Local Incremental Planning for a Car-Like Robot Navigating among Obstacles*

Alberto Bemporad

Dipartimento di Sistemi e Informatica
Università di Firenze
Via di S. Marta 3, 50139 Firenze, Italy
bemporad@dsi.ing.unifi.it

Alessandro De Luca   Giuseppe Oriolo

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
{adeluca,oriolo}@giannutri.caspur.it

Abstract

We present a local approach for planning the motion of a car-like robot navigating among obstacles, suitable for sensor-based implementation. The nonholonomic nature of the robot kinematics is explicitly taken into account. The strategy is to modify the output of a generic local holonomic planner, so as to provide commands that realize the desired motion in a least-squares sense. A feedback action tends to align the vehicle with the local force field. In order to avoid motion stops away from the desired goal, various force fields are considered and compared by simulation.

1 Introduction

Motion planning for mobile robots in the presence of obstacles usually deals with *holonomic* vehicles, that can move in any direction of the free configuration space [1]. Following a hierarchical approach, the specific kinematics of the robot is not taken into account in the higher level planning phase. With this simplifying assumption, two major approaches can be identified, namely *algorithmic* and *incremental* planning.

Methods of the first class search for a solution path in the free configuration space, directly facing the combinatorial complexity of the problem. The associated algorithms, although complete (i.e., a solution is found whenever one exists), are often difficult to implement and require a priori knowledge of the environment.

Incremental methods are heuristic in nature but can operate in a feedback mode, thus being more suitable for sensor-based navigation through partially unknown environments. With *artificial potential field* methods, the robot moves under the local effects of repulsive fields associated to obstacles and an attractive field pulling toward the goal [2]. The main limitation is the arising of *local minima* in the total potential field, where no descent direction exists for the motion.

Several modifications have been introduced to overcome this problem, such as repulsive fields with elliptic isocontours [3], biharmonic functions [4], navigation functions [5] and numerical potential fields [6]. Another method that can avoid motion stops using only *local* information is the *vortex field* method [7]: repulsive actions are replaced by vortical velocity flows so that the robot is forced to turn around the obstacles.

On the other hand, wheeled mobile robots are subject to *nonholonomic* (non-integrable) constraints involving time derivatives of the configuration variables [8]. These constraints depend on the kinematics of the vehicle and limit the local mobility without restricting in the large the accessibility of the whole free configuration space. Due to the nonholonomy, the design of feasible trajectories and of stabilizing control laws to a desired goal configuration is fairly complicated. *Open-loop* commands that exactly drive the mobile robot to the goal are easily obtained when the kinematic equations can be transformed in the so-called chained form [9], e.g., for a car with N trailers. Moreover, since continuously differentiable *feedback* schemes cannot stabilize a nonholonomic system to a given configuration [10], more complex discontinuous [11] or smooth time-varying [12] feedback control laws have been proposed. All the above techniques do not consider the presence of obstacles.

Methods that solve the planning problem for mobile robots by taking explicitly into account both the nonholonomic constraints and the presence of obstacles have been proposed in [13–16]. In [13, 14], a two-stage approach is proposed for a car-like robot: a complete path avoiding obstacles is generated first with a conventional (holonomic) planner, and then decomposed and approximated with feasible segments complying

*Work supported by EEC BR Project 6546 (PROMotion) and by MURST 40% funds

with the nonholonomic constraints. Shortest paths of bounded curvature were obtained in [15], while a discretization of the configuration space is used in [16]. A common requirement of this class of methods is the a priori knowledge of the environment, so that the solutions are inherently off-line.

There is indeed a lack of work considering all the following aspects together: *(i)* nonholonomy of the wheeled vehicle kinematics, *(ii)* presence of obstacles, *(iii)* a priori unknown environment, with local information acquired by sensors, *(iv)* on-line feedback control solution. In [17], we proposed an algorithm working under these assumptions for a robot with unicycle kinematics. Heuristic rules have been used in [18] for a car-like robot so as to guarantee collision avoidance.

In this paper we extend the results of [17], proposing a nonholonomic planner for a car-like vehicle among obstacles. Both front- and rear-wheel driving are considered in a unified model. Suitably defined artificial force fields are used as holonomic planner. Their output is filtered through the vehicle kinematics, so as to realize the desired cartesian motion in a least-squares sense. A feedback scheme is used to define reference values for the remaining free variables, in order to align the vehicle to the local field.

2 Car-like robot kinematics

We refer to the car-like robot model depicted in Fig. 1. As usual, front and rear axles are collapsed respectively in a front and rear median wheel, reducing the model to that of a *bicycle* or, equivalently, of a *unicycle with one trailer*. We assume pointwise and pure rolling contact between the ground and the wheels.

Let $P = (x, y)$ be the position of the front wheel, $P_r = (x_r, y_r)$ the position of the rear wheel, ℓ the distance between the two wheels, θ the orientation of the robot w.r.t. the x-axis, ϕ the steering angle, and

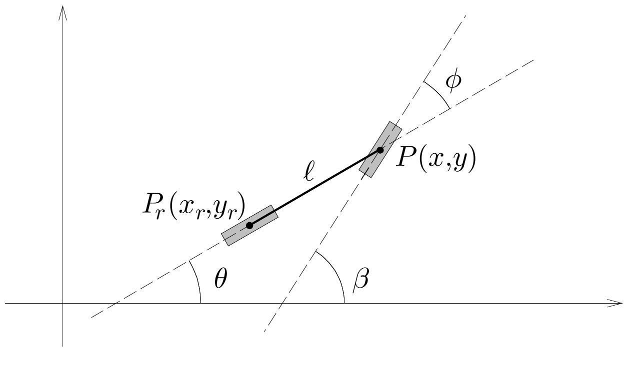

Figure 1 – Car-like robot model

$\beta = \theta + \phi$ the heading angle of the front wheel w.r.t. the x-axis. Moreover, denote by u_f and u_r the velocity of the front and rear wheel, respectively, and by u_ϕ the steering rate of the front wheel.

The coordinates of the two points P and P_r are related by the rigid body constraint

$$\begin{aligned} x &= x_r + \ell\cos\theta \\ y &= y_r + \ell\sin\theta. \end{aligned} \tag{1}$$

Depending on which wheel is active, we distinguish between *rear-wheel* and *front-wheel driving*. We are interested in obtaining a model format that is valid for both types of driving.

Rear-wheel driving. The kinematic equations can be written in the form

$$\begin{aligned} \dot{x}_r &= u_r\cos\theta \\ \dot{y}_r &= u_r\sin\theta \\ \ell\dot{\theta} &= u_r\tan\phi \\ \dot{\phi} &= u_\phi. \end{aligned} \tag{2}$$

Defining two new system inputs u_1 and u_2 through

$$u_r = u_1\cos\phi \tag{3}$$

$$u_\phi = u_2 - \frac{u_1\sin\phi}{\ell} \tag{4}$$

and using eqs. (1), model (2) can be transformed in

$$\begin{aligned} \dot{x} &= u_1\cos\beta \\ \dot{y} &= u_1\sin\beta \\ \ell\dot{\theta} &= u_1\sin(\beta-\theta) \\ \dot{\beta} &= u_2. \end{aligned} \tag{5}$$

Front-wheel driving. The kinematic equations are

$$\begin{aligned} \dot{x} &= u_f\cos(\theta+\phi) \\ \dot{y} &= u_f\sin(\theta+\phi) \\ \ell\dot{\theta} &= u_f\sin\phi \\ \dot{\phi} &= u_\phi. \end{aligned} \tag{6}$$

Letting

$$u_f = u_1 \tag{7}$$

and u_ϕ as in eq. (4), model (6) takes the form (5).

Note that, by dropping the third equation, eqs. (5) formally represent the kinematic model of a unicycle. We will develop motion control algorithms directly for model (5), independently of which is the driving wheel.

3 Control law

In general, the motion of a wheeled mobile robot is described by the kinematic model

$$\dot{X} = G(X)u, \tag{8}$$

where $X \in I\!R^n$ is the vector of generalized coordinates, and $u \in I\!R^m$ $(m < n)$ is the control input vector. Equation (8) represents an *underactuated* mechanical system, with less inputs than generalized coordinates.

Given any desired smooth trajectory $X_d(t)$ (feasible or not), a straightforward approach is to design the input command u using the pseudoinverse control law

$$u = G^{\#}(X)\dot{X}_d = [G^{\mathrm{T}}(X)G(X)]^{-1}G^{\mathrm{T}}(X)\dot{X}_d. \quad (9)$$

This solution locally minimizes the error $(\dot{X}_d - \dot{X})$ in a least-squares sense. If the desired velocity $\dot{X}_d$ is feasible at the current X, eq. (9) will result in zero velocity error. In order to balance error components and to handle nonhomogeneous units, the state X can be pre-weighted or, equivalently, a weighted pseudoinverse can be used. Note that the pseudoinversion (9) gives the command input u as a *feedback* law depending on the current state X.

Consider now the car-like model (5) and define

$$X = \begin{bmatrix} x \\ y \\ \alpha\ell\theta \\ \beta \end{bmatrix}, \quad (10)$$

where $\alpha > 0$ is a weighting real number. By eqs. (5) the pseudoinverse of $G(X)$ takes the form

$$G^{\#}(X) = \frac{1}{1 + \alpha^2 \sin^2(\beta - \theta)} \cdot \begin{bmatrix} \cos\beta & \sin\beta & \alpha\sin(\beta-\theta) & 0 \\ 0 & 0 & 0 & 1 + \alpha^2\sin^2(\beta-\theta) \end{bmatrix}$$

and the feedback law (9) for tracking a desired trajectory $X_d = (x_d, y_d, \alpha\ell\theta_d, \beta_d)$ becomes

$$u_1 = \frac{\dot{x}_d \cos\beta + \dot{y}_d \sin\beta + \alpha^2 \ell \dot{\theta}_d \sin\phi}{1 + \alpha^2 \sin^2\phi} \quad (11)$$

$$u_2 = \dot{\beta}_d. \quad (12)$$

When $\phi = \{0, \pm\pi\}$, the desired $\dot{\theta}_d$ has no effect on u_1.

In order to apply the control law (11–12), we need to specify the desired values for $\dot{x}_d$, $\dot{y}_d$, $\dot{\theta}_d$, and $\dot{\beta}_d$.

4 Local trajectory generation

Let an artificial *force field* be defined in the cartesian workspace so as to pull the robot toward the target position while avoiding obstacles. We suppose that this field is given by the superposition of an attractive component, acting only on the front wheel (i.e., the reference point (x, y)), and of a component aimed at

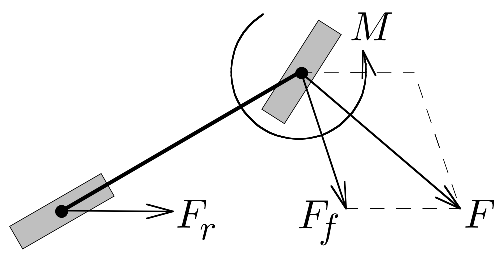

Figure 2 – Forces and torque acting on the robot

keeping the robot away from the obstacles, acting on both wheels. Analytical forms for the fields will be discussed in the next section. Referring to Fig. 2, let $F_f = (F_{f,x}, F_{f,y})$ and $F_r = (F_{r,x}, F_{r,y})$ be the force acting on the front and rear wheel, respectively.

Defining $X_r = (x_r, y_r, \alpha\ell\theta, \beta)$, we have

$$\dot{X}_r = J(X)\dot{X},$$

where, from eqs. (1) and (10), the Jacobian has the form

$$J(X) = \begin{bmatrix} 1 & 0 & \ell\sin\theta & 0 \\ 0 & 1 & -\ell\cos\theta & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By the *virtual work principle*, a force F_r on the rear wheel is equivalent to a force F_f^r on the front wheel and a torque M_θ acting on the vehicle orientation, as given by

$$\begin{bmatrix} F^r_{f,x} \\ F^r_{f,y} \\ M_\theta \\ M_\beta \end{bmatrix} = J^T(X) \begin{bmatrix} F_{r,x} \\ F_{r,y} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} F_{r,x} \\ F_{r,y} \\ \ell(F_{r,x}\sin\theta - F_{r,y}\cos\theta) \\ 0 \end{bmatrix}. \quad (13)$$

Hereafter, we shall indicate with

$$F = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = F_f^r + F_f = F_r + F_f$$

$$M = M_\theta$$

the total force and torque performing work on the x, y, and θ coordinates.

The desired values $\dot{x}_d$, $\dot{y}_d$, and $\dot{\theta}_d$ are selected as the natural motion in quasi-static conditions arising from the above force field, i.e., from eq. (13)

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = k_f F \qquad \dot{\theta}_d = k_f M, \quad (14)$$

with $k_f > 0$. As a result, the control input u_1 is completely defined by eq. (11) and (14).

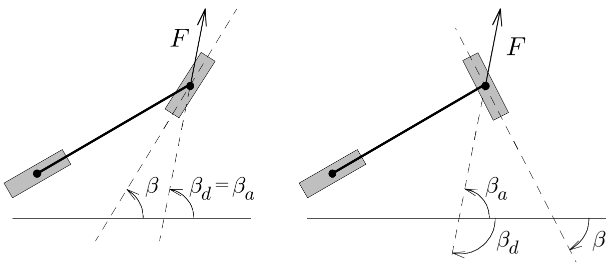

Figure 3 – Choice of β_d: $\beta_a - \beta < \pi/2$ (left) and $\beta_a - \beta > \pi/2$ (right)

We must still define the steering command u_2 in eq. (12). To this end, we will determine a desired angle β_d such that, aligning β to β_d, the robot tends to move in the direction of the field. With reference to Fig. 3, when $F \neq 0$, β_d will be defined as the angle β_a of the direction of the force F, up to a rotation of π rad. When $F = 0$ but $M \neq 0$, one should select β_d so as to allow the rotation of the robot. If both F and M are zero, the robot is at rest. Therefore, we have two possible situations.

Case $F \neq 0$. We have

$$\beta_a = \mathrm{ATAN2}(F_y, F_x).$$

For β_d, we choose the differentiable expression

$$\beta_d = \beta - \arcsin(\sin(\beta - \beta_a)),$$

so that $\beta_d - \beta$ is an acute angle (see Fig. 3). In order to let β track β_d, we impose

$$\frac{d}{dt}(\beta - \beta_d) + k_\beta(\beta - \beta_d) = 0, \;\; k_\beta > 0. \tag{15}$$

The parameter k_β determines the readiness of the steering subsystem, and hence the dynamic range of the steering command u_2. We have

$$\frac{d}{dt}(\beta - \beta_d) = \mathrm{sign}(\cos(\beta - \beta_a))(\dot\beta - \frac{F_x\dot F_y - F_y \dot F_x}{F_x^2 + F_y^2})$$

and

$$\begin{bmatrix} \dot F_x \\ \dot F_y \end{bmatrix} = \nabla F \begin{bmatrix} \dot x \\ \dot y \end{bmatrix}, \;\; \nabla F = \begin{bmatrix} \frac{\partial F_x}{\partial x} & \frac{\partial F_x}{\partial y} \\ \frac{\partial F_y}{\partial x} & \frac{\partial F_y}{\partial y} \end{bmatrix}.$$

Recalling eq. (5), and defining $\mathrm{sign}(0) = 1$, we obtain

$$u_2 = -\frac{k_\beta(\beta - \beta_d)}{\mathrm{sign}(\cos(\beta - \beta_a))} + \frac{u_1}{||F||^2}\left[-F_y \;\; F_x\right] \nabla F \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} \tag{16}$$

as a steering control law. ∎

Case $F = 0$. If this situation arises with both F_f and F_r being zero, the robot will rest and we can park the steering wheel at any desired angle ϕ_g. Otherwise, we choose the desired direction β_a as the angle formed by vector F_f with the x-axis

$$\beta_a = \mathrm{ATAN2}(F_{f,y}, F_{f,x}).$$

Then, we define

$$\beta_d = \begin{cases} \beta - \arcsin(\sin(\beta - \beta_a)), & \text{if } F_f = -F_r \neq 0, \\ \theta + \phi_g, & \text{if } F_f = F_r = 0, \end{cases} \tag{17}$$

and

$$u_2 = -k_\beta(\beta - \beta_d). \tag{18}$$

In the case of rear-wheel driving, if $F = 0$ choice (17) may lead to problems when F_f is almost orthogonal to θ. In this case, ϕ would tend to $\pm\frac{\pi}{2}$ and the robot would stop. We overcome this problem by saturating the term $\arcsin(\sin(\beta - \beta_a))$ to $\pm\pi/4$ in eq. (17). ∎

The following theorem gives a stability result for the proposed control scheme in the absence of obstacles and under mild hypotheses on the force field.

Theorem 1 *In the absence of obstacles ($F_r = 0$), assume a* potential function $U : I\!R^2 \to I\!R$ *is defined with the following properties:*

$$\begin{array}{ll} (1.) & U(x_g, y_g) = 0 \\ (2.) & U(x,y) > 0, \; \forall (x,y) \neq (x_g, y_g) \\ (3.) & \nabla U(x,y) \neq 0, \; \forall (x,y) \neq (x_g, y_g), \end{array}$$

where $G = (x_g, y_g)$ is the goal position for the representative point (x, y). Then, the force field

$$F(x,y) = \begin{cases} -\nabla U(x,y), & \text{if } (x,y) \neq (x_g, y_g), \\ 0, & \text{if } (x,y) = (x_g, y_g), \end{cases}$$

along with the control law (11–12) and the associated definitions, drives the car-like robot (5) to G parking the steering angle ϕ to a desired ϕ_g.

Proof. Define the Lyapunov-like function

$$V(x, y, \beta) = U(x, y) + \frac{1}{2}(\beta - \beta_d)^2.$$

Then, by eqs. (5) and (15)

$$\begin{aligned} \dot V &= \frac{\partial U}{\partial x}\dot x + \frac{\partial U}{\partial y}\dot y + (\beta - \beta_d)(\dot\beta - \dot\beta_d) \\ &= -(F_x \cos\beta + F_y \sin\beta)u_1 - k_\beta(\beta - \beta_d)^2. \end{aligned}$$

Since $F_r = 0$, we have $M = 0$ and $\dot\theta_d = 0$, and hence from eq. (11) and (14)

$$\dot V = -k_f \frac{(F_x\cos\beta + F_y \sin\beta)^2}{1 + \alpha^2 \sin^2\phi} - k_\beta(\beta - \beta_d)^2 \leq 0.$$

If $F \neq 0$ and F is perpendicular to $(\cos\beta, \sin\beta)$, then $\beta - \beta_d = \pm\pi/2$, and hence $\dot{V} < 0$. If $F = 0$, then $\dot{V} = 0$ iff $\beta = \beta_d$, or, by eq. (17), iff $\phi = \phi_g$. By LaSalle's theorem, the result follows. ∎

Theorem 1 applies for example when F is the negated gradient of a potential with paraboloidic profile

$$U(x, y) = \frac{1}{2}\|(x - x_g, y - y_g)\|^2, \tag{19}$$

or with conical profile

$$U(x, y) = \|(x - x_g, y - y_g)\|. \tag{20}$$

In the presence of obstacles, the force F_r acting on the rear wheel may introduce some difficulties. As $F \to 0$, the desired direction β_d becomes undetermined and the tracking defined by eq. (15) is impossible with bounded inputs (see also eq. (16)). In order to overcome this shortcoming, we remove $\dot{\beta}_d$ from eq. (15), setting u_2 as in eq. (18). In the performed simulations, this modification did not appreciably change the robot behavior.

5 Artificial force fields

In order to complete the design of the motion planner, we have to define the way in which the forces acting on the wheels of the car-like robot are generated from the task environment (goal+obstacles).

For the goal attractive potential, we can either use eq. (19) or (20) or, even better, a combination of the two. In fact, an attractive force depending linearly on the error norm (paraboloidic potential) becomes very large at large distances from the goal, dominating over the other forces acting on the robot. On the other hand, such a field smoothly slows down the robot near the goal, avoiding chattering phenomena.

For obstacle avoidance, three kinds of force fields are considered, viz. *repulsive*, *vortex* and *circumventive* fields. For the sake of illustration, we describe the fields generated by a single, simply connected obstacle in the two-dimensional workspace.

Repulsive fields. An hyperboloidic repulsive potential is defined at each point (x, y) as [2]

$$U_{\text{rep}}(x, y) = \begin{cases} \frac{1}{\gamma}\left(\frac{1}{\eta(x,y)} - \frac{1}{\eta_0}\right)^{\gamma}, & \text{if } \eta(x, y) \le \eta_0, \\ 0, & \text{else,} \end{cases}$$

where $\eta = \eta(x, y)$ is the minimum distance of the point from the obstacle, η_0 is the distance of influence of the repulsive field, and $\gamma > 1$ determines how fast the potential decays away from the obstacle.

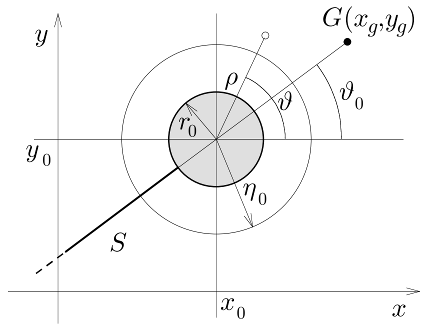

Figure 4 – Obstacle polar coordinate frame

When the obstacles are modeled as circles (e.g., by approximating real shapes acquired by sensors), let (x_0, y_0) and r_0 be, respectively, the coordinates of the center and the radius of the obstacle (see Fig. 4). It is convenient to consider a polar coordinate frame (ρ, ϑ) centered in (x_0, y_0), with $\vartheta = 0$ the direction of the x-axis. The repulsive force becomes then

$$F_{\text{rep}}(\rho, \vartheta) = \begin{cases} \left(\frac{1}{\eta} - \frac{1}{\eta_0}\right)^{\gamma-1} \frac{1}{\eta^2}\, i_\vartheta, & \text{if } \eta \le \eta_0, \\ 0, & \text{else,} \end{cases}$$

with $\eta = \rho - r_0$ and $i_\vartheta = (\cos\vartheta, \sin\vartheta)$. ∎

Vortex fields. Although vortex fields can be defined for obstacles of generic shape [7], we describe here a modification for the case of circular obstacles (or for circles enclosing real obstacles). The idea is to build a field similar to the velocity field described by the steady flow of a liquid in the presence of a cylindrical obstacle. Referring again to Fig. 4, let ϑ_0 be the angle formed by vector $(x_g - x_0,\ y_g - y_0)$ with the x-axis. A possible choice for the field is

$$F_{\text{vor}}(\rho, \vartheta) = \begin{cases} -\operatorname{sign}(\sin(\vartheta - \vartheta_0))\left(\frac{1}{\eta} - \frac{1}{\eta_0}\right)^{\gamma-1} i_\vartheta^\perp, & \text{if } \eta \le \eta_0, \\ 0, & \text{else,} \end{cases}$$

where $i_\vartheta^\perp = (-\sin\vartheta,\ \cos\vartheta)$. ∎

Circumventive fields. Both fields described above have some drawbacks. The first repels the robot in a direction which is orthogonal to the border of the obstacle, often keeping it too far from the obstacle itself, while the second may lead the robot to graze the obstacle. Hence, it is advisable to have a field that is repulsive close to the obstacle and vortical at larger distances.

Consider the unit vectors respectively associated to F_{rep} and F_{vor}

$$\begin{aligned} E_\vartheta &= i_\vartheta \\ E_\vartheta^\perp &= -\operatorname{sign}(\sin(\vartheta - \vartheta_0))\, i_\vartheta^\perp, \end{aligned}$$

and the smooth weighting function

$$\sigma(\eta) = \left(1 + \frac{\eta}{\eta_\sigma}\right) e^{-\frac{\eta}{\eta_\sigma}},$$

which is monotonically decreasing from 1 to 0. The rate of decay of $\sigma(\eta)$ depends on the parameter η_σ. We call *circumventive field* the convex combination of a repulsive and a vortical component

$$F_{\text{cir}}(\rho,\vartheta)=\begin{cases}(\sigma(\eta)E_\vartheta+(1-\sigma(\eta))E_\vartheta^{\perp})(\frac{1}{\eta}-\frac{1}{\eta_0})^{\gamma-1} & \text{if } \eta\leq\eta_0\\ 0, & \text{else.}\end{cases} \quad \blacksquare$$

In the presence of multiple obstacles, the total field is obtained by adding the fields independently defined for each obstacle.

6 Simulation results

We simulated the proposed planner for a car-like robot moving among circular obstacles in a two-dimensional workspace. The integration of the kinematic and planning equations was performed using Simulink and the fifth-order Runge-Kutta method.

In Fig. 5 we report the results obtained with the three fields proposed in Sect. 5 for a first simulation. All of them are successful, even if the path generated by the repulsive field (dotted line) is more erratic. We have set $\gamma = 4$, $\alpha = 1$, $k_\beta = 10$, $k_f = 1$, $\eta_\sigma = \eta_0/10$, while $\dot{\beta}_d$ was removed from eq. (15) by setting u_2 as in eq. (18). The region of influence around each obstacle is represented with a circle. The attractive potential is conical outside a circle of unit radius centered at the goal, and paraboloidic inside.

While in the circumventive and the vortex field case the robot reached the goal in about 10 s, with a repulsive field this time resulted in about 20 s. However, in comparing these values it is fair to consider

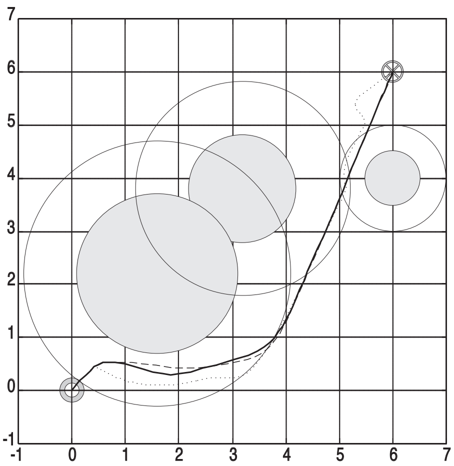

Figure 5 – First simulation: results with repulsive (dotted), vortex (dashed), and circumventive (solid) fields

Field Type	max $\|u_r\|$	max $\|u_\phi\|$
Repulsive	1.0145	5.9419
Vortex	28.8776	15.75
Circumventive	5.3505	4.3648

Table 1 – Comparison of input dynamic ranges in the simulation of Fig. 5

the dynamic range of the inputs, which are reported in Table 1. Since the geometric paths are invariant to time-scaling, the trade-off between traveling time and input effort can be regulated by premultiplying the inputs u_r (or u_f) and u_ϕ by the same factor k_u.

In the second simulation (Fig. 6), the rightmost obstacle of Fig. 5 was slightly shifted to the left. The repulsive field method fails in this case, since the robot meets a local minimum of the overall force field. Instead, the circumventive field successfully drives the robot to the goal, thanks to its vortical component.

In the simulation shown in Fig. 7, the rightmost obstacle was shifted further to the left. The vortex field method fails as the robot touches the obstacle

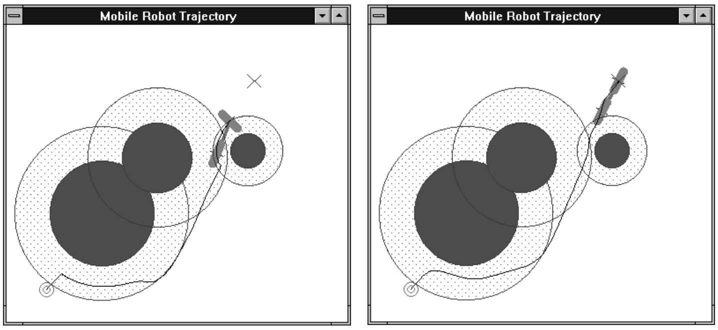

Figure 6 – Second simulation: results with repulsive (left) and circumventive (right) field method

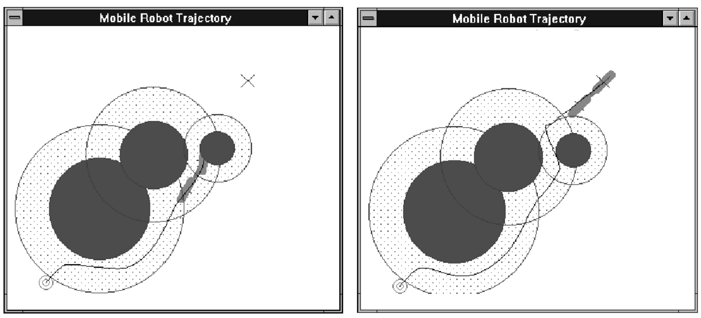

Figure 7 – Third simulation: results with vortex (left) and circumventive (right) field method

($\|F_{\rm vor}\| \to \infty$). This is avoided when the circumventive field is used, thanks to its repulsive component.

7 Conclusions

We have proposed an integrated approach for locally planning the motion of a car-like nonholonomic robot among obstacles. The feedback component of the scheme is able to stabilize the robot to a given cartesian position, in the absence of obstacles. The obstacle avoidance action is obtained through a combined use of repulsive and vortex fields. An additional command is used to align the steering wheel with the force field.

Simulation results show that the proposed planner performs satisfactorily in situations where high maneuverability is not essential. Therefore, the planner is more suited for navigation through a semi-cluttered environment rather than for parking purposes with limited clearance.

Future work includes the implementation of this method on a laboratory prototype and its extension to other vehicles, e.g. the car with N trailers.

References

[1] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic, Boston, 1991.

[2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. of Robotics Research*, vol. 5, no. 1, pp. 90–99, 1986.

[3] R. Volpe and P. Khosla, "Artificial potential with elliptical isopotential contours for obstacle avoidance," *28th IEEE Conf. on Decision and Control*, Los Angeles, CA, pp. 180–185, 1987.

[4] A. A. Masoud, S. A. Masoud, and M. M. Bayoumi, "Robot navigation using a pressure generated mechanical stress field—The biharmonic potential approach," *1994 IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, pp. 124–129, 1994.

[5] D. E. Koditschek, "Exact robot navigation by means of potential functions: Some topological considerations," *1987 IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, pp. 1–6, 1987.

[6] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential fields techniques for robot path planning," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, pp. 224–241, 1992.

[7] C. De Medio and G. Oriolo, "Robot obstacle avoidance using vortex fields," in *Advances in Robot Kinematics*, S. Stifter and J. Lenarčič (Eds.), Springer-Verlag, Wien, pp. 227–235, 1991.

[8] J.-P. Laumond, "Nonholonomic motion planning versus controllability via the multibody car system example," Rep. STAN-CS-90-1345, Stanford University, CA, Dec. 1990.

[9] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. on Automatic Control*, vol. 38, pp. 700–716, 1993.

[10] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, R. W. Brockett, R. S. Millmann, and H. J. Sussmann (Eds.), Birkhäuser, Boston, pp. 181–191, 1983.

[11] C. Canudas de Wit and O. J. Sørdalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Trans. on Automatic Control*, vol. 37, pp. 1791–1797, 1992.

[12] C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *Int. J. of Robotics Research*, vol. 12, pp. 55–64, 1993.

[13] J.-P. Laumond, P. E. Jacobs, M. Taïx, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 10, pp. 577–593, 1994.

[14] J.-C. Latombe, "A fast path planner for a car-like indoor mobile robot," *9th Nat. Conf. on Artificial Intelligence*, Anaheim, CA, pp. 659–665, 1991.

[15] A. Bicchi, G. Casalino, and C. Santilli, "Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles," *1995 IEEE Int. Conf. on Robotics and Automation*, Nagoya, J, pp. 1349–1354, 1995.

[16] J. Barraquand and J.-C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, pp. 2328–2335, 1991.

[17] A. De Luca and G. Oriolo, "Local incremental planning for nonholonomic mobile robots," *1994 IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, pp. 104–110, 1994.

[18] D. Feng and B. H. Krogh, "Satisficing feedback strategies for local navigation of autonomous mobile robots," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, pp. 1383–1395, 1990.