Wheeled mobile robot navigation using proportional navigation

FETHI BELKHOUCHE and BOUMEDIENE BELKHOUCHE∗

Electrical Engineering and Computer Science Department, Tulane University,
New Orleans, LA 70118, USA

Received 21 February 2006; accepted 7 April 2006

Abstract—We present a method for wheeled mobile robot navigation based on the proportional navigation law. This method integrates the robot’s kinematics equations and geometric rules. According to the control strategy, the robot’s angular velocity is proportional to the rate of turn of the angle of the line of sight that joins the robot and the goal. We derive a relative kinematics system which models the navigation problem of the robot in polar coordinates. The kinematics model captures the robot path as a function of the control law parameters. It turns out that different paths are obtained for different control parameters. Since the control parameters are real, the number of possible paths is infinite. Results concerning the navigation using our control law are rigorously proven. An extensive simulation confirms our theoretical results.

Keywords: Robot navigation; proportional navigation; relative polar kinematics equations.

1. INTRODUCTION

Robot navigation and obstacle avoidance are among the most important issues in robotics. Various navigation and obstacle avoidance methods are discussed in the literature. Among these methods, potential field methods [1–5] play a major role. The idea of the potential field was originally suggested for manipulator collision avoidance [1] and is widely used for mobile robots as well. In potential field methods the robot moves in a potential field that represents the sum of an attractive force resulting from the goal and repulsive forces resulting from the obstacles. These methods suffer from the problem of local minima in the resultant potential field. Also, a potential function with repulsive features must be constructed for each obstacle. Borenstein and Koren suggested a method called the virtual force field (VFF) [6]. This method combines the histogram grid world model with the concept

∗To whom correspondence should be addressed. E-mail: belkhouf@eeecs.tulane.edu
of potential fields. The VFF suffers from problems that are inherent to potential field methods, where the robot oscillates in a passage in which repulsive forces are applied simultaneously from opposite sides [7]. To remedy this shortcoming, Borenstein and Koren developed another method called the vector field histogram (VFH) [7, 8]. The VFH method uses a polar histogram with angular sectors, which gives a representation of the robot environment. Fox et al. [9] suggested the dynamic window approach, which searches through the space of instantaneous velocities of the robot and takes into account the motion dynamics of the robot. The instantaneous velocities searching time can slow down the navigation process, especially in dynamic environments. The path-velocity decomposition method was suggested by Kant and Zucker [10] to navigate the robot in the presence of moving obstacles. The path-planning problem is divided into two steps, (i) find an obstacle-free path for static obstacles and (ii) plan the velocity of the robot in order to avoid moving obstacles. In the first step, the authors used the visibility graph algorithm, which results in semi-free paths. This approach was also discussed in Ref. [11]. A dynamic potential field approach was suggested in Ref. [12] in order to navigate the robot in a dynamic environment. Dynamic potential field methods suffer from the same drawbacks as the standard potential field methods. In Ref. [13], the problem of robot navigation in a dynamically changing environment is addressed and a solution based on limit cycles is suggested. Other methods such as nominal path planning [14], point-to-point navigation [15], nearness diagram navigation [16–18], curvature velocity method [19], beam curvature method [20], adaptive navigation [21] and navigation using dynamical systems [22] were also suggested. The nearness diagram navigation method [16–18] uses a cell decomposition approach, where the cells are divided into three types: free, occupied, and unknown. Approaches based on cell decomposition may be time consuming. The curvature velocity method [19] treats the problem as one of a constrained optimization in the velocity space of the robot. The choice of the objective function is critical and approximation is necessary; these are the main drawbacks of the method. Adaptive navigation [21] uses a navigation function which consists of a first-order differential equation. However, a number of conditions were imposed on the movement and the velocity of the robot, and the location of the obstacle. The beam curvature method [20] combines a directional control method with the curvature velocity method. Similar to Ref. [19], this method is based on the optimization of an objective function in the linear and angular velocities. Methods based on point-to-point algorithms [15] are known for their computational complexity. This makes them interesting from a theoretical point of view, but they are rarely used in practice.

Reactive sensor-based navigation is another important family of methods used for robot navigation [23–26]. Different types of sensors, such as sonar sensors [27] and ultrasonic sensors [28, 29], are used for this purpose. These methods are used for local navigation and focus mainly on obstacle avoidance. Visual servoing [30–32] belongs to the same family of methods. For a survey on the literature on visual
servoing navigation, see Ref. [32]. Visual servoing methods may be slow in some situations due to the processing of the huge amount of data coming from the camera.

Our aim in this paper is to address the problem of robot navigation and obstacle avoidance using a simple and effective model-based control law. The method can be used for both online and offline navigation and obstacle avoidance. It can also be used in indoor and outdoor environments as well, especially to reach distant goals. These goals may be out of the sensors range of view, but their position is known to the robot. Our method consists of a new family of methods for robot navigation based on the proportional navigation law, where we use the robot kinematics equations integrated with geometric rules. The proportional navigation guidance law is a navigation method well known and widely discussed in the aerospace community [33–36]. Several attempts to use proportional navigation for robotics applications have been made, mainly for the interception of moving objects using robots [37] or robotic arms [38–40]. In Ref. [33], the authors discussed a unified approach to different variants of proportional navigation, which allows us to elaborate a common basis for the analysis of these variants. In Ref. [34], the authors presented a nice comparison between the most important variants of proportional navigation, i.e. the true and the pure proportional navigation. In Ref. [35], the author proved the zero miss distance for pure proportional navigation. This problem was studied extensively in other references, but without satisfactory success. In Ref. [36], the authors discussed the optimality of proportional navigation, where according to the authors, proportional navigation is optimal for certain values of its parameters. In Ref. [37], a new formulation of proportional navigation is used for tracking a moving object using a wheeled mobile robot. The paper by Piccardo and Hondred [38] is among the first papers to adopt proportional navigation to solve robotic problems by applying the method to catch a moving object using a robotic arm. Mehrandezh et al. [39, 40] addressed a similar problem and proved their results rigorously. In Ref. [40], two different variants of proportional navigation were used.

Even though proportional navigation is a well-known method, the application of proportional navigation to wheeled mobile robots navigation problems is not straightforward, and requires major additions and modifications. First, the formulation we present in this paper is different from the classical formulation. Here, the robot’s control input is the robot’s steering angle and the proportional navigation is written in terms of the robot’s steering angle. This formulation is more suitable for wheeled mobile robot navigation applications than the classical formulation, where proportional navigation is expressed in terms of the lateral acceleration. Also, our formulation of proportional navigation can be easily adapted to the collision avoidance mode, since the proportional navigation is written as a function of the robot’s steering angle. This allows a quick change in the robot’s path using the proportional navigation itself.

This paper is organized as follows. In Section 2, we discuss the statement of the problem and the motivations. In Section 3, we present the geometry and the
kinematics equations, where we derive a polar representation of the kinematics equations. In Section 4, we discuss the control law and present our main theoretical results. In Section 5, the solution of the path in polar coordinates is derived. In Section 6, the obstacle avoidance modes are discussed. In Section 7, an extensive simulation is carried out to illustrate the method.

2. STATEMENT OF THE PROBLEM AND MOTIVATIONS

Given the initial position of the robot and the position of the goal, our objective is to design a control law which allows the robot to reach the goal and avoid obstacles. The method can be adapted to both online and offline strategies. As we mentioned previously, our solution is based on the proportional navigation law. We are motivated by the fact that this method is a powerful method that presents a good solution to the navigation and obstacle avoidance problem. The method uses a different principle and can be particularly interesting in the following cases:

(i) **Long-range navigation**. The method is highly appropriate for long-range navigation, since the control loop requires only the position of the goal. In this case, sensor-based methods (e.g. vision) fail when the goal is out of the view range of the sensors.

(ii) **Obstacle avoidance**. In some situations, such as the case when a large obstacle appears in the line of sight robot–goal, the method allows the robot to reach the goal and avoid the obstacle using curved paths, while other classical methods, e.g. the potential field method, fail.

(iii) **Outdoor navigation**. Even though the method can be used for indoor navigation, it may be more interesting for outdoor environments. This is due mainly to the lack of navigation methods used for outdoor environments compared with indoor environments.

(iv) Unlike most navigation methods, the method can be adapted to both global and local navigation.

The robustness of the method is also a critical issue. It is important to note that the method belongs to a family of methods that are based on geometric rules and kinematics equations. These methods are well known for their robustness.

3. KINEMATICS AND GEOMETRY

Let \( O \) be the origin of the inertial frame of coordinates. The mobile robot moves in a two-dimensional workspace cluttered with obstacles. The robot is a simple wheeled mobile robot moving according to the following kinematics equations:

\[
\begin{align*}
\dot{x}_R &= v_R \cos \theta_R \\
\dot{y}_R &= v_R \sin \theta_R \\
\dot{\theta}_R &= \omega_R,
\end{align*}
\]
Figure 1. Robot and goal representation in the inertial frame of reference.

where \((x_R, y_R)\) are the robot coordinates in the inertial frame of coordinates, \(\theta_R(t)\) is the robot orientation angle with respect to the reference line (parallel to the \(x\)-axis), and \(v_R\) and \(\omega_R\) are the robot linear and angular velocities, respectively. We assume that the robot control inputs are \((v_R, \theta_R(t))\). The coordinates of the reference point of the goal in the inertial frame of coordinates are given by \((x_G, y_G)\). The aim is to design a control that allows the robot to reach the goal and avoid possible collision with obstacles. With reference to Fig. 1, we define the following geometric quantities:

(i) The line of sight \(L_{GR}\): this is the imaginary straight line which starts at the robot’s reference point and is directed towards the goal reference point.

(ii) The line of sight angle \(\sigma_{GR}\): this is the angle between the reference line and the line of sight.

(iii) The distance vectors \(r_R\) and \(r_G\): these are the distance vectors from the origin of the inertial frame of coordinates to the robot and the goal, respectively.

(iv) The robot and the goal line of sight angles \(\sigma_R\) and \(\sigma_G\): these are the angles from the reference line to \(r_R\) and \(r_G\), respectively.

(v) The relative range \(r_{GR}\): this vector represents the relative distance between the robot and the goal, with \(r_{GR} = \sqrt{(y_G - y_R)^2 + (x_G - x_R)^2}\).

The line of sight angle is given by:

\[
\tan \sigma_{GR} = \frac{y_G - y_R}{x_G - x_R}.
\]
In this paper, polar coordinates are used to model the kinematics equations. This simplifies the analysis of the control law. Polar coordinates are used in various situations to design control laws for wheeled mobile robots [41–43] and underwater vehicles [44]. Consider the following change of variables:

\[ x = r \cos \sigma, \quad y = r \sin \sigma. \]  

(3)

The time derivatives of \( r \) (the radial variable) and \( \sigma \) (the angular variable) are given by

\[ \dot{r} = \frac{x \dot{x} + y \dot{y}}{r}, \]

(4)

\[ \dot{\sigma} = \frac{x \dot{y} - y \dot{x}}{r^2}, \]

(5)

respectively. By using the robot’s kinematics equations and the change of variable given by (3), (4) becomes:

\[ \dot{r}_R = \frac{r_R \cos(\sigma_R) v_R \cos(\theta_R) + r_R \sin(\sigma_R) v_R \sin(\theta_R)}{r_R} = v_R [\cos(\sigma_R) \cos(\theta_R) + \sin(\sigma_R) \sin(\theta_R)]. \]  

(6)

(7)

By using a simple trigonometric transformation, we get:

\[ \dot{r}_R = v_R \cos(\sigma_R - \theta_R). \]  

(8)

In a similar way we obtain for (5):

\[ \dot{\sigma}_R = \frac{r_R \cos(\sigma_R) v_R \sin(\theta_R) - r_R \sin(\sigma_R) v_R \cos(\theta_R)}{r_R^2} = v_R \frac{[\cos(\sigma_R) \sin(\theta_R) - \sin(\sigma_R) \cos(\theta_R)]}{r_R}, \]  

(9)

(10)

which gives:

\[ r_R \dot{\sigma}_R = v_R [\cos(\sigma_R) \sin(\theta_R) - \sin(\sigma_R) \cos(\theta_R)]. \]  

(11)

By using a simple trigonometric transformation, we get:

\[ r_R \dot{\sigma}_R = v_R \sin(\sigma_R - \theta_R). \]  

(12)

Equations (8) and (12) describe the motion of the robot in polar coordinates with respect to the origin. In fact, \( \dot{r}_R^\parallel = \dot{r}_R \) and \( \dot{r}_R^\perp = r_R \dot{\sigma}_R \) represent the components of the robot’s velocity along and across the line of sight origin–robot (along the vector of \( r_R \)). In a very similar way, it is possible to obtain the components of the robot velocity along and across the line of sight robot–goal \( L_{GR} \), which are given as follows:

\[ \dot{r}_R^\parallel = v_R \sin(\theta_R - \sigma_{GR}) \]

\[ \dot{r}_R^\perp = v_R \cos(\theta_R - \sigma_{GR}). \]  

(13)
Consider the vector for the relative distance robot-goal given by:
\[ \mathbf{r}_{GR} = \mathbf{r}_G - \mathbf{r}_R. \]  
(14)

Clearly \( \mathbf{r}_{GR} \) lies on the line of sight \( L_{GR} \). By taking the derivative of (14) we get:
\[ \dot{\mathbf{r}}_{GR} = \dot{\mathbf{r}}_G - \dot{\mathbf{r}}_R. \]  
(15)

The components of \( \dot{\mathbf{r}}_{GR} \) along and across \( L_{GR} \) are given by:
\[ \dot{\mathbf{r}}_{GR}^\parallel = \dot{\mathbf{r}}_{GR} \]  
(16)
\[ \dot{\mathbf{r}}_{GR}^\perp = \mathbf{r}_{GR} \dot{\sigma}_{GR}. \]  
(17)

Since the goal is not moving, we have \( \dot{\mathbf{r}}_G = 0 \) and thus \( \dot{\mathbf{r}}_{GR} = -\dot{\mathbf{r}}_R \). The components of \( \dot{\mathbf{r}}_R \) along and across \( L_{GR} \) are given by (13). Thus, the values of the components of \( \dot{\mathbf{r}}_{GR} \) are given by:
\[ \dot{\mathbf{r}}_{GR} = -v_R \cos(\theta_R - \sigma_{GR}), \]  
(17)
\[ r_{GR} \dot{\sigma}_{GR} = -v_R \sin(\theta_R - \sigma_{GR}). \]  
(18)

Equation (17) gives the variation of the relative range between the robot and the goal, and (18) gives the rate of turn of the line of sight angle between the robot and the goal. Equations (17) and (18) are used in this paper to model the navigation problem. In the next section, we introduce and discuss our control strategy, which is based on the proportional navigation law.

4. PROPORTIONAL NAVIGATION LAW

The proportional navigation law is a closed loop control law used for real-time tracking and pursuit of moving objects. This law is widely used in the aerospace community [33–36]. This is due to its simplicity and effectiveness. Different aspects of proportional navigation are discussed in the literature, such as the optimality of the control law [35]. The application of proportional navigation to robotics is relatively recent [38–40], where proportional navigation is mainly used for the interception of moving objects using a robotic arm. To the best of our knowledge, proportional navigation has not been used for wheeled mobile robot navigation towards a static goal and obstacle avoidance. There exist various definitions of the proportional navigation law. In the aerospace context, proportional navigation is defined in terms of the acceleration of the pursuer, where the acceleration is proportional to the line of sight angle rate. In this paper, we define the proportional navigation law in a simple way in terms of the robot orientation angle as follows:
\[ \theta_R(t) = N \sigma_{GR}(t) + a, \]  
(19)

where \( N \) is a real constant called the navigation constant with \( N \geq 1 \) and \( a \) is a real number representing the deviation angle. This angle refers to the initial state...
in some situations. Under the proportional navigation law, different values of $N$ result in different paths. The robot moves in a straight line when $N = 1$. When $N$ increases, the robot path becomes more curved, where the robot performs long turns. This property allows the robot to turn around large obstacles. Under proportional navigation, the robot kinematics equations are given by:

$$\dot{x}_R = v_R \cos(N \sigma_{GR} + a)$$
$$\dot{y}_R = v_R \sin(N \sigma_{GR} + a)$$
$$\dot{\theta}_R = \omega_R,$$

and in polar coordinates:

$$\dot{r}_R = v_R \cos(N \sigma_{GR} + a - \sigma_R)$$
$$r_R \dot{\sigma}_R = v_R \sin(N \sigma_{GR} + a - \sigma_R).$$

Recall that $\sigma_{GR}$ is given by (2). By considering the relative motion robot–goal, the relative kinematics equations under the proportional navigation are as follows

$$\dot{r}_{GR} = -v_R \cos(M \sigma_{GR} + a)$$
$$r_{GR} \dot{\sigma}_{GR} = -v_R \sin(M \sigma_{GR} + a),$$

where $M = N - 1$. From the robot kinematics equations under proportional navigation, it can be seen that the robot path under this control law depends on the navigation constant $N$ and the deviation angle $a$. Different paths are obtained for different values of $N$ and $a$. Since $N$ and $a$ are real, an infinite number of paths is possible. By taking the time derivative in (19), the robot’s angular velocity is obtained as follows:

$$\omega_R = N \dot{\sigma}_{GR}.$$  \hspace{1cm} (23)

It is worth noting that the proportional navigation law as defined in this paper has a periodic aspect with the navigation constant. That is, the same value for the robot orientation angle is obtained for different values of the navigation constant, i.e. the orientation angle of the robot can be written as

$$\theta_R(t) = N_1 \sigma_{GR}(t) + 2\pi m = N \sigma_{GR},$$  \hspace{1cm} (24)

with $m = 0, 1, 2, \ldots$ and $N_1 \leq N$. Thus, the robot angular velocity is written as $\omega_R = N \dot{\sigma}_{GR}$. This becomes important when the constraint on the maximum value of the robot’s angular velocity is taken into account. The particular case when $N = 1$ corresponds to pursuit. When $a = 0$, it corresponds to pure pursuit, where the robot’s linear velocity vector lies on the line of sight robot–goal and the robot’s orientation angle is equal to the line of sight angle. The case when $a \neq 0$ corresponds to deviated pursuit. In both cases the robot angular velocity is equal to the rate of turn of the line of sight. This particular case is addressed in Ref. [45] in more details for the navigation towards a moving object. The approach is integrated with a dynamic polar-like histogram to avoid obstacles.
Results concerning robot navigation towards a fixed point using the proportional navigation are stated as follows.

**PROPOSITION 1.** In the case of pure pursuit (proportional navigation with \( N = 1, a = 0 \)), the robot reaches the goal from any initial state.

**Proof.** The proof is simple if we consider the relative kinematics model for \( N = 1, a = 0 \), which gives for the relative range:

\[
\dot{r}_{GR} = -v_R. \tag{25}
\]

Since \( \dot{r}_{GR} < 0 \), the relative range function is decreasing and the robot reaches the goal successfully, with a final orientation angle \( \theta_R(t_f) = \sigma_{GR}(t_0) \). In this particular case, the robot moves in a straight line as \( \dot{\sigma}_{GR} = 0 \).

**PROPOSITION 2.** In the case of deviated pursuit (proportional navigation with \( N = 1, a \neq 0 \)), the robot reaches the goal when:

\[
a \in \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right]. \tag{26}
\]

**Proof.** By writing the first equation in the relative kinematics model, we get:

\[
\dot{r}_{GR} = -v_R \cos a. \tag{27}
\]

\( r(t) \) is a decreasing function when \( a \in \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right] \).

When \( v_R \) is constant, robot navigating under the proportional navigation law with \( N = 1, a = 0 \) reaches the goal at time:

\[
t_f = \frac{r_{GR}(t_0)}{v_R}. \tag{28}
\]

In a similar way, the robot navigating under deviated pursuit with constant speed reaches the goal at time:

\[
t_f = \frac{r_{GR}(t_0)}{v_R \cos a}. \tag{29}
\]

**PROPOSITION 3.** For \( N \neq 1 \), the robot navigating under the control law (19) reaches the goal for almost all initial states.

**Proof.** Similar to the previous cases, our aim is to prove that the relative range is a decreasing function. The equation for the line of sight angle rate can be written as follows:

\[
r_{GR} \dot{\sigma}_{GR} = -v_R \sin(M \sigma_{GR} + a). \tag{30}
\]

The equilibrium positions for (30) are given by:

\[
\sigma_{GR} = \sigma_{eq1} = \frac{\pi + 2\pi n - a}{M} \tag{31}
\]

\[
\sigma_{GR} = \sigma_{eq2} = \frac{2\pi n - a}{M}. \tag{32}
\]
where $n$ is an integer. From linearization near these equilibrium positions, we get:

\[
\begin{align*}
\frac{\partial \dot{\sigma}_{GR}}{\partial \sigma_{GR}} \bigg|_{\sigma_{GR}=\sigma_{eq1}} &= a_1 = \frac{v_R M}{r_{GR}} \\
\frac{\partial \dot{\sigma}_{GR}}{\partial \sigma_{GR}} \bigg|_{\sigma_{GR}=\sigma_{eq2}} &= a_2 = -\frac{v_R M}{r_{GR}},
\end{align*}
\]

respectively. From the signs of $a_1$ and $a_2$, equilibrium positions situated at $\sigma_{eq1}$ are unstable and equilibrium positions situated at $\sigma_{eq2}$ are asymptotically stable. Since the solutions for the line of sight angle go to their asymptotically stable equilibrium positions, we get $\sigma_{GR} \to \sigma_{eq2} = \frac{(2\pi n - a)}{M}$ with time. Since $\sigma_{GR} \to (2\pi n - a)/M$ with time, $\cos(M\sigma_{GR} + a)$ is positive in a given time interval $[t_1, t_f]$, $t_1 \geq t_0$; and thus, after $t_1$, we have $\dot{r}_{GR} < 0$. The only case where the relative range is not decreasing is when the robot starts from an initial state that satisfies $M\sigma_{GR}(t_0) + a = \pi + 2n\pi$. In this particular case the robot does not reach the goal. Therefore, it is necessary to use appropriate values for $M$ and $a$. □

We have already mentioned that there exists an infinite number of possible paths under proportional navigation, where the paths can be very different for different values of the guidance parameters. This property can be used for both offline and online obstacle avoidance by choosing the values of the navigation parameters, which result in obstacle-free paths. This property will be discussed later.

The application of proportional navigation requires the following equation to be satisfied at the initial time:

\[
\theta_R(t_0) = N\sigma_{GR}(t_0) + a.
\]

In order to satisfy this equation, two approaches can be used:

(i) Choose the values of $N$ and $a$ such that constraint (35) on the initial conditions is satisfied with $N \geq 1$.

(ii) Apply a heading regulation which drives the robot’s orientation angle from its initial value to the value which satisfies (35) for the values of $N$ and $a$ being used. This approach is recommended, because it gives more flexibility for the choice of $N$ and $a$.

An illustration is shown in the following example.

**Example 1.** In order to illustrate these approaches, we consider the scenario of Fig. 2. The robot’s initial coordinates are (20, 0), with $\theta_R(t_0) = 225^\circ$. The robot aims to reach a goal situated at (20, 20) and thus $\sigma_{GR}(t_0) = 90^\circ$. The solutions based on the two approaches discussed above are as follows.

(i) **For the first approach.** Since $\theta_R(t_0) = 225^\circ$ and $\sigma_{GR}(t_0) = 90^\circ$, $N$ and $a$ are calculated such that:

\[
N\frac{\pi}{2} + a = \frac{5\pi}{4}.
\]
There exists an infinite number of solutions for \((N, a)\) which satisfies system (36) and therefore there exists an infinite number of possible paths for the robot. For example, we can take

\[
N = 3 \quad \rightarrow \quad a = -\frac{\pi}{4} \text{ rad}
\]
\[
N = 4 \quad \rightarrow \quad a = -\frac{3\pi}{4} \text{ rad}
\]

Note that the path of the robot is predefined by the initial conditions and the values of \((N, a)\). The robot navigation using this approach is illustrated in Fig. 3.

(ii) For the second approach. The values of \(N\) and \(a\) are predetermined, thus a heading regulation phase is necessary before the application of proportional navigation. The aim of the heading regulation is to take the robot orientation angle from its initial value to the intermediary value \(\theta_{R0}\) that satisfies (35). Let us take the predetermined values of \(N\) and \(a\) as \(N = 2, a = 0\). Let \((x^i_R, y^i_R, \theta^i_{R0})\) be the robot’s configuration at which the proportional navigation is applied. There exist various possibilities for the choice of \((x^i_R, y^i_R)\). \(\theta^i_{R0}\) is calculated based on (35). We can take for example:

\[
(x^i_R, y^i_R) = (10, 0) \quad \rightarrow \quad \theta^i_{R0} = 2.21 \text{ rad}
\]
\[
(x^i_R, y^i_R) = (6, 0) \quad \rightarrow \quad \theta^i_{R0} = 1.92 \text{ rad}
\]

Various techniques from control theory can be used for the purpose of heading regulation. The robot navigation using this approach is shown in Fig. 4. The robot path during the heading regulation phase is shown in dashed lines.
Figure 3. An illustration of the first approach, where the initial configuration is satisfied by the choice of the control law parameters.

Figure 4. An illustration of the second approach, where a heading regulation is used. The robot’s path during the heading regulation phase is shown in dashed lines.
5. SOLUTION IN THE PLANE \((r_{GR}, \sigma_{GR})\) AND FINAL VALUE FOR THE LINE OF SIGHT ANGLE

Let us rewrite the relative kinematics equations modeling the navigation problem under proportional navigation:

\[
\dot{r}_{GR} = -v_R \cos(M\sigma_{GR} + a) \\
\dot{r}_{GR}\sigma_{GR} = -v_R \sin(M\sigma_{GR} + a).
\]  

(39)

Dividing the radial velocity by the tangential velocity, we get:

\[
\frac{dr_{GR}}{d\sigma_{GR}} = r_{GR} \frac{\cos(M\sigma_{GR} + a)}{\sin(M\sigma_{GR} + a)}.
\]  

(40)

Integrating equation (40) we get:

\[
r_{GR} = r_{GR0} \exp\left(\int_{\sigma_{GR0}}^{\sigma_{GR}} \frac{d\sigma_{GR}}{\tan(M\sigma_{GR} + a)}\right),
\]  

(41)

where \(r_{GR0} = r_{GR}(t_0)\) and \(\sigma_{GR0} = \sigma_{GR}(t_0)\) are the initial values of the relative range and the line of sight angle, respectively. The solution for the relative distance as a function of the line of sight angle is given by:

\[
r_{GR} = r_{GR0} \left(\frac{\sin(M\sigma_{GR} + a)}{\sin(M\sigma_{GR0} + a)}\right)^{1/M},
\]  

(42)

for \(M > 0\) (this equation is not valid for the pursuit). This equation gives the variation of the relative range as a function of the line of sight angle. Note that the robot’s path is a function of the initial states and the control law parameters, and does not depend on the robot speed. It follows that \(r_{GR} = 0\) when \(\sin(M\sigma_{GR} + a) = 0\).

The robot navigating under proportional navigation reaches the goal with a given value for the line of sight angle \(\sigma_{GR}(t_f)\). This value can be determined in general, except when \(M \approx 0\) (which corresponds to the case when proportional navigation acts like pursuit). Consider the following change of variable:

\[
x = M\sigma_{GR} + a,
\]  

(43)

which gives \(\dot{x} = M\dot{\sigma}\). Note that \(x(t) = \theta_R(t) - \sigma_{GR}(t)\). The equation for the rate of the line of sight angle gives:

\[
\alpha \dot{x} = -\sin(x) = f(x),
\]  

(44)

with \(\alpha = r/(Mv_R)\). The real number \(\alpha\) is positive, since \(r(t)\), \(M\) and \(v_R\) are always positive. The dynamics of system (44) depend on \(f(x)\), which is a periodic function with an infinite number of asymptotically stable equilibrium positions situated at \(0, \pi + 2\pi n (n = \ldots, -2, -1, 0, 1, 2, \ldots)\). If the initial state for \(x\) satisfies \(x_0 \in (2n - 1)\pi, (2n + 1)\pi\), then \(x(t) \to 2\pi n\); which is equivalent to \(\sigma_{GR} \to (2\pi n - a)/M\).
Let us restrict the interval for the initial state of \( x \) to \( x_0 \in [0, 2\pi] \). From Fig. 5 representing the vector field of system (44), it can be deduced that:

(i) \( x(t) \to 0 \), when \( x_0 \in [0, \pi] \), since \( x_0 \in [0, \pi] \) is in the attraction domain of the equilibrium position \( x_{eq} = 0 \). For the line of sight angle, it results that \( \sigma_{GR}(t) \to -a/M \), when \( \theta_R(t_0) - \sigma_{GR}(t_0) \in [0, \pi] \).

(ii) \( x(t) \to 2\pi \), when \( x_0 \in [\pi, 2\pi] \), since \( x_0 \in [\pi, 2\pi] \) is in the attraction domain of the equilibrium position \( x_{eq} = 2\pi \). For the line of sight angle, it results that \( \sigma_{GR}(t) \to (2\pi - a)/M \), when \( \theta_R(t_0) - \sigma_{GR}(t_0) \in [\pi, 2\pi] \).

6. NAVIGATION IN THE PRESENCE OF OBSTACLES

The problem of navigation becomes more difficult in the presence of obstacles. Our control strategy can be integrated with various obstacle avoidance algorithms. In this paper we suggest two different strategies based on proportional navigation.

6.1. Obstacle modeling

Let \( \chi_i \) be an arbitrarily-shaped obstacle detected by the robot sensory system. Obstacle \( \chi_i \) is enclosed in a circle \( C_i \). Circle \( C_i \) is enlarged by the robot radius to give the final representation of the obstacle denoted by \( F_i \). Circle \( F_i \) has \( d \) as a radius. Under this representation the robot is seen as a point vehicle. Consider Fig. 6 where the obstacle appears in the line of sight robot–goal. The distance from the robot’s reference point to the center of the obstacle is \( r_i \) and the angle of the line of sight robot–\( F_i \) is \( \sigma_i \). Let points \( A_0 \) and \( A_1 \) be as shown in Fig. 6. The distances from the robot to point \( A_0 \) and point \( A_1 \) are given by \( r_{i1} \) and \( r_{i2} \), respectively.
angles of the line of sight are given by $\sigma_{i1}$ and $\sigma_{i2}$. The relative kinematics model between the robot and the center of obstacle $F_i$ are given by:

$$\begin{align*}
\dot{r}_i &= -v_R \cos(N\sigma_{GR} + a - \sigma_i) \\
\dot{\sigma}_i &= -v_R \sin(N\sigma_{GR} + a - \sigma_i).
\end{align*}$$

(45)

In a similar way, the kinematics equations between the robot and points $A_0$ or $A_1$ are given by:

$$\begin{align*}
\dot{r}_{ik} &= -v_R \cos(N\sigma_{GR} + a - \sigma_{ik}) \\
\dot{\sigma}_{ik} &= -v_R \sin(N\sigma_{GR} + a - \sigma_{ik}),
\end{align*}$$

(46)

with $k = 1, 2$. It is easy to determine whether the robot is approaching or moving away from the obstacle using (45) and (46). The robot is on a collision course with the obstacle when:

$$\theta_R(t) \in [\sigma_{i1}, \sigma_{i2}],$$

(47)

when $\sigma_{i1} < \sigma_{i2}$. Otherwise, we exchange $\sigma_{i1}$ and $\sigma_{i2}$. If we consider the proportional navigation control law, the collision course is given by:

$$N\sigma_{GR} + a \in [\sigma_{i1}, \sigma_{i2}].$$

(48)

The aim is to choose the navigation constant $N$ and $a$ so that:

$$N\sigma_{GR} + a \notin [\sigma_{i1}, \sigma_{i2}],$$

(49)

when the robot is within a certain distance ($r_0$) from the obstacle. In order to avoid an obstacle, two different approaches can be used, i.e., predetermined path and online deviation.
6.2. Predetermined path

This approach is used when the information about the workspace is prelearned by specifying the geometric features of the obstacles (e.g. the position and the radius) and the position of the goal. Given the necessary information, a collision-free trajectory can be found offline by calculating the appropriate values for $N$ and $a$. These values allow the robot to turn around the obstacle and reach the goal. An illustration is shown in Fig. 6, where an obstacle appears in the line of sight robot-goal. To avoid the obstacle, the robot must deviate to the right towards points $B_1, B_3, \ldots$ or to the left towards points $B_0, B_2, \ldots$

Given the value of $\sigma_{GR}(t_0)$ and a desired value for $a$, the aim is to compute the value of $N$, which allows the robot to reach the goal by passing through a point $B_i$. Points $B_i$ are characterized by a given distance from the goal $r_{GB_i}$ and a given value for the line of sight angle $\sigma_{GB_i}$. $r_{GB_i}$ and $\sigma_{GB_i}$ are known, since the coordinates of points $B_i$ are assumed to be known. Equation (42) is also satisfied for points $B_i$ as they are in the robot’s path. The graphical solution of the nonlinear system:

$$M \ln \frac{r_{GB_i}}{r_{GR0}} = \ln \left( \frac{\sin(M\sigma_{GB_i} + a)}{\sin(M\sigma_{GR0} + a)} \right)$$

allows us to determine the appropriate values of the control parameters that allow the robot to reach the goal by passing through a point $B_i$. This is illustrated in our simulation.

6.3. Online deviation

In general, there is no a priori knowledge about the obstacles and their geometric distribution, and only the initial positions of the robot and the goal are given. The aim is to design an online collision-free path using proportional navigation by using information obtained from the sensory system. In the online approach, we construct a polar histogram, which provides obstacle directions and free directions. The polar histogram is constructed by considering all the obstacles in the active region as follows:

$$p_i = \begin{cases} 1 & \theta_R(t) \in [\sigma_{i1}, \sigma_{i2}] \\ 0 & \text{otherwise} \end{cases}$$

with $P = \bigcup_{i=1}^{K} p_i$, where $K$ is the total number of obstacles in the active region (within a specific distance from the goal). After the histogram is constructed, a point $G_1$ that represents an intermediary goal is chosen. Point $G_1$ corresponds to a free direction and is characterized by a given range and line of sight angle from the position of the robot. Let point $G_0$ be the point where the robot starts deviating from a possible obstacle. We suggest the following algorithm:

Navigation mode:

(i) Construct the polar histogram.
(ii) Navigate the robot using control law (19) if no obstacle appears in the robot’s path.

(iii) If the robot is in a collision course with an obstacle with \( r_i - d < r_0 \), then the obstacle avoidance mode is activated.

Obstacle avoidance mode:

The robot is initially navigating towards point \( G \) (final goal). In the obstacle avoidance mode, the aim is to drive the robot to an intermediary goal that appears in a free direction in the histogram. Let \( \sigma_{GR0}^0 \) be the angle of the line of sight robot–goal measured at point \( G_0 \) at time \( t_1^0 \), and \( \sigma_{GR0}^1 \) the angle of the line of sight robot–point \( G_1 \) measured at point \( G_0 \) at the same time. \( \sigma_{GR0}^0 \) and \( \sigma_{GR0}^1 \) are given by:

\[
\tan \sigma_{GR0}^0 = \frac{y_G - y_R(t_1^0)}{x_G - x_R(t_1^0)} \quad (52)
\]

\[
\tan \sigma_{GR0}^1 = \frac{y_G - y_R(t_1^0)}{x_G - x_R(t_1^0)} \quad (53)
\]

where \((x_{G1}, y_{G1})\) are the coordinates of point \( G_1 \). For the smoothness of the path, it is required that:

\[
N_0 \sigma_{GR0}^0 + a_0 = N_1 \sigma_{GR0}^1 + a_1 \quad (54)
\]

It is simple to determine values of \( N_1 \) and \( a_1 \) that satisfy (54), and move the robot to point \( G_1 \). The procedure is repeated if the robot meets other obstacles in its path.

In the next section we present an extensive simulation study.

7. SIMULATION

Several examples illustrating navigation based on our control strategy are considered in our simulation. Different aspects are also taken into account. It is assumed that the position, time and speed are without units. This assumption simplifies the analysis and does not affect the simulation results.

7.1. Navigation towards different points

Control law (19) allows the robot to reach goals at different positions starting from the same initial state and using the same control parameters. This is shown in Fig. 7, where the robot navigates towards different goals: \( G(25, 25) \), \( G(30, 30) \), \( G(35, 35) \), starting from the origin with \( \theta_R(t_0) = 132.57^\circ \), and navigating using the control law (19) with \( N = 3, a = 0 \).

7.2. Path symmetry

Figure 8 shows the robot paths for \( N = 3, a = \pm \frac{\pi}{3} \). The goal is situated at \((20, 20)\) and the robot starts approximately from point \((20, 0)\). The deviation angle
Figure 7. Robot navigation towards different goals starting from the same initial state and using the same control parameters, $N = 3$, $a = 0$.

Figure 8. Symmetric robot paths.

$a = -\frac{\pi}{3}$ results in a left detour, and the deviation angle $a = \frac{\pi}{3}$ results in a right detour. Figure 9 shows the line of sight angles as a function of time for both cases. For the left detour the robot reaches the goal with a final line of sight angle $\sigma_{GR}(t_f) = -a/M = 30^\circ$ and for the right detour the robot reaches the goal with a
Figure 9. Time evolution of the line of sight (LOS) angle for the scenario of Fig. 8.

final line of sight angle \( \sigma_{GR}(t_f) = (2\pi - a)M = 150^\circ \). The line of sight angles also present a symmetry.

7.3. Obstacle avoidance using path deviation

Here, we present an important particular case for obstacle avoidance, where a large obstacle appears in the line of sight robot–goal as shown in Fig. 6. Our aim is to show how to use the navigation parameters to avoid the obstacle and reach the goal. Note that various classical methods fail in this case. For example, this case corresponds to a local minimum for the potential field method (as shown in Fig. 10a). A similar problem appears for the VFF (Fig. 10b), where occupied cells exert repulsive forces onto the robot and the magnitude is proportional to the certainty value of each cell. Both the repulsive force \( F_r \) due to the cells and the attractive force \( F_a \) due to the goal are on the line of sight robot-goal, and therefore the resulting force \( \mathbf{R} = \mathbf{F}_r + \mathbf{F}_a \) lies on the line of sight too. Thus, if \( |\mathbf{F}_r| < |\mathbf{F}_a| \), the robot moves towards the obstacle, and if \( |\mathbf{F}_r| > |\mathbf{F}_a| \), the robot moves away from both the obstacle and the goal. We compare our method with the path obtained using adaptive navigation [21] and the path obtained using straight lines. The obstacle is a rectangle that covers the area \([4, 7] \times [-2, 2] \). The goal is situated at point \((0, 0) \) and the robot starts from point \((10, 0) \). The paths are shown in Fig. 11. The robot under adaptive navigation reaches the goal. However, the method results in a half-free path, where the robot touches the corner of the obstacle. The path obtained by using the proportional navigation is also shown in Fig. 11, where the values of the control parameters are \( N = 3.25, a = 35^\circ \). The obstacle is avoided by using a left detour.
Figure 10. (a) Potential field and (b) VFF methods when the obstacle lies in the line of sight robot–goal.

Figure 11. Obstacle avoidance, comparison between proportional navigation (PN) for $N = 3.25$, $\alpha = \pm 35^\circ$ and the adaptive navigation method.

The path symmetry property results in a right detour for $N = 3.25$, $\alpha = -35^\circ$ as shown in Fig. 11. The solution based on straight lines is shown in Fig. 12. Even though this solution is successful, it presents two disadvantages:
Figure 12. Obstacle avoidance, comparison with linear segments motion (dashed lines).

(i) The path is composed of at least two segments, which means two phases for the navigation process. The number of segments may become larger for bigger obstacles.

(ii) Smoothing the path between each two segments is necessary.

Navigation using proportional navigation is accomplished in one phase using the same control inputs and no smoothing is required. The choice of the navigation constant is very important. As we mentioned previously, small values of $N$ result in pursuit-like behavior, and large values of $N$ result in more curved path and long turns. This means that small values of $N$ are more appropriate when there is no obstacle between the robot and the goal, and large values of $N$ are more appropriate in the presence of large obstacles, since large values of $N$ allow long turns.

7.4. Obstacle avoidance using path deviation towards a given point

We saw in the previous example that our navigation law allows the robot to avoid the obstacle through deviation by controlling the navigation law parameters. Here, our aim is to show the possibility to navigate the robot towards a given point $B_i$ in the path and reach the goal by choosing the control parameters (for simplicity we assume that the deviation angle is equal to zero). Equation (50) is used to compute the navigation constant that allows the robot to reach points $B_i$. The coordinates of these points are assumed to be known.

Tables 1 and 2 show the values of the navigation constant $N$ obtained using (50) and the corresponding coordinates for points $B_i$. Table 1 corresponds to a left detour and Table 2 corresponds to a right detour. Figure 13 shows the robot path obtained using simulation. It is assumed for each case that (19) is satisfied initially. To avoid larger obstacles, a deviation towards the extreme right ($B_5$) or the extreme left ($B_4$) is used.
Table 1. Values of $N$ which correspond to a left detour

<table>
<thead>
<tr>
<th>Points</th>
<th>Coordinates</th>
<th>$\left(\frac{r_{GB_i}}{\sigma_{GB_i}}\right)$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0$</td>
<td>(40, 25)</td>
<td>$\left[26.93^* 68.13^\circ\right]$</td>
<td>1.402</td>
</tr>
<tr>
<td>$B_2$</td>
<td>(30, 25)</td>
<td>$\left[32.02^* 51.34^\circ\right]$</td>
<td>1.814</td>
</tr>
<tr>
<td>$B_4$</td>
<td>(20, 25)</td>
<td>$\left[39.05^* 39.81^\circ\right]$</td>
<td>2.166</td>
</tr>
</tbody>
</table>

Table 2. Values of $N$ which correspond to a right detour

<table>
<thead>
<tr>
<th>Points</th>
<th>Coordinates</th>
<th>$\left(\frac{r_{GB_i}}{\sigma_{GB_i}}\right)$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>(60, 25)</td>
<td>$\left[26.93^* 111.88^\circ\right]$</td>
<td>4.1488</td>
</tr>
<tr>
<td>$B_3$</td>
<td>(70, 25)</td>
<td>$\left[32.02^* 128.66^\circ\right]$</td>
<td>3.679</td>
</tr>
<tr>
<td>$B_5$</td>
<td>(80, 25)</td>
<td>$\left[39.05^* 140.19^\circ\right]$</td>
<td>3.166</td>
</tr>
</tbody>
</table>

Figure 13. Avoidance of an obstacle that appears in the line of sight robot–goal using predetermined deviation. Navigation towards points $B_4$ or $B_5$ allows us to avoid larger obstacles.

7.5. Navigation using proportional navigation in the presence of obstacles

In the presence of obstacles, navigation using proportional navigation becomes more difficult. The robot uses a point-to-point navigation strategy to navigate
towards the goal and avoid the obstacles. The robot starts from the origin and aims to reach a goal situated at (100, 100). The navigation path towards this point is shown in Fig. 14. Online deviation towards intermediary goals \( G_1, G_2, G_3 \) and \( G_4 \) is used with different control parameters for the proportion navigation:

- Phase \( R_0G_1 \): \( N = 3, a = 0 \)
- Phase \( G_1G_2 \): \( N = 1.5, a = -1.6 \)
- Phase \( G_2G_3 \): \( N = 2, a = -1.17 \)
- Phase \( G_3G_4 \): \( N = 5, a = 0 \)
- Phase \( G_4G \): \( N = 5, a = 0 \).

The angular histogram at different intermediary goals is shown in Fig. 15. Clearly the subgoals \( G_1, G_2, G_3 \) and \( G_4 \) belong to the free workspace. These points are

Figure 14. Online deviation in the presence of obstacles.

Figure 15. Polar histogram \( P \) for the scenario of Fig. 14.
chosen so that the deviation from the nominal path is small, which allows us to keep smoothness of the path.

8. CONCLUSION

We presented a new approach for robot navigation using the proportional navigation law. The approach combines geometrical rules with the kinematics equations of the robot. We derived a navigation kinematics model in polar coordinates that simplifies the analysis of the method. The control strategy states that the robot angular velocity is proportional to the rate of turn of the line of sight angle. The robot path in polar coordinates is derived as a function of the control parameters. It turns out that different paths are obtained for different control parameters. Results of the navigation method are proven rigorously. The control law is simple and requires only the position of the goal. For obstacle avoidance, both offline and online strategies are used. The method can also be used for indoor and outdoor navigation as well, especially to reach goals that are at a long distance from the robot, and as a result they are out of the range of view of the sensors (such as the camera), but their position is known to the robot. In this case, sensor-based methods fail. Obstacles that appear in the line of sight robot–goal can be easily avoided by adjusting the control parameters. Note that some classical methods, such as the potential field method, fail in this case. Our results are confirmed using various simulation examples. Our approach opens new directions for research, such as navigation using the proportional navigation law under the robot’s kinematics and dynamics constraints, and the influence of the control law parameters, especially $N$.

REFERENCES


ABOUT THE AUTHORS

Fethi Belkhouche received the MS degree in Physical Electronics from Universite de Tlemcen in July 2001 and the MS degree in Electrical Engineering from Tulane University in December 2004. He received a PhD in Electrical Engineering from Tulane University in May 2005. He is currently a Visiting Assistant Professor in the Department of Electrical Engineering and Computer Science, Tulane University. His research interests include robotics, image processing and dynamical systems.

Boumediene Belkhouche is a Professor of Electrical Engineering and Computer Science at Tulane University. His areas of research include motion planning for autonomous mobile robots and object-oriented modeling.