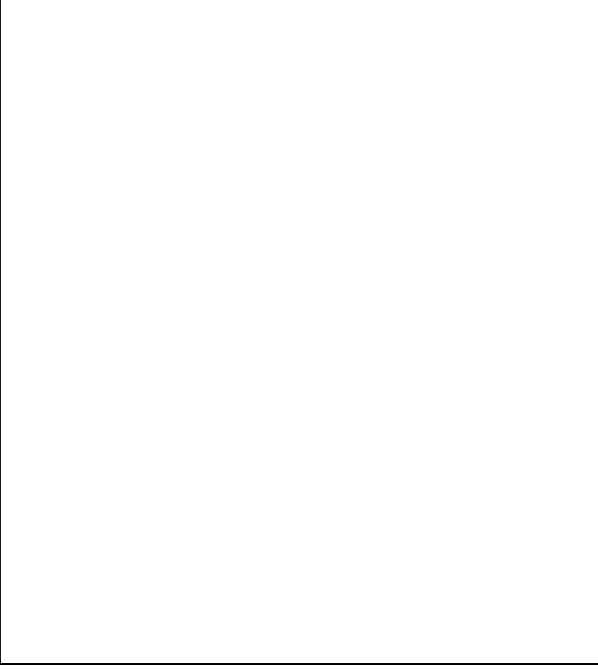


Figure 1. Geometric representation of the robot and the moving object in the horizontal plane.

$P_d = (x_d, y_d)$ , with  $x_d = x_b - x_r$ ,  $y_d = y_b - y_r$ . If  $R$  and  $B$  have radii  $\varepsilon_1$  and  $\varepsilon_2$ , respectively, a collision occurs if:

$$\|P_d\| = \varepsilon'_1 + \varepsilon'_2 \quad (3)$$

where  $\varepsilon'_1 \rightarrow 0, \varepsilon'_2 \rightarrow \varepsilon_1 + \varepsilon_2$ .

### 3. Modeling the Kinematics and Geometry

Consider the geometry shown in Fig. 1. We define the following quantities:

1. The visibility line  $\vec{V}_l$  is the imaginary straight line that starts at the robot reference point and is directed towards the moving object.
2. The visibility line angle  $\gamma_d$  is the angle from the reference line to the visibility line.
3. The relative range or distance between the robot and the moving object reference points is denoted by  $l_d$ .

Points  $b_1$  and  $b_2$  represent the upper and lower tangents to the circle from the robot's reference point. We also define the following quantities:

4. The visibility lines between the robot and points  $b_1$  and  $b_2$  are given by  $\vec{V}_{b1}$  and  $\vec{V}_{b2}$ , respectively.
5. The visibility line angles constructed based on  $\vec{V}_{b1}$  and  $\vec{V}_{b2}$  are  $\gamma_{b1}$  and  $\gamma_{b2}$ , respectively. The distances between the robot and points  $b_1$  and  $b_2$  are given by  $l_{b1}$  and  $l_{b2}$ , respectively.
6. The collision cone  $C_{con}$  is defined as follows  $C_{con} = \{\gamma : \gamma \in [\gamma_{b1}, \gamma_{b2}]\}$ .
7. The collision front is the set of points on the enlarged moving object circumference for which  $\gamma \in [\gamma_{b1}, \gamma_{b2}]$  is

satisfied. Let  $m$  denote a point on the collision front.  $\vec{V}_m$  is the visibility line between  $R$  and  $m$ ,  $\gamma_m$  is the visibility line angle based on  $\vec{V}_m$ , and  $l_m$  is the distance between point  $m$  and  $R$ .

8. The direction of the linear velocities of the robot and the moving object are  $\vec{\lambda}_r$  and  $\vec{\lambda}_b$ , respectively. The direction of the linear velocity of the moving object at points  $b_1$ ,  $b_2$  and  $m$  are  $\vec{\lambda}_{b1}$ ,  $\vec{\lambda}_{b2}$  and  $\vec{\lambda}_m$ , respectively.
9. The velocity ratio  $k$  is defined as follows:

$$k = \frac{v_b}{v_r} \quad (4)$$

Clearly,  $\gamma_d$  and  $l_d$  are given by:

$$l_d^2 = x_d^2 + y_d^2, \tan \gamma_d = \frac{y_d}{x_d} \quad (5)$$

Similar to  $l_d$ , the visibility line angle  $\gamma_d$  can be calculated based on the robot and the moving object coordinates when they are known. This is possible even if the moving object and the robot are not visible to each other. Let  $\vec{v}_r$  and  $\vec{v}_b$  be the velocity vectors for the robot and the moving object, respectively. Consider the relative velocity vector:

$$\vec{v}_{br} = \vec{v}_b - \vec{v}_r \quad (6)$$

The velocity vector  $\vec{v}_{br}$  can be decomposed into two components in the Cartesian plane as follows:

$$\begin{aligned} v_{br}^x &= \dot{x}_d = v_b \cos \theta_b - v_r \cos \theta_r \\ v_{br}^y &= \dot{y}_d = v_b \sin \theta_b - v_r \sin \theta_r \end{aligned} \quad (7)$$

System (7) gives the relative motion in Cartesian coordinates. Transformation to polar coordinates of systems (1) and (2) yields the following system for the robot's equation:

$$\begin{aligned} \dot{l}_r &= v_r \cos(\theta_r - \gamma_r) \\ l_r \dot{\gamma}_r &= v_r \sin(\theta_r - \gamma_r) \\ \dot{\theta}_r &= \omega_r \\ \dot{v}_r &= a_r \end{aligned} \quad (8)$$

where  $(l_r, \gamma_r)$  gives the robot's position in polar coordinates, and for the moving object

$$\begin{aligned} \dot{l}_b &= v_b \cos(\theta_b - \gamma_b) \\ l_b \dot{\gamma}_b &= v_b \sin(\theta_b - \gamma_b) \\ \dot{\theta}_b &= \omega_b \\ \dot{v}_b &= a_b \end{aligned} \quad (9)$$

where  $(l_b, \gamma_b)$  gives the moving object's position in polar coordinates.  $\vec{v}_{br}$  can be decomposed into two components along and across the visibility line  $\vec{V}_l$  as follows:

$$\vec{v}_{br} = \dot{l}_d \vec{u}_{\parallel} + l_d \dot{\gamma}_d \vec{u}_{\perp} \quad (10)$$

where  $\vec{u}_{\parallel}$  and  $\vec{u}_{\perp}$  are the unit vectors along and across the visibility line, respectively. By considering systems (8), (9) and (6) we get:

$$\begin{aligned} \dot{l}_d &= v_b \cos(\theta_b - \gamma_d) - v_r \cos(\theta_r - \gamma_d) \\ l_d \dot{\gamma}_d &= v_b \sin(\theta_b - \gamma_d) - v_r \sin(\theta_r - \gamma_d) \end{aligned} \quad (11)$$

It is easy to deduce from the sign of  $\dot{l}_d$  whether the robot and the moving object are approaching or moving away from each other. The second equation gives the turning rate of the moving object with respect to the robot. The kinematics model given by (11) becomes in the case of a stationary object

$$\begin{aligned} \dot{l}_d &= -v_r \cos(\theta_r - \gamma_d) \\ l_d \dot{\gamma}_d &= -v_r \sin(\theta_r - \gamma_d) \end{aligned} \quad (12)$$

Equation (11) can be reduced to a system similar to (12) in the case of a moving object by re-writing  $\vec{v}_{br}$  in a different way. This allows us to write

$$\begin{aligned} v_{br}^{\parallel} &= \dot{l}_d = -v_{br} \cos(\theta_{br} - \gamma_d) \\ v_{br}^{\perp} &= l_d \dot{\gamma}_d = -v_{br} \sin(\theta_{br} - \gamma_d) \end{aligned} \quad (13)$$

with

$$v_{br} = \sqrt{\dot{y}_d^2 + \dot{x}_d^2}, \tan \theta_{br} = \frac{\dot{y}_d}{\dot{x}_d} \quad (14)$$

It is easy to see the similarity between equation (13) and equation (12), which is derived for a stationary object. The main advantage of equation (13) is that the collision detection with a moving object is reduced to a stationary object situated at distance  $l_d$  from the robot and orientation  $\gamma_d$ , and thus very complex scenarios can be reduced to simple ones. Note that equation (13) is reduced to equation (12) when  $v_b = 0$ . We introduce the following definitions:

### Definitions

*Collision course:* A collision course between the robot and the moving object means that, given the instantaneous motion profile and the geometric positions of the robot and the moving object, a collision will take place in the future.

*Pure pursuit:*  $R$  is applying a pure pursuit towards another object  $B$  if the velocity vector of  $R$ ,  $\vec{v}_r$ , lies on the visibility line  $\vec{V}_l$  between  $R$  and  $B$ . As a result the orientation angle of  $R$  is equal to the visibility line angle:  $\theta_r = \gamma_d$ .

*Deviated pursuit:* Here  $R$  is not directly heading towards  $B$ , but there exists a non-zero angle between the velocity vector of  $R$  and the line of sight, that is:  $\theta_r = \gamma_d + \alpha_0$ ,  $\alpha_0$  is the deviation angle.

*Exact collision course:* Two moving objects  $R$  and  $B$  are in a collision course if the visibility line angle rate between them satisfies  $\dot{\gamma}_d = 0$ , and a decreasing range.

*Weak collision course:* The weak collision course between objects  $R$  and  $B$  is characterized by  $\dot{\gamma}_d \rightarrow 0$ , and a decreasing range.

*Point collision course:* A collision course between two geometric points. It also characterizes the collision course between a geometric point and a point on a moving body.

*Circle collision course:* A collision course between a circle shaped object and a geometric point.

*Collision point:* The collision point  $P_{col} = (x_{col}, y_{col})$  is given by:

$$P_{col} = P_r(t_f) = P_b(t_f) \quad (15)$$

where  $t_f > t \geq 0$  is the collision time.

*Path intersection:* The paths traced by the robot and the moving object intersect if:

$$P_r(t_1) = P_b(t_2) \quad (16)$$

with  $t_1, t_2 > t \geq 0$ . The intersection point  $P_{int} = (x_{int}, y_{int})$  satisfies:  $P_{int} = P_r(t_1) = P_b(t_2)$ .  $P_{int} = P_{col}$  when  $t_1 = t_2$ .

## 4. Case of a Stationary Object

In this section we briefly discuss the collision course with a stationary object. Our analysis is based on (12). It is well known [1, 14] that collision with a stationary circle takes place when the instantaneous velocity vector direction of the robot lies in the collision cone  $C_{con}$ , that is:

$$\vec{\lambda}_r \in [\vec{V}_{b1}, \vec{V}_{b2}] \quad (17)$$

or:

$$\theta_r \in [\gamma_{b1}, \gamma_{b2}] \quad (18)$$

In the pure pursuit towards a stationary object, the visibility line angle is constant. Characterizing the collision using (18) does not take into account the collision course resulting from the deviated pursuit. The deviated pursuit is characterized by:

$$\begin{aligned} \theta_r &= \gamma_m + \alpha_0 \\ \gamma_m &\in [\gamma_{b1}, \gamma_{b2}] \end{aligned} \quad (19)$$

with  $\alpha_0 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ . When  $\alpha_0 \rightarrow \pm\frac{\pi}{2}$ , (with  $\alpha_0 \neq \pm\frac{\pi}{2}$ ) we talk about the weak deviated pursuit. Unlike the pure pursuit collision course with a stationary object, the weak deviated pursuit is not characterized by a constant value of  $\gamma_d$ , except near collision. In this case, the distance robot-object is decreasing very slowly, and the rate of change of the visibility line angle is near its maximum value. This makes the weak deviated pursuit collision course more difficult to detect. The following result is a generalization of condition (18) to the deviated pursuit.

**Proposition 1.** *The robot is in a collision course with the stationary circle if:*

$$\begin{aligned} \theta_r &= \gamma_m + \alpha_0 \\ \gamma_m &\in [\gamma_{b1}, \gamma_{b2}] \end{aligned} \quad (20)$$

with  $\alpha_0 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ .

**Proof:** The proof is simple if we consider the range rate equation in (12). ■

A more general case resulting in the collision course is when  $\gamma_d$  goes to an asymptotically stable position that results in a decreasing range. The equilibrium solutions for the visibility line angle rate in (12) are given by:

$$\gamma_d^- = \theta_r + 2\pi n \quad (21)$$

$$\gamma_d^+ = \pi + \theta_r + 2\pi n \quad (22)$$

where  $n$  is an integer. Clearly, from the range rate equation, solution  $\gamma_d^-$  results in a decreasing range, while solution  $\gamma_d^+$  results in an increasing range. Based on this observation, we have the following result.

**Proposition 2.** *The robot is in a collision course with the stationary circular object when  $\gamma_d^-$  is an asymptotically stable equilibrium solution for  $\gamma_m$ , with  $\gamma_m \in [\gamma_{b1}, \gamma_{b2}]$ .*

**Proof:** If  $\gamma_d^-$  is an asymptotically stable equilibrium position for  $\gamma_d$  then,  $\gamma_d \rightarrow \theta_r + 2\pi n = \gamma_d^-$  with time, and as a result  $\cos(\gamma_d^- - \gamma_d) \rightarrow 1$  and  $\dot{l}_d < 0$ . The proof is similar for any point  $m$  on the collision front. ■

It is relatively easy to detect the collision course with a stationary object due to the deviated pursuit. This can be accomplished using the property stated as follows.

**Proposition 3.** *When the robot is in a collision course resulting from the deviated pursuit with a stationary object, it results that:*

$$\frac{l_d \dot{\gamma}_d}{v_r} = c_0 = \text{constant} \quad (23)$$

with  $c_0 \in (-1, 1)$ .

**Proof:** From the second equation in (12), we get under the deviated pursuit

$$\frac{l_d \dot{\gamma}_d}{v_r} = -\sin(\alpha_0) \quad (24)$$

This is also true for any point  $m$  on the collision front. For the weak deviated pursuit,  $c_0$  is near  $\pm 1$ . ■

Note that the range rate is constant when the robot's speed is constant, which means that under the deviated pursuit, the robot approaches any point on the collision front at the same rate.

## 5. Collision Course Between Two Moving Geometric Points

We begin our analysis by considering the case where both the robot and the moving object are modeled as geometric points. Finding the collision course conditions between two geometric points is not necessarily simple. For example, finding the collision course between two accelerating geometric points is more difficult than finding the collision course between two non-rotating polygonal objects

moving in a constant direction with constant speed. Path intersection is a necessary condition for collision, but not sufficient. The path intersection is characterized by using the instantaneous values of the orientation angles of the robot and the moving goal. We have the following result:

**Proposition 4.** *For forward motions, the paths intersection can be characterized by the following conditions:*

$$A_1 = \frac{\sin(\theta_r - \theta_b)}{\sin(\theta_r - \gamma_d)} < 0 \quad (25)$$

and

$$A_2 = \frac{\sin(\theta_r - \theta_b)}{\sin(\theta_b - \gamma_d)} < 0 \quad (26)$$

We will see later that  $\sin(\theta_b - \gamma_d)$  and  $\sin(\theta_r - \gamma_d)$  have the same sign when the paths intersect. The proof for proposition 4 is as follows:



Figure 2. Path intersection.

**Proof:** Without loss of generality, consider the geometric configuration of Fig. 2. The plane is divided by the visibility line into two parts. The upper part, shown in Fig. 2(a) corresponds to:

$$\theta_b - \gamma_d \in (-\pi, 0) \quad (27)$$

The lower part, shown in Fig. 2(b) corresponds to

$$\theta_b - \gamma_d \in (0, \pi) \quad (28)$$

It is clear that path intersection requires  $\theta_r - \gamma_d$  and  $\theta_b - \gamma_d$  to belong to the same interval  $((0, \pi)$  or  $(-\pi, 0))$ . Therefore,  $\sin(\theta_b - \gamma_d)$  and  $\sin(\theta_r - \gamma_d)$  have the same sign. The proof for each configuration is as follows:

*Upper half*

From Fig. 2(a), the paths of the robot and the moving object intersect when

$$\theta_r \in (\theta_b, \gamma_d) \quad (29)$$

From which it is possible to write

$$\theta_r - \theta_b \in (0, \gamma_d - \theta_b) \quad (30)$$

and

$$\theta_r - \gamma_d \in (\theta_b - \gamma_d, 0) \quad (31)$$

Since  $\sin(\kappa) = -\sin(-\kappa)$ , from the intervals given in (30), (31) it turns out that  $\sin(\theta_r - \theta_b)$  and  $\sin(\theta_r - \gamma_d)$  have opposite signs. Thus, the real number  $A_1$  is always negative when the paths intersect.

*Lower half*

From Fig. 2(b), the paths of the robot and the moving object intersect when

$$\theta_r \in (\gamma_d, \theta_b) \quad (32)$$

From which it is possible to write

$$\theta_r - \theta_b \in (\gamma_d - \theta_b, 0) \quad (33)$$

and

$$\theta_r - \gamma_d \in (0, \theta_b - \gamma_d) \quad (34)$$

Similarly to the previous case,  $\sin(\theta_r - \theta_b)$  and  $\sin(\theta_r - \gamma_d)$  have opposite signs. Thus, path intersection requires the real number  $A_1$  to be always negative in this case also. The proof is similar for the real number  $A_2$  in (26). ■

Conditions (25) and (26) give a simple test for determining whether the velocity directions  $\vec{\lambda}_r$  and  $\vec{\lambda}_b$  intersect. This test is instantaneous and valid for both constant and time varying orientation angles. Our result concerning the exact collision course is stated as follows.

**Proposition 5.** *If (25) and (26) are satisfied for forward motions, with*

$$v_r \sin(\theta_r - \gamma_d) = v_b \sin(\theta_b - \gamma_d) \quad (35)$$

*then, the robot and the moving goal are in a collision course.*

Equation (35) expresses the collision course in terms of the speeds, orientation angles, and positions. This can be seen in the formula of the visibility line angle given in (5). The proof for proposition 5 is stated as follows.

**Proof:** The proof is based on the time derivative of the relative range  $l_d$ . By using (35) and the equation for the relative range between the robot and the moving object, we get

$$\dot{l}_d = \frac{v_r \sin(\theta_r - \gamma_d)}{\sin(\theta_b - \gamma_d)} \cos(\theta_b - \gamma_d) - v_r \cos(\theta_r - \gamma_d) \quad (36)$$

By using trigonometric identities, we get in terms of the robot velocity

$$\dot{l}_d = v_r \frac{\sin(\theta_r - \theta_b)}{\sin(\theta_b - \gamma_d)} = A_2 v_r \quad (37)$$

or in terms of the velocity of the moving object  $v_b$

$$\dot{l}_d = v_b \frac{\sin(\theta_r - \theta_b)}{\sin(\theta_r - \gamma_d)} = A_1 v_b \quad (38)$$

Since  $A_1$  and  $A_2$  are negative,  $\dot{l}_d < 0$  for forward motions. ■

When the path intersection conditions and (35) are satisfied, the relative range  $\dot{l}_d$  changes sign at the collision point, i.e., when  $l_d = 0$ . Also, when this point is reached, the visibility line angle  $\gamma_d$  switches by  $\pi$  (the new value of  $\gamma_d$  equals the old value  $+\pi$ ), and  $A_1$  and  $A_2$  become positive. The following is a property of the collision course.

**Proposition 6.** *When the robot and the moving object are in a collision course, the visibility line angle is constant.*

**Proof:** This result is obtained directly from the second equation in (11), which gives the rate of turn of the visibility line angle. Using (35), we get  $\dot{\gamma}_d = 0$  in the second equation in (11), which implies constant value for the visibility line angle. ■

From this proposition, it results that, when the robot and the moving object are in a collision course, the visibility lines at different times are parallel to each other.

## 6. Weak Collision Course

Equation (35) does not take into account the collision course that corresponds to a non-constant visibility line angle. We define the collision course resulting from (35) as the exact collision course, and we define the collision course corresponding to  $\dot{\gamma}_d \neq 0$  as the weak collision course. In many situations, the weak collision course results in an exact collision course that appears suddenly just before collision. Assume that  $v_r > v_b$  (the other case, when  $v_b > v_r$  is similar). The weak collision course resulting from the pure pursuit is characterized by

$$\theta_r^* = \gamma_d \quad (39)$$

and the exact collision course formulated in terms of the robot's orientation angle is obtained from equation (35) as follows:

$$\theta_r^{**} = \sin^{-1}(k \sin(\theta_b - \gamma_d)) + \gamma_d \quad (40)$$

The difference between  $\theta_r^*$  and  $\theta_r^{**}$  describes the transition from the pursuit and the exact collision course. Orientation angles between  $\theta_r^*$  and  $\theta_r^{**}$  result in a collision. It is important to note that there exist other weak collision course scenarios that are not captured by the pursuit.

## 7. Collision Course: Case of Circles

Here, we generalize our previous results. We saw that the exact collision course in the case of geometric points is characterized by a constant visibility line angle; and thus, at any time, there are single values of  $\theta_r, v_r$  that lead to collision. In the case of a circular moving object, there exist intervals for  $\theta_r, v_r$  that lead to collision. Here, instead of a collision point  $P_{col}$ , we define a collision circle  $C_{col}$ , which has  $\varepsilon_1 + \varepsilon_2$  as radius. We have the following result concerning the exact collision course.

**Proposition 7.** *The robot and the moving object are in a collision course if*

1. *The robot is in a collision course with the moving object's reference point.*
2. *There exists a point  $m$  on the collision front for which*

$$v_r \sin(\theta_r - \gamma_m) = v_b \sin(\theta_b - \gamma_m) \quad (41)$$

**Proof:** The proof is quite similar to the case of geometric points. ■

### Proposition 8.

1. *When  $k < 1$ . Let*

$$\theta_{r1} = \sin^{-1}(k \sin(\theta_b - \gamma_{b1})) + \gamma_{b1} \quad (42)$$

$$\theta_{r2} = \sin^{-1}(k \sin(\theta_b - \gamma_{b2})) + \gamma_{b2} \quad (43)$$

*Assume that  $\theta_{r1} < \theta_{r2}$ . The collision course takes place when*

$$\theta_r \in [\theta_{r1}, \theta_{r2}] \quad (44)$$

*When  $\theta_{r1} > \theta_{r2}$ , it suffices to permute  $\theta_{r1}$  and  $\theta_{r2}$  in (44).*

2. *When  $k > 1$ . Let*

$$\theta_{b1} = \sin^{-1}\left(\frac{1}{k} \sin(\theta_r - \gamma_{b1})\right) + \gamma_{b1} \quad (45)$$

$$\theta_{b2} = \sin^{-1}\left(\frac{1}{k} \sin(\theta_r - \gamma_{b2})\right) + \gamma_{b2} \quad (46)$$

*Assume that  $\theta_{b1} < \theta_{b2}$ . The collision course takes place when*

$$\theta_b \in [\theta_{b1}, \theta_{b2}] \quad (47)$$

*When  $\theta_{b1} > \theta_{b2}$ , it suffices to permute  $\theta_{b1}$  and  $\theta_{b2}$  in (47).*

**Proof:** The proof is similar for both cases, thus we consider only case 1.  $\theta_{r1}$  results in a collision with point  $b_1$ , and  $\theta_{r2}$  results in a collision with point  $b_2$ . Therefore any orientation angle between  $\theta_{r1}$  and  $\theta_{r2}$  results in a collision with a given point in the collision front. ■

### Proposition 9.

*The collision course between the robot and the circular object is given in terms of the speeds as follows*

1. *Let*

$$v_{r1} = \frac{v_b \sin(\theta_b - \gamma_{b1})}{\sin(\theta_r - \gamma_{b1})} \quad (48)$$

$$v_{r2} = \frac{v_b \sin(\theta_b - \gamma_{b2})}{\sin(\theta_r - \gamma_{b2})} \quad (49)$$

*Assume that  $v_{r1} < v_{r2}$ . The collision course takes place when*

$$v_r \in [v_{r1}, v_{r2}] \quad (50)$$

*When  $v_{r1} > v_{r2}$ , it suffices to permute  $v_{r1}$  and  $v_{r2}$  in (50).*

2. *Let*

$$v_{b1} = \frac{v_r \sin(\theta_r - \gamma_{b1})}{\sin(\theta_b - \gamma_{b1})} \quad (51)$$

$$v_{b2} = \frac{v_r \sin(\theta_r - \gamma_{b2})}{\sin(\theta_b - \gamma_{b2})} \quad (52)$$

*Assume that  $v_{b1} < v_{b2}$ . The collision course takes place when*

$$v_b \in [v_{b1}, v_{b2}] \quad (53)$$

*When  $v_{b1} > v_{b2}$ , it suffices to permute  $v_{b1}$  and  $v_{b2}$  in (53).*

**Proof:** The proof is similar for both cases, thus we consider only case 1.  $v_{r1}$  results in a collision with point  $b_1$ , and  $v_{r2}$  results in a collision with point  $b_2$ . Therefore any speed between  $v_{r1}$  and  $v_{r2}$  results in a collision with a given point in the collision front. ■

Equations (44) and (50) state that the visibility line angle varies, but within given limits. Propositions 7, 8 and 9 deal with the exact collision course, where the visibility line angle varies within a given interval. The next result concerns the weak collision course characterized by the pursuit. Now we discuss the collision course characterization based on model (13), which transforms the relative kinematics equations between two moving objects to a kinematics model between the robot and a stationary object. The following result concerns the exact collision course.

**Proposition 10.** *The exact collision course between the moving object and the robot is equivalent to the pure pursuit under (13).*

**Proof:** As we have seen the exact collision course is characterized by  $y_d = x_d \tan \gamma_d$ , with  $\gamma_d = \text{constant}$ . By taking the derivative, we get  $\dot{y}_d = \dot{x}_d \tan \gamma_d$ . From (13) it turns out that  $\gamma_d = \theta_{br}$ , which indicates the pure pursuit. ■

Under this formulation, the exact collision course between the robot and the moving circle can be described as

$$\begin{aligned}\theta_{br} &= \gamma_m \\ \gamma_m &\in [\gamma_{b1}, \gamma_{b2}]\end{aligned}\tag{54}$$

Recall that  $\tan \theta_{br} = \frac{y_d}{x_d}$ , which is different from the orientation angle of the robot. The deviated pursuit under (13) results also in a collision course.

**Proposition 11.** *The robot and the moving circle are in a collision course when*

$$\begin{aligned}\theta_{br} &= \gamma_m + \alpha_0 \\ \gamma_m &\in [\gamma_{b1}, \gamma_{b2}]\end{aligned}\tag{55}$$

with  $\alpha_0 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ .

**Proof:** The proof is similar to the case of a stationary object. ■

The deviated pursuit is relatively simple in the case of a stationary object. However, in the case of a moving object, the model given by (13) provides an important simplification of very complex scenarios.

## 8. Simulation

Our methods are illustrated using several simulation examples. These examples are carefully chosen to illustrate various scenarios in both the exact and the weak collision course. The paths of the robot and the moving object are shown in the Cartesian frame of reference. We also show other important quantities such as the rate of turn of the visibility line angle as a function of time. Our simulation is obtained using Matlab.

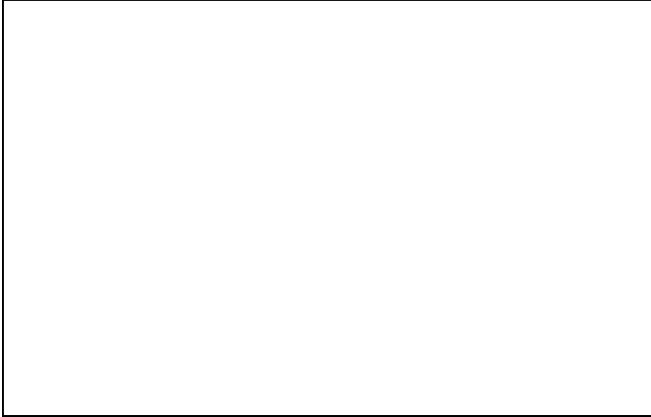


Figure 3. Exact collision course with points  $b_1$  and  $b_2$ ,  $\theta_r = \theta_{r1}$ .

### 8.1 Example 1: Exact Collision Course

This example illustrates the exact collision course. This scenario is shown in Fig. 3, where  $\theta_r = \theta_{r1}$  result in a collision with point  $b_1$ , and  $\theta_r = \theta_{r2}$  results in a collision with  $b_2$ .  $\theta_{r1}$  and  $\theta_{r2}$  satisfy the collision course equation given in (42) and (43). The robot's orientation angle profile

is shown in Fig. 4. Any orientation angle of the robot  $\theta_r \in [\theta_{r1}, \theta_{r2}]$  results in a collision with the moving object. This interval is represented by shaded area in the figure.



Figure 4. Robot's orientation angle profile  $\theta_r(t)$ .

### 8.2 Example 2: Transition from a Weak to an Exact Collision Course

This example is to illustrate the transition from the weak collision course due to the pure pursuit and the exact collision course. The moving circular object starts from point  $B_0(0,0)$  and moves in a vertical line at constant speed, the robot starts from point  $R_0(20,20)$  and moves with constant speed also. Three paths are shown for the robot. In path (1), the robot is in an exact collision course with  $B$ , with  $\theta_r$  given by (39). In (2) the robot is in a weak collision course with  $B$ , such that  $\theta_r$  is given by (40). In path (3), the robot moves with an orientation angle that is between (39) and (40). Orientation angles between (40) and (39) result in paths between (1) and (2) in Fig. 5, ending up with a collision.

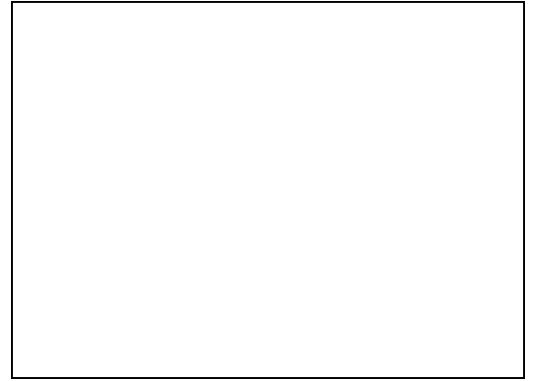


Figure 5. Comparison in the paths between the weak and the exact collision course.

### 8.3 Example 3: Multiple Moving Objects

This example shows multiple moving objects. The collision course is determined using the virtual plane and the collision cone. In the scenario shown in Fig. 6, collision takes place with  $B_3$ . This collision is easily detected by using the



virtual plane shown in Fig. 7, where the velocity vector of  $B_3$  lies inside the collision cone. Note that in this example, the robot is transformed into a stationary robot, while the moving objects are replaced by the virtual moving objects with new speeds and orientation angles.



Figure 6. A scenario showing multiple (4) moving objects and a robot in the workspace, collision takes place with  $B_3$ .



Figure 7. Virtual plane for the scenario of Fig. 6. The robot is transformed into a stationary robot, while the moving objects  $B_i$  are transformed into virtual moving objects, collision takes place with  $B_3$ .

## 9. Conclusion

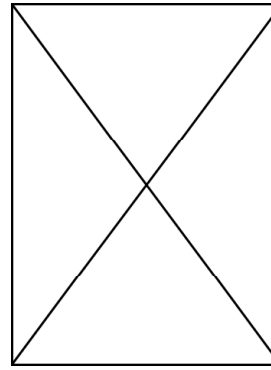
This paper deals with the problem of collision course characterization between a wheeled mobile robot and a circular moving object. The problem is modeled using relative kinematics equations based on two different models. Two types of collision course are defined: the exact collision course and the weak collision course. The weak collision course is more difficult to detect in general; however, in many situations, it leads to an exact collision course. The problem is first studied for two geometric points and generalized to the case of a circular object later. The collision course conditions are derived as a function of the instantaneous states of the robot and the moving object. The state consists of the positions, the speeds, and orientation angles. This presents an important particularity of the method compared with other methods. An extensive simulation is carried out to illustrate different collision scenarios.

## References

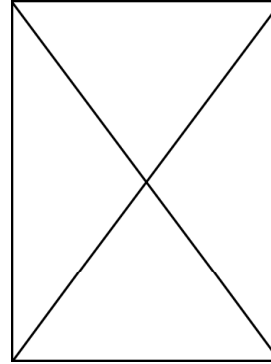
- [1] J.G. de Lamadrid & M. Gini, Path tracking through uncharted moving obstacles, *IEEE Transactions on Systems, Man and Cybernetics*, 20(6), 1990, 1408–1422.
- [2] N. Bourbakis, A traffic priority language for collision-free navigation of autonomous mobile robots in dynamic environments, *IEEE Transactions on Systems, Man and Cybernetics Part B*, 27(4), 1997, 573–587.
- [3] C. Shih, T. tian Lee, & W. Gruver, A unified approach for robot motion planning with moving polyhedral obstacles, *IEEE Transactions on Systems, Man and Cybernetics*, 20(4), 1990, 903–915.
- [4] R. Spence & S. Hutchinson, An integrated architecture for robot motion planning and control in the presence of obstacles with unknown trajectories, *IEEE Transactions on Systems, Man and Cybernetics*, 25(1), 1995, 100–110.
- [5] J.G. de Lamadrid, Avoidance of obstacles with unknown trajectories: Locally optimal paths and periodic sensor readings, *The International Journal of Robotics Research*, 13(6), 1994, 496–507.
- [6] J. Reif & M. Sharir, Motion planning in the presence of moving obstacles, *Journal of the Association of Computing Machinery*, 41(4), 1994, 764–790.
- [7] J. Hopcroft, J. Schwartz, & M. Sharir, On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the warehouseman's problem, *The International Journal of Robotics Research*, 3(4), 1984, 76–88.
- [8] M. Akella & K. Alfriend, Probability of collision between space objects, *Journal of Guidance, Control, and Dynamics*, 23(5), 2000, 769–772.
- [9] R. Patera, Satellite collision probability for nonlinear relative motion, *Journal of Guidance, Control, and Dynamics*, 26(5), 2003, 728–731.
- [10] K. Fujimura & H. Samet, Time minimal paths among moving obstacles, *Proc. IEEE International Conf. on Robotics and Automation*, Arizona, 1989, 1110–1115.
- [11] K. Fujimura & H. Samet, Motion planning in dynamic domain, *Proc. IEEE International Conf. on Robotics and Automation*, Ohio, 1990, 324–330.
- [12] K. Fujimura & H. Samet, Planning a time minimal motion among moving obstacles, *Algorithmica*, 10, 1993, 41–63.
- [13] D. Vasquez, F. Large, T. Fraichard, & C. Laugier, Moving obstacles' motion prediction for autonomous navigation, *Proc. International Conf. on Control, Automation, Robotics and Vision*, China, 2004, 149–154.
- [14] P. Fiorini & Z. Shiller, Motion planning in dynamic environments using velocity obstacles, *The International Journal of Robotics Research*, 17(7), 1998, 760–772.
- [15] A. Chakravarthy & D. Ghose, Obstacle avoidance in dynamic environment: A collision cone approach, *IEEE Transactions on Systems, Man and Cybernetics Part A*, 28(5), 1998, 562–574.
- [16] V. Hayward, S. Auby, A. Foisy, & Y. Ghallab, Efficient collision prediction among many moving objects, *The International Journal of Robotics Research*, 14(2), 1995, 129–143.
- [17] S.H. Suh & M.-S. Kim, An algebraic approach to collision-avoidance trajectory planning for dual-robot systems: Formulation and optimization, *Robotica*, 10, 1992, 173–182.
- [18] E. Gilbert & D. Johnson, Distance functions and their application to robot path planning in the presence of obstacles, *IEEE Journal of Robotics and Automation*, 1(1), 1985, 21–30.
- [19] K.M. Krishna & P. Karla, Detection, tracking and avoidance of multiple dynamic objects, *Journal of Intelligent and Robotic Systems*, 33, 2002, 371–408.
- [20] K. Kyriakopoulos & G. Saridis, Distance estimation and collision prediction for online robotic motion planning, *Automatica*, 28(2), 1992, 389–394.
- [21] M. Lin & J. Canny, A fast algorithm for incremental distance calculation, *Proc. IEEE International Conf. on Robotics and Automation*, California, 1991, 1008–1014.
- [22] E. Gilbert, D. Johnson, & S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional, *IEEE Journal of Robotics and Automation*, 4(2), 1988, 193–203.

- [23] E. Bernabeu & J. Tornero, Hough transform for distance computation and collision avoidance, *IEEE Transactions on Robotics and Automation*, 18(3), 2002, 393–398.
- [24] B. Mirtich, V-clip: Fast and robust polyhedral collision detection, *ACM Transactions on Graphics*, 17, 1998, 177–208.
- [25] S. Ehmann & M. Lin, Accelerated proximity queries between convex polydra using multi-level Voronoi marching, *Proc. IEEE/RSJ International Conf. on Intelligent Robots Systems*, Japan, 2000, 2101–2106.
- [26] S. Cameron, A comparison of two fast algorithms for computing the distance between convex polyhedra, *IEEE Transactions on Robotics and Automation*, 13(6), 1997, 915–920.
- [27] S. Cameron, Enhancing GJK: Computing minimum and penetration distance between convex polyhedra, *Proc. IEEE International Conference on Robotics and Automation*, Albuquerque, 1997, 3112–3117.
- [28] S. Ehmann & M. Lin, Accurate and fast proximity queries between polyhedra using convex surface decomposition, *Computer Graphics*, 20(3), 2001, 500–511.
- [29] K.E. Ho, A. Zaferakis, M.C. Lin, & D. Manocha, Fast and simple 2D geometric proximity queries using graphics hardware, *Proc. ACM Symposium on Interactive 3D Graphics*, New Mexico, 2001, 145–148.
- [30] L. Ortega, F. Feito, C. Grima, & A. Marquez, Collision detection using polar diagrams, *Proc. International Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, 2004, 117–120.
- [31] B. Aronov, S. Fortune, & G. Wilfong, Minimum-speed motions, *The International Journal of Robotic research*, 10(3), 1991, 228–238.
- [32] K. Fujimura, Time minimum routes in time dependent networks, *IEEE Transaction on Robotics and Automation*, 11(3), 1995, 343–351.
- [33] K. Fujimura & H. Samet, A hierarchical strategy for path planning among moving obstacles, *IEEE Transactions on Robotics and Automation*, 5(1), 1989, 61–69.
- [34] V. Lumelsky & A. Stepanov, Path planning strategies for a point mobile automation moving amidst unknown obstacles of arbitrary shape, *Algorithmica*, 2, 1987, 403–430.

## Biographies



*Fethi Belkhouche* is Assistant Professor of Systems Engineering at Texas A&M International University. He holds a Ph.D. in electrical engineering from Tulane University. His research interests include guidance theory, robot navigation, robot cooperation and formation control, and linearization methods.



*Boumediene Belkhouche* is Professor of Computer Science at the UAE University. His research areas include programming languages, software engineering, and path planning for autonomous mobile robots.