Hierachical Local Incremental Planners for Mobile Robots 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

September 18, 1997

Abstract

We present a simple local approach for planning the motion of nonholonomic robots navigating among obstacles, suitable for sensor-based implementation. Existing methods lead to open-loop solutions which are either obtained in two stages, approximating a previously built holonomic path, or computationally intensive, being based on configuration space discretization. Our nonholonomic planner, by explicitly taking into account the nonholonomic nature of the robot kinematics, employs a direct projection strategy to modify on-line the output of a generic local holonomic planner, and generates velocity control inputs that realize the desired motion in a least-squares sense. As a result, a feedback scheme is obtained which can use only local sensor information. The proposed approach is applied to unicycle and car-like kinematics, and in order to avoid motion stops away from the desired goal, various force fields are considered and compared by simulation.

1 Introduction

Robots whose motion is subject to non-integrable constraints involving time derivatives of the configuration variables belong to the class of *nonholonomic* mechanical systems [1]. Typical examples are wheeled mobile robots moving on the plane under perfect rolling constraints. The effect of these constraints is to limit the local mobility of the robotic system, though not restricting in the large the accessibility of the whole configuration space.

For nonholonomic robots, the design of feasible trajectories joining arbitrary initial and final configurations is not straightforward, and can be tackled as an intrinsic nonlinear control problem. In this respect, *open-loop* schemes and *feedback* control are possible solutions. Based on a differential geometric analysis, open-loop commands that exactly drive to the goal have been derived in [4, 5, 6] for the class of mobile robots that can be put in the so-called chained form [4], e.g., for a car with N trailers. Feedback schemes are indeed more robust but subject to a basic limitation: nonholonomic systems cannot be stabilized to a given configuration

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

by means of a continuously differentiable feedback law, as follows from a general result due to Brockett [7]. This motivated the more complex design of *discontinuous* [8], smooth *time-varying* [9], or hybrid [10] feedback control laws. A common characteristic of these open- or closed-loop methods is that a sequence of maneuvers is required in order to complete a point-to-point motion, like in parking a car or docking a truck with trailers. The *presence of obstacles* in the operational space is not taken into account at this stage.

On the other hand, the literature on motion planning with obstacle avoidance is quite rich [11] for conventional (holonomic) robots, that can move in any direction of the free configuration space. 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. Techniques from real algebraic geometry are used to guarantee completeness, that is finding a solution whenever one exists. The resulting algorithms are powerful but very difficult to implement for high-dimensional configuration spaces. Moreover, shape and location of all obstacles must be known a priori.

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 [12]. These fields may be defined in the configuration space or, more conveniently, in the cartesian space. 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 [13], biharmonic functions [23], navigation functions [14] and numerical potential fields [19]. Another method that can avoid motion stops using only *local* information is the *vortex field* method [16]: Repulsive actions are replaced by vortical velocity flows so that the robot is forced to turn around the 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 [14, 15, 16, 17]. In [3, 18], 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 with the nonholonomic constraints. Shortest paths of bounded curvature were obtained in [24], while a discretization of the configuration space is essential in [19], where graph search based on Dijkstra algorithm is performed. A common requirement of this class of methods is the a priori knowledge of the environment, so that the solutions are inherently off-line. Heuristic rules have been used in [28] for a car-like robot so as to guarantee collision avoidance.

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 this paper we propose an on-line local navigation technique for nonholonomic vehicles moving in the presence of obstacles. The idea is to employ a feasible projection strategy to modify on-line the output of a holonomic incremental planner.

The resulting feedback scheme, depicted in Fig. 1, uses only local information, limited

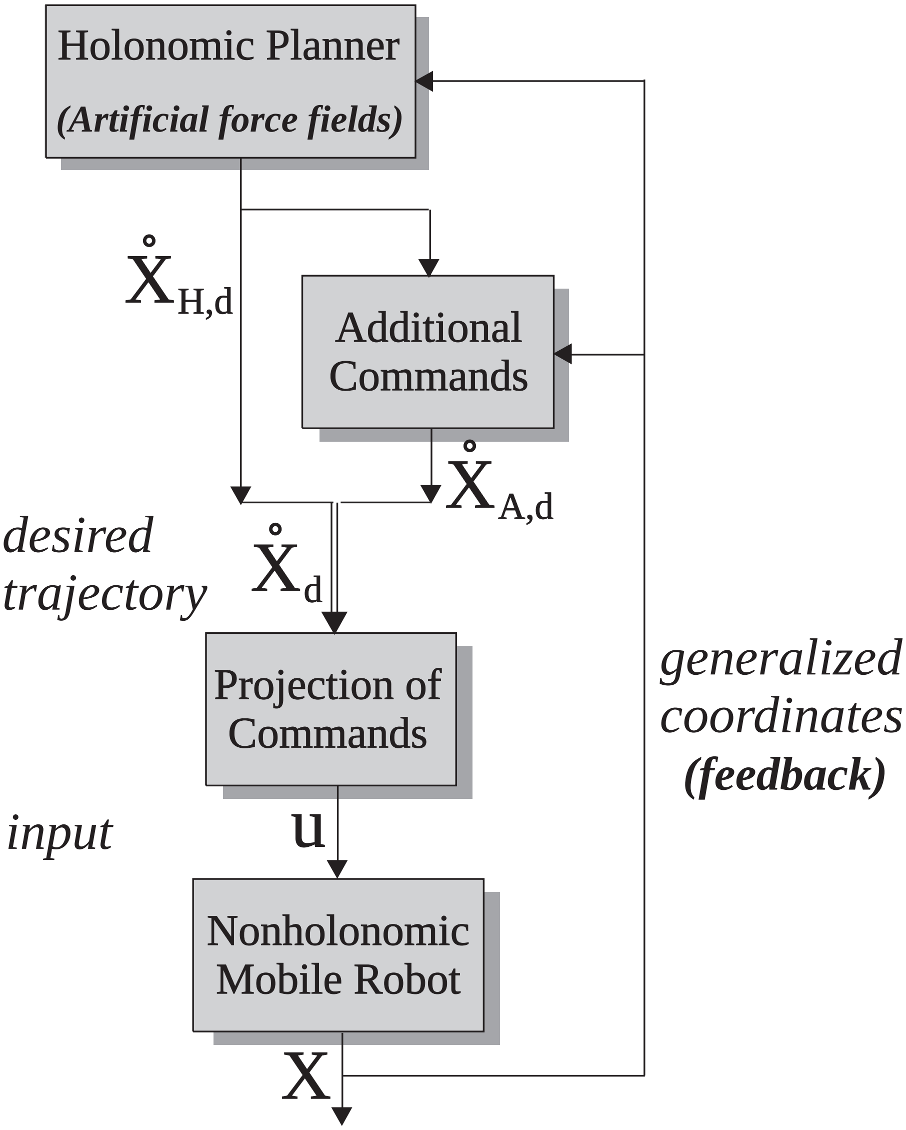

Figure 1: Hierarchical approach to incremental nonholonomic motion planning.

e.g. to the range of distance sensors. The crucial feature of this strategy is that it should not create additional blocking points in the configuration space of the nonholonomic vehicle. We highlight the general approach and then apply it to two common mobile robot kinematic configurations: Unicycle and car-like. For the latter, 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. Simulation results are reported which compare different kinds of artificial fields used in the holonomic planner.

2 Incremental motion planning for nonholonomic robots

We consider the planar motion of a wheeled mobile robot. Denoting by $X \in \mathbb{R}^n$ the vector of generalized coordinates, assume that the system motion is subject to a set of $p < n$ nonholonomic constraints in the form

$$A(X)\,\dot{X} = 0, \tag{1}$$

that arise in connection with the 'rolling without slipping' condition on the wheels. Note that the presence of rolling wheels does not necessarily imply that motion on the plane is subject to nonholonomic constraints. Wheels with side rollers or other complex mechanisms may be used to guarantee omnidirectionality.

Since constraint (1), involving the time derivatives of the generalized coordinates, is not integrable, the dimension of the configuration space cannot be reduced. However, all feasible velocities $\dot{X}$ should satisfy the following equation

$$\dot{X} = G(X)\,u, \qquad u \in \mathbb{R}^{n-p}, \tag{2}$$

where the $n-p$ independent columns of $G(X)$ are a basis for the null space of $A(X)$. Equation (2) is the *kinematic model* of the mobile robot. Depending on the choice of the null space basis, the components u_i will have a different meaning. Indeed, there is one choice which leads to a convenient physical interpretation of u in terms of the available command inputs. Notice that eq. (2) represents an *underactuated* mechanical system, with less inputs than generalized coordinates.

We assume in the following that control inputs are at the velocity level. In practice, this is not restrictive for real mobile robot control. Although the modeling can easily be extended to second order and to include the system dynamics [20], eq. (2) already displays the main difficulty due to nonholonomy. In fact, system (2) is essentially *underactuated*, having strictly less independent inputs u than motion planning variables X.

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

$$u = G^{\#}(X)\dot{X}_d = [G^{\mathrm{T}}(X)G(X)]^{-1}G^{\mathrm{T}}(X)\dot{X}_d. \tag{3}$$

This solution locally minimizes the error $(\dot{X}_d - \dot{X})$ in a least-squares sense. If the desired velocity $\dot{X}_d$ satisfies the nonholonomic constraint (1) at the current X, eq. (3) will result in zero velocity error. Note that the pseudoinversion (3) gives the command input u as a *feedback* law depending on the current state X. 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.

In eq. (3), $\dot{X}_d$ can be chosen as the output of an incremental holonomic planner. If artificial potentials are used to drive the robot, then

$$\dot{X}_d = -\nabla_X U(X) = -\nabla_X (U_a(X) + U_r(X)), \tag{4}$$

with an attractive potential U_a to the goal X_g and a repulsive potential U_r.

However, working with potentials defined in the *whole* configuration space X is computationally inefficient [11]. In view of the planar nature of the motion problem, one can partition X as (X_p, X_θ), with the positional part $X_p = (x, y) \in \mathbb{R}^2$ and the angular part $X_\theta \in \mathbb{R}^{n-2}$. For example, in a car with N trailers, X_p are the cartesian coordinates of one representative point of the robot (typically, the position of the last trailer [21]), while X_θ contains the orientation and the steer angle of the car as well as the relative orientation of each trailer. Potential fields can then be set up for X_p, i.e., directly in the operational space where obstacles live, by defining several cartesian points $P_i = (x_i, y_i)$ on the (multibody) mobile robot. Each of these *control points* will be subject to a field $U_{a,i} + U_{rep}$, being $U_{a,i}$ the attractive field associated to the goal for P_i. The desired motion becomes

$$\dot{X}_d = -\sum_i J_i^T(X) \nabla_{X_p} \left(U_{a,i}(P_i) + U_r(P_i) \right), \tag{5}$$

where $J_i(X)$ is the Jacobian of the kinematic mapping $P_i = f_i(X)$ of the i-th control point.

Substitution of eqs. (4) or (5) in (3) yields a nonholonomic motion planner. Its performance obviously depends on the characteristics of the holonomic planner, but also on the interaction between the latter and the projection scheme.

In order to allow for more flexibility in the design, we split X_d in two components $X_{H,d}$ and $X_{A,d}$, keeping the first in eq. (5), and specifying the desired motion of the remaining generalized coordinates $X_{A,d}$ in a more general form

$$\dot{X}_{A,d} = \Phi(X, \dot{X}_{H,d}), \tag{6}$$

where an explicit hierarchic dependence of $X_{A,d}$ on $X_{H,d}$ has been introduced (Fig. 1). As an example, in Sect. 4 we will set $X_A = X_p = (x, y)$, $X_H = X_\theta = \theta$ for the unicycle, and in Sect 5 $X_A = (x, y, \theta)$, $X_H = X_\theta = \beta$ for the car-like robot. The design of a suitable Φ is strictly related to the kinematic structure of the vehicle and is critical for the success of the method. In particular, one should guarantee that *no additional blocking points* are generated in the configuration space X of the nonholonomic vehicle, beside those possibly existing for X_p.

We note that, as long as the overall feedback law (3) is continuously differentiable, the scheme will not be able to stabilize the mobile robot at a given configuration X_g (or, equivalently, not all control points P_i will reach their final position). In fact, this would violate the theoretical result of Brockett [7], as applied to nonholonomic systems. However, since our objective is the definition of a navigation method among obstacles, we are not interested in the specific configuration reached at the end of the motion, provided that the positional coordinates of the robot reach their destination $X_{p,g}$.

Finally, we point out a possible shortcoming of the method. Since the commands of the holonomic planner are realized only in a least-squares sense, there is no complete guarantee that obstacles will be avoided during motion. Therefore, use of motion safety margins related to the maximum velocity error is advisable.

3 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 robot are generated from the task environment (goal+obstacles). One basic component of the proposed planner is its holonomic part which can be essentially based on any local approach. According to eq. (5), we will work with artificial potentials

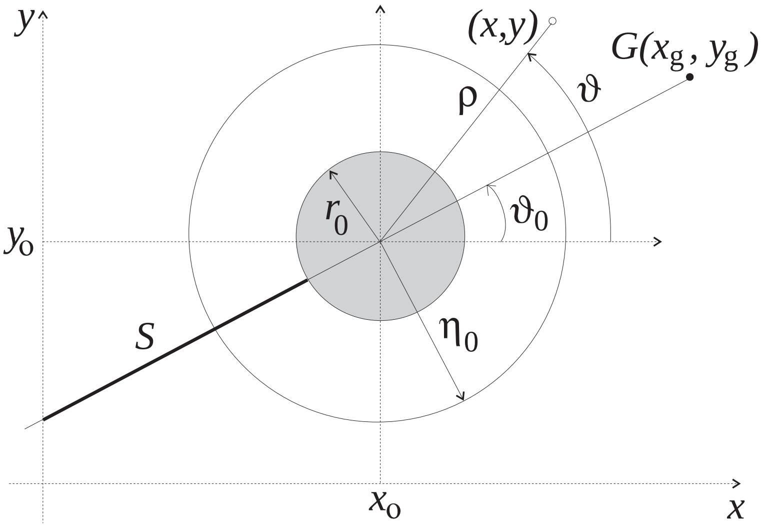

Figure 2: Obstacle polar coordinate frame.

defined in the cartesian space $\mathbb{R}^2$, where $X_p = (x, y)$. More specifically, we follow [12] and superpose one field $U_a(x, y)$, which attracts the robot towards the goal position (x_g, y_g), with a repulsive field $U_{\rm r}(x, y)$ for each obstacle, devoted to prevent collisions.

For obstacle avoidance, three kinds of fields are considered as repulsive forces $\nabla U_{\rm r}$, viz. *stricly 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.

3.1 Attractive Fields

For the goal attractive potential, we can either use the paraboloidic profile

$$U_a(x, y) = \frac{k_a}{2} ||(x - x_g, y - y_g)||^2, \tag{7}$$

or the conical profile

$$U_a(x, y) = k_a ||(x - x_g, y - y_g)||. \tag{8}$$

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. ■

3.2 Stricly Repulsive Fields

An hyperboloidic stricly repulsive potential is defined at each point (x, y) as [12]

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

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.

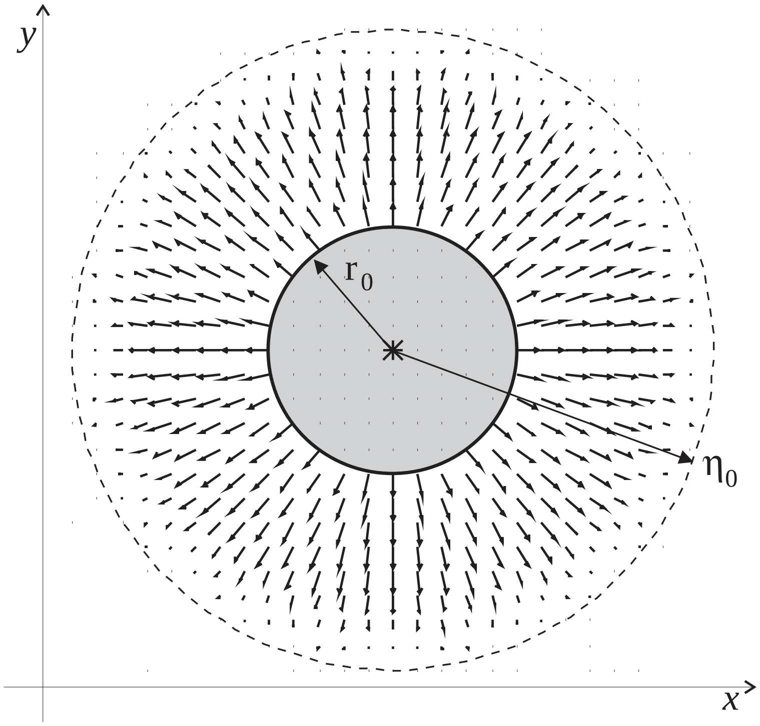

Figure 3: Strictly repulsive field for a circular obstacle.

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, (x_g, y_g) the coordinates of the goal position, and ϑ_0 the angle formed by vector $(x_g - x_0,\ y_g - y_0)$ with the x-axis (see Fig. 2). 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

$$F_{\text{rep}}(\rho, \vartheta) = \begin{cases} (\frac{1}{\eta} - \frac{1}{\eta_0})^{\gamma-1} \frac{1}{\eta^2}\, i_\vartheta, & \text{if } \eta \leq \eta_0, \\ 0, & \text{else}, \end{cases} \tag{9}$$

with $\eta = \rho - r_0$ and $i_\vartheta = (\cos\vartheta, \sin\vartheta)$, and is depicted in Fig. 3.

It is easy to see that, in the case of m circular obstacles, the *total potential field* has always at least m saddle points but no isolated local minima other than the goal [14]. In the presence of obstacles of arbitrary shape, and in particular with boundary segments of zero curvature, local minima will appear with possibly large basins of attraction. ■

3.3 Vortex Fields

As a less conventional approach, we will also use the *vortex field* method [16]. The basic idea is to replace the antigradient of the given strictly repulsive field with a flow rotating around the obstacle. Analytically, this amounts to moving along the direction defined by

$$F_{\text{vor}}(X_p) = \pm \begin{bmatrix} \frac{\partial U_{\text{rep}}(X_p)}{\partial y} \\ -\frac{\partial U_{\text{rep}}(X_p)}{\partial x} \end{bmatrix}, \tag{10}$$

with the + sign corresponding to counterclockwise (CCW) rotation of the vortex (see Fig. 4). Note that the norm of the vortex field F_{vor} is the same of ∇U_{rep}, and it goes to zero at distance η_0 with continuity. The attractive field is left unchanged.

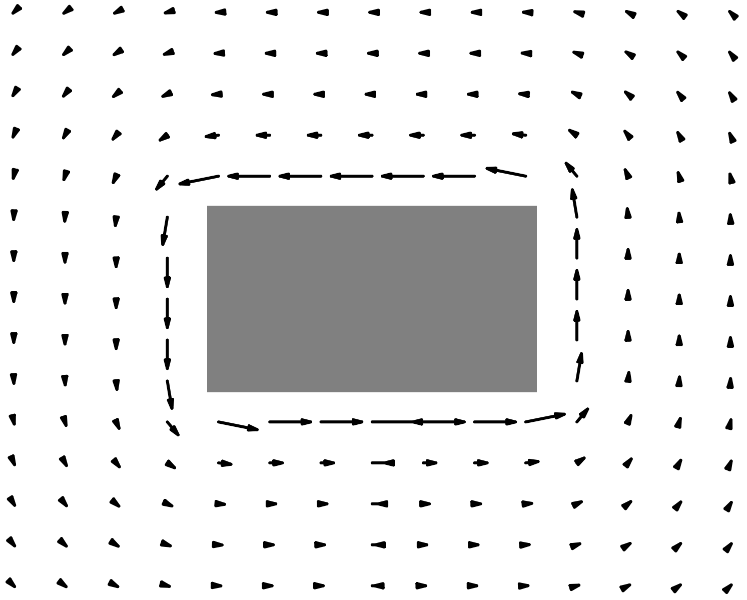

Figure 4: Vortex field for a rectangular obstacle.

In the case of convex obstacles with domains of influence that do not overlap, it is possible to guarantee the *completeness* of the vortex field method. The choice of CW or CCW rotation for an obstacle should be made so that the relative angle between the attractive direction $-\nabla U_a$ and $F_{\rm vor}$ when entering its influence range is $\leq 90°$. The method has to be complemented with a suitable *relaxation* procedure which discards the vortex —and thus the influence of the 'bypassed' obstacle— when the relative angle $|\vartheta - \vartheta_0|$ in Fig. **??** is smaller that 90° [16]. Other higher-level strategies can be devised to generalize the method to non-convex obstacles or uncertain environments [22].

In alternative to the above relaxation procedure, 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. A possible choice for the field is

$$F_{\rm vor}(\rho,\vartheta)=\begin{cases} -\,\mathrm{sign}(\sin(\vartheta-\vartheta_0))\,(\frac{1}{\eta}-\frac{1}{\eta_0})^{\gamma-1} i_\vartheta^\perp, & \text{if } \eta \leq \eta_0, \\ 0, & \text{else,} \end{cases} \tag{11}$$

where $i_\vartheta^\perp = (-\sin\vartheta,\ \cos\vartheta)$. The resulting force field is shown in Fig. 5.

We stress that the heuristics for vortex rotation and relaxation are locally defined and can be implemented on line in a sensor-based planner. Finally, note that the vortex field method does not enforce an explicit repulsive action, so that a more conservative choice of the gain in (7) will be necessary to avoid the robot approaching too closely the obstacles. ■

3.4 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.

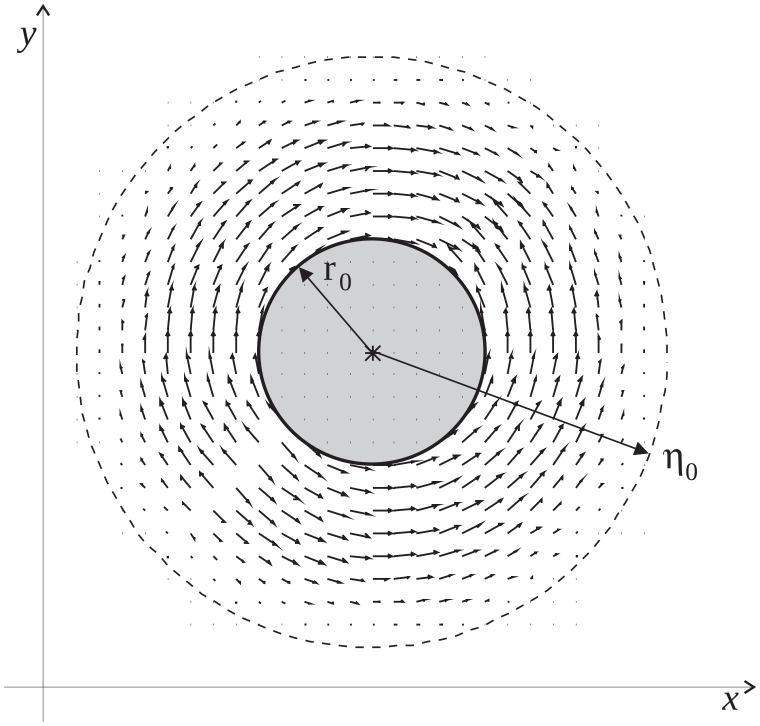

Figure 5: Vortex field flowing around a circular obstacle.

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

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

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

$$F_{\text{cir}}(\rho, \vartheta) = \begin{cases} (\sigma(\eta)E_\vartheta + (1-\sigma(\eta))E_\vartheta^\perp)(\frac{1}{\eta} - \frac{1}{\eta_0})^{\gamma-1}, & \text{if } \eta \leq \eta_0 \\ 0, & \text{else.} \end{cases} \quad ■ \tag{12}$$

Fig. 6 reports an example for a circular obstacle, while Fig. 7 depicts its superposition with the attractive field (8), which gives the total force field used in the holonomic planner.

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

4 Application to the unicycle

In this section, we apply the proposed approach to the kinematics of a unicycle (Fig. 8), where $X = (x, y, \theta)$ is the configuration vector. In this case, there is only one nonholonomic

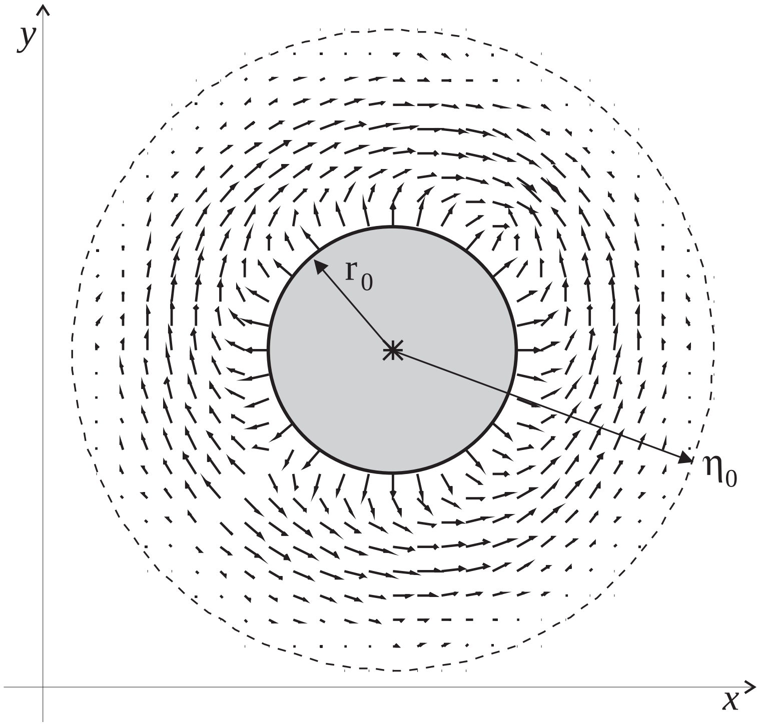

Figure 6: Circumventive field for a circular obstacle.

rolling constraint of the form (1):

$$\begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0. \tag{13}$$

The kinematic model is determined as

$$\begin{cases} \dot{x} & = \cos\theta \; u_1 \\ \dot{y} & = \sin\theta \; u_1 \\ \dot{\theta} & = u_2, \end{cases} \tag{14}$$

where u_1 is the *driving* velocity and u_2 is the *steering* velocity. This model applies to a large class of mobile robots, including the Nomad 200$^{\text{TM}}$ available in our Laboratory. Nomad has a circular base of radius $R \approx 28$ cm with three wheels that translate together, controlled by one motor, and rotate together, controlled by a second motor.

According to eq. (3), the control input is chosen as

$$u = G^{\#}(X)\dot{X}_d = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix}. \tag{15}$$

This expression has a direct geometric interpretation. The driving velocity u_1 is the *orthogonal projection* of the (possibly infeasible) desired cartesian velocity along the robot main axis, while the steering velocity u_2 exactly realizes the desired rotation. Note that in the present case $G^{\#} = G^T$; also, unitary translational and rotational velocities are given the same weight.

The desired velocity $\dot{X}_d$ in eq. (15) comes from an incremental holonomic planner. In the following, we assume that the orientation of the mobile robot has no relevance for obstacle

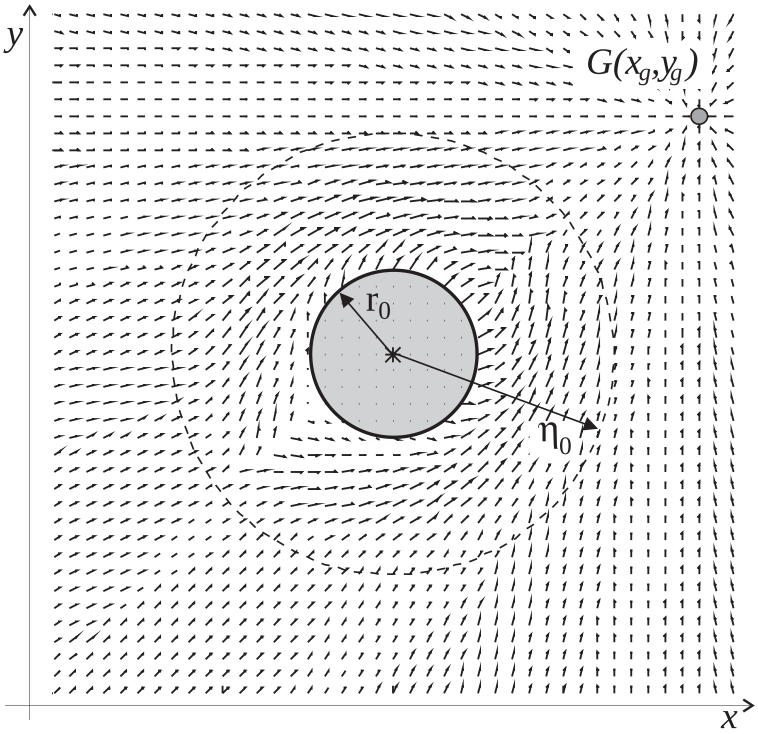

Figure 7: Total field (conic attractive + circumventive).

collision, as in the case of Nomad. As a consequence, the obstacles can be grown by R and holonomic planning will be made only for the central point of the robot[1]. The positional part $\dot{X}_{p,d} = (\dot{x}_d, \dot{y}_d)$ is then obtained from eq. (5) as

$$\dot{X}_{p,d} = -\nabla_{X_p} U_a(X_p) + F_r(X_p) \tag{16}$$

where F_r is one of the fields (9)–(12) described in the previous section.

To complete the planning method we assign the rotational part of $\dot{X}_{\theta,d} = \dot{\theta}_d$ by specifying the form of Φ in eq. (6). For the unicycle it is convenient to use

$$\dot{\theta}_d = \text{atan2}\{\dot{x}_d, \dot{y}_d\} - \theta. \tag{17}$$

The rationale for this choice is simple. Since the unicycle can instantaneously execute linear motions only along its main axis, we force the vehicle to *align* itself with the field flow. Although the robot has circular symmetry, it is implicit in eq. (17) that the *forward* direction (i.e. the one characterized by θ) will be aligned. By defining atan2$\{0, 0\} = \theta$, the above function remains continuous along any approaching direction to the goal. The resulting command will be

$$\begin{aligned} u_1 &= k_p(\dot{x}_d \cos\theta + \dot{y}_d \sin\theta) \\ u_2 &= k_\theta(\text{atan2}\{\dot{x}_d, \dot{y}_d\} - \theta), \end{aligned} \tag{18}$$

where $\dot{x}_d$ and $\dot{y}_d$ are given by eq. (16). Gains k_p and k_θ are introduced to allow for additional freedom in weighting the two input commands. This is equivalent to use a weighted pseudoinverse in eq. (15).

[1] We are using a single control point. In the case of non-circular robots, it will be necessary to specify multiple control points on the robot body for successful motion planning.

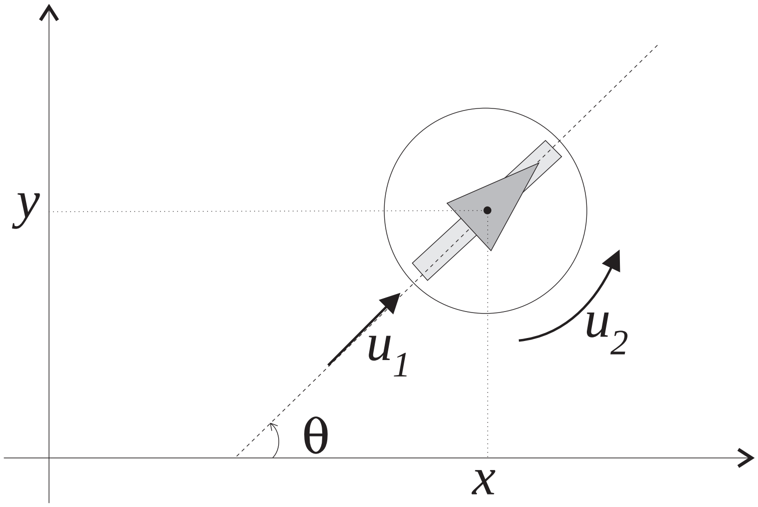

Figure 8: Kinematic structure of a unicycle.

Theorem 1 *The motion of the unicycle under the control law (18) and attractive field (7) converges to the desired position goal $X_{p,g}$ in the absence of obstacles.*

Proof. . Let

$$V = \frac{1}{2}\|X_{p,g} - X_p\|^2 \triangleq \frac{1}{2}\|E_p\|^2 \geq 0. \tag{19}$$

Since $\dot{X}_{p,d} = -k_a E_p$ from eq. (16), the time derivative of V along the closed-loop trajectories is

$$\begin{aligned} \dot{V} &= -\dot{X}_p^T E_p = -u_1 \begin{pmatrix} \cos\theta & \sin\theta \end{pmatrix} E_p \\ &= -k_p k_a E_p^T \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \end{pmatrix} E_p \leq 0. \end{aligned}$$

We have that $\dot{V} = 0$ iff the row vector $\begin{pmatrix} \cos\theta & \sin\theta \end{pmatrix}$ is orthogonal to E_p. Whenever $\dot{V} = 0$ and $E_p \neq 0$, the system dynamics becomes $\dot{x} = \dot{y} = 0$, $\dot{\theta} = \pm\pi/2$. Hence, the largest invariant set such that $\dot{V} = 0$ is $X_p = X_{p,g}$. By LaSalle's theorem, the result follows. ■

The above argument shows that the control law does not introduce further local minima in the configuration space of the nonholonomic robot, beside those possibly due to the holonomic field in $\mathbb{R}^2$. Note that it is not possible to prove convergence for position *and* orientation, in view of the continuous differentiability of the chosen control law in the absence of obstacles.

5 Application to car-like robots

We refer to the car-like robot model depicted in Fig. 9. 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 $\beta = \theta + \phi$ the heading angle of the front wheel w.r.t. the x-axis.

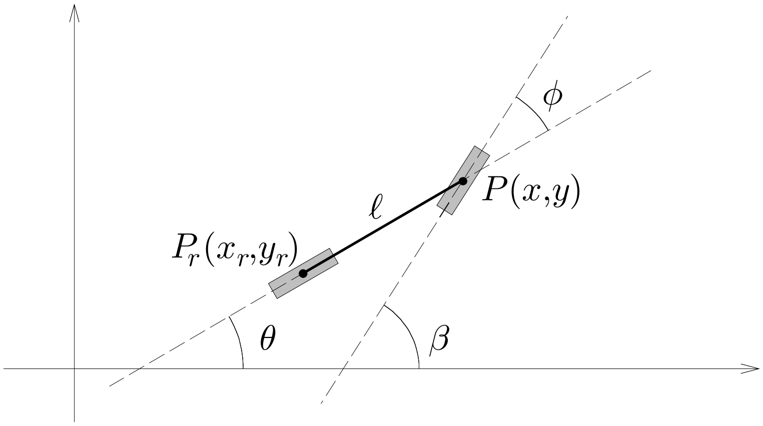

Figure 9: Car-like robot model.

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

$$\begin{aligned} x &= x_r + \ell \cos\theta \\ y &= y_r + \ell \sin\theta. \end{aligned} \tag{20}$$

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

$$\begin{aligned} \dot{x}_r &= u_r \cos\theta \\ \dot{y}_r &= u_r \sin\theta \\ \ell\dot{\theta} &= u_r \tan\phi \\ \dot{\phi} &= u_\phi. \end{aligned} \tag{21}$$

Defining two new system inputs u_1 and u_2 through

$$u_r = u_1 \cos\phi \tag{22}$$

$$u_\phi = u_2 - \frac{u_1 \sin\phi}{\ell} \tag{23}$$

and using eqs. (20), model (21) can be transformed in

$$\begin{aligned} \dot{x} &= u_1 \cos\beta \\ \dot{y} &= u_1 \sin\beta \\ \ell\dot{\theta} &= u_1 \sin(\beta - \theta) \\ \dot{\beta} &= u_2. \end{aligned} \tag{24}$$

Front-wheel driving. The kinematic equations are

$$\begin{aligned} \dot{x} &= u_f \cos(\theta + \phi) \\ \dot{y} &= u_f \sin(\theta + \phi) \\ \ell\dot{\theta} &= u_f \sin\phi \\ \dot{\phi} &= u_\phi. \end{aligned} \tag{25}$$

Letting

$$u_f = u_1 \tag{26}$$

and u_ϕ as in eq. (23), model (25) takes the form (24).

Note that, by dropping the third equation, eqs. (24) formally represent the kinematic model of a unicycle. We develop motion control algorithms directly for model (24), independently of which is the driving wheel. Define

$$X = \begin{bmatrix} x \\ y \\ \alpha\ell\theta \\ \beta \end{bmatrix}, \tag{27}$$

where $\alpha > 0$ is a weighting real number. By eqs. (24) 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 (3) for tracking a desired trajectory $X_d = (x_d, y_d, \alpha\ell\theta_d, \beta_d)$ becomes

$$u_1 = \frac{\dot{x}_d\cos\beta + \dot{y}_d\sin\beta + \alpha^2\ell\dot{\theta}_d\sin\phi}{1+\alpha^2\sin^2\phi} \tag{28}$$

$$u_2 = \dot{\beta}_d. \tag{29}$$

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

In order to apply the control law (28–29), we need to specify the desired values for $\dot{x}_d$, $\dot{y}_d$, $\dot{\theta}_d$, and $\dot{\beta}_d$.

5.1 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, according to the structures detailed in Sect. 3. 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 keeping the robot away from the obstacles, acting on both wheels. Referring to Fig. 10, 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. (20) and (27), the Jacobian has the form

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

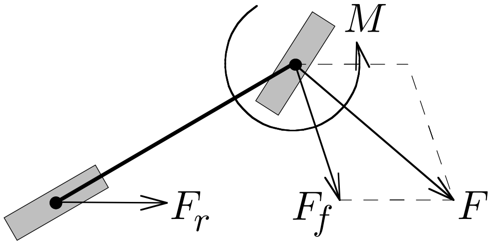

Figure 10: Forces and torque acting on the robot.

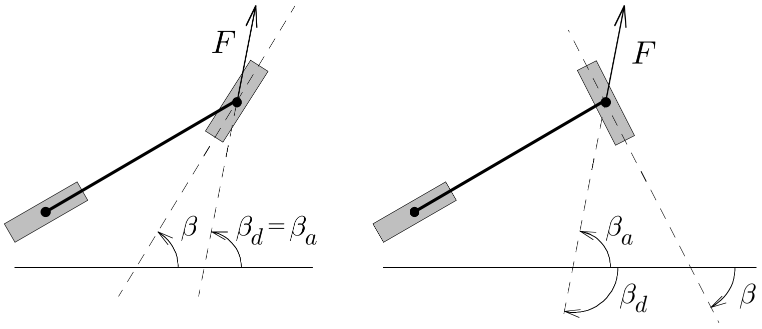

Figure 11: Choice of β_d: $\beta_a - \beta < \pi/2$ (left) and $\beta_a - \beta > \pi/2$ (right).

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

$$\begin{bmatrix} F^r_{f,x} \\ F^r_{f,y} \\ M_\theta \\ M_\beta \end{bmatrix} = J^T(X) \begin{bmatrix} F_{r,x} \\ F_{r,y} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} F_{r,x} \\ F_{r,y} \\ \ell(F_{r,x}\sin\theta - F_{r,y}\cos\theta) \\ 0 \end{bmatrix}. \tag{30}$$

Hereafter, we shall indicate with

$$\begin{aligned} F &= \begin{bmatrix} F_x \\ F_y \end{bmatrix} = F_f^r + F_f = F_r + F_f \\ M &= M_\theta \end{aligned}$$

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. (30)

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = k_f F \qquad \dot{\theta}_d = k_f M, \tag{31}$$

with $k_f > 0$. As a result, the control input u_1 is completely defined by eq. (28) and (31).

We must still define the steering command u_2 in eq. (29). 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. 11, 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. 11). In order to let β track β_d, we impose

$$\frac{d}{dt}(\beta - \beta_d) + k_\beta(\beta - \beta_d) = 0, \quad k_\beta > 0. \tag{32}$$

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

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

Recalling eq. (24), and defining $\mathrm{sign}(0) = 1$, we obtain

$$u_2 = -\frac{k_\beta(\beta - \beta_d)}{\mathrm{sign}(\cos(\beta - \beta_a))} + \frac{u_1}{||F||^2}\begin{bmatrix} -F_y & F_x \end{bmatrix} \nabla F \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} \tag{33}$$

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

$$\beta_d = \begin{cases} \beta - \arcsin(\sin(\beta - \beta_a)), & \text{if } F_f = -F_r \neq 0, \\ \theta + \phi_g, & \text{if } F_f = F_r = 0, \end{cases} \tag{34}$$

and

$$u_2 = -k_\beta(\beta - \beta_d). \tag{35}$$

In the case of rear-wheel driving, if $F = 0$ choice (34) 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. (34). ■

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 2 *In the absence of obstacles ($F_r = 0$), assume a* potential function $U_{\mathrm{a}} : \mathbb{R}^2 \to \mathbb{R}$ *is defined with the following properties:*

(1.) $U_{\mathrm{a}}(x_g, y_g) = 0$
(2.) $U_{\mathrm{a}}(x, y) > 0, \ \forall(x, y) \neq (x_g, y_g)$
(3.) $\nabla U_{\mathrm{a}}(x, y) \neq 0, \ \forall(x, y) \neq (x_g, y_g),$

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

$$F(x,y) = \begin{cases} -\nabla U_{\rm a}(x,y), & \textit{if } (x,y) \neq (x_g, y_g), \\ 0, & \textit{if } (x,y) = (x_g, y_g), \end{cases}$$

along with the control law (28–29) and the associated definitions, drives the car-like robot (24) to G parking the steering angle ϕ to a desired ϕ_g.

Proof. Define the Lyapunov-like function

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

Then, by eqs. (24) and (32)

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

Since $F_r = 0$, we have $M = 0$ and $\dot{\theta}_d = 0$, and hence from eq. (28) and (31)

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

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. (34), iff $\phi = \phi_g$. By LaSalle's theorem, the result follows. ∎

Theorem 2 applies for example when F is the negated gradient of a potential with paraboloidic profile (7) or conical profile (8).

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. (32) is impossible with bounded inputs (see also eq. (33)). In order to overcome this shortcoming, we remove $\dot{\beta}_d$ from eq. (32), setting u_2 as in eq. (35). In the performed simulations, this modification did not appreciably change the robot behavior.

6 Simulation Results

6.1 Unicycle

We simulated the proposed planner for a unicycle in a scene with circular obstacles, although both the artificial potential and the vortex field methods apply to obstacles of arbitrary shape. Input *saturations* were included, with bounds on u_1 and u_2 respectively at 2 m/sec and 360°/sec, the bounds of Nomad scaled by a factor of 5 to speed up simulations. The steering saturation limits the reorientation capability of the vehicle. Controller gains were always set to $k_p = 1$, $k_\theta = 5$, while the holonomic planner parameters were $k_a = 1$, $k_r = 2$, $\gamma = 2$, $\eta_0 = 2$ m. Integration was performed using the fifth order Runge-Kutta method of SIMULINK, with sampling interval T_c=0.001÷0.01 sec.

In Figs. 3 and 4, we first show for comparison two successful outputs of the *holonomic* planners obtained respectively with artificial potentials and with vortex fields. The two paths are topologically different, due to the choice of a CW vortex direction for the first encountered

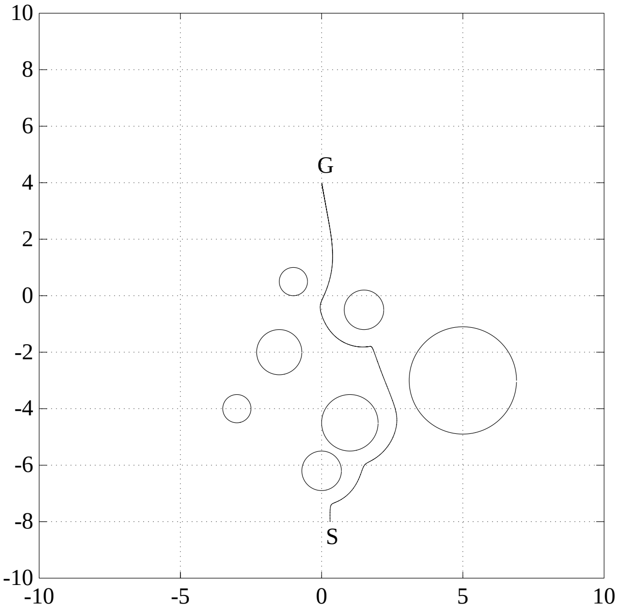

Figure 12: Holonomic motion using potential fields (Example I).

obstacle. The *nonholonomic* motions obtained for $\theta(0) = 0$ are shown respectively in Figs. 5 and 6, with the associated control inputs. In both cases the nonholonomic motion closely approximates the holonomic one, and the position error exponentially goes to zero in the terminal phase. Note that the unicycle in Fig. 15 undergoes initially a large reorientation, because its heading opposes the chosen vortex direction (CW) of the first obstacle. Correspondingly, the steering input saturates. This suggests more in general to keep into account the actual vehicle orientation when deciding the vortex direction for a sensed obstacle.

While navigating among obstacles, the driving velocity obtained using vortex fields is saturated at all instants. This is due to the *non-repulsive* nature of the vortex: when approaching an obstacle the robot does not experience an opposing field. On the other hand, artificial potential fields tend to slow down motion when facing an obstacle. A nonholonomic planner based on vortex fields is thus expected to generate *faster* point-to-point motions.

In some cases, the performance of the two holonomic planners is drastically different. Using artificial potentials, a blocking saddle point can be met; this motivates the unsuccessful output of the nonholonomic planner in Fig. 16. Three large reorientations occur near saddle points located in front of obstacles, before the motion definitely stops. Instead, in Fig. 17 a solution path is obtained with the scheme based on vortex fields.

We have applied the proposed method to several other situations, and the performance was always satisfactory. Accurate tuning of the controller gains was not necessary, but we argue that it may be needed to handle complex situations (e.g., when only sudden reorientation would avoid collision).

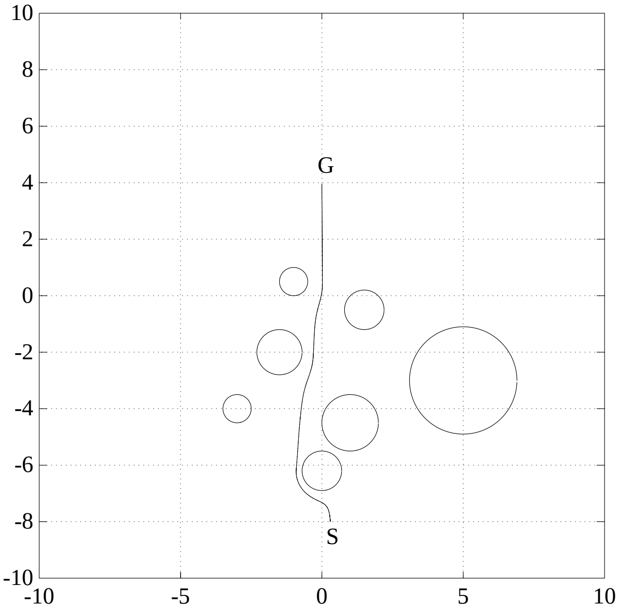

Figure 13: Holonomic motion using vortex fields (Example I).

Field Type	max $\|u_r\|$	max $\|u_\phi\|$
Strictly 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

6.2 Car-like

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 as using SIMULINK and the fifth-order Runge-Kutta method, with sampling interval $T_c = 0.001 \div 0.01$ sec.

In Fig. 18 we report the results obtained with the fields (9), (11), and (12) proposed in Sect. 3. All of them are successful, even if the path generated by the strictly 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. (32) by setting u_2 as in eq. (35). 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 strictly repulsive field this time resulted in about 20 s. However, in comparing these values it is fair to consider 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. 19), the rightmost obstacle of Fig. 18 was slightly shifted to the left. The strictly 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. 20, the rightmost obstacle was shifted further to the left. The vortex field method (11) fails as the robot touches the obstacle ($\|F_{\text{vor}}\| \to \infty$). This is avoided when the circumventive field (12) is used, thanks to its repulsive component.

7 Conclusions

We have proposed a general integrated approach for locally planning the motion of wheeled mobile robots among obstacles. The nonholonomic planner consists of two basic components: a local holonomic planner generating an incremental output, and an on-line projection scheme which modifies it so to yield a feasible path together with the corresponding control inputs. As a result, a hierarchical feedback scheme is obtained which can be used in real-time by feeding sensor data simply to the holonomic planner. This is derived by attractive, repulsive, and vortex field methods, whose combined use produces the desired obstacle avoidance action. The projection scheme have been devised so to avoid generation of additional blocking points. The feedback component of the scheme is guaranteed to stabilize the robot to a given cartesian position, in the absence of obstacles.

This approach has been applied to the case of a unicycle with circular symmetry, and of a car-like robot. Simulation results has shown 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 the NOMAD 200 (unicycle) and on a laboratory prototype (car-like), where state feedback is obtained by integrating ultrasonic and odometric measurements, and the extension to other vehicles, e.g. the car with N trailers.

8 Acknowledgments

This work is supported by ESPRIT III Basic Research Project #6546 (PROMotion) and MURST 40% funds.

References

[1] J.I. Neimark and N.A. Fufaev, *Dynamics of Nonholonomic Systems*, Transl. of Math. Monographs, vol. 33, American Mathematical Society, Providence, 1972.

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

[3] 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.

[4] R.M. Murray and S.S. Sastry, "Nonholonomic motion planning: steering using sinusoids," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 700–716, 1993.

[5] G. Laferriere and H.J. Sussmann, "Motion planning for controllable systems without drift: A preliminary report," Rep. SYSCON-90-04, Rutgers Univ., NJ, 1990.

[6] S. Monaco and D. Normand-Cyrot, "An introduction to motion planning under multirate digital control," *31th IEEE Conf. Decision Contr.*, Tucson, AZ, pp. 1780–1785, 1992.

[7] 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.

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

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

[10] J.-B. Pomet, B. Thuilot, G. Bastin, and G. Campion, "A hybrid strategy for the feedback stabilization of nonholonomic mobile robots," *1992 IEEE Int. Conf. Robotics Automat.*, Nice, F, pp. 129–134, 1992.

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

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

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

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

[15] J. Barraquand, B. Langlois, and J.C. Latombe, "Numerical potential fields techniques for robot path planning," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 224–241, 1992.

[16] 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.

[17] P. Jacobs, J.P. Laumond, and M. Taïx, "Efficient motion planners for nonholonomic mobile robots," *1991 IEEE/RSJ Int. Work. Intell. Robots Syst.*, Osaka, J, pp. 1229–1235, 1991.

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

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

[20] A.M. Bloch, M. Reyhanoglu and N.H. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1746–1757, 1992.

[21] O.J. Sørdalen, “Conversion of the kinematics of a car with N trailers into a chained form,” *1993 IEEE Int. Conf. Robotics Automat.*, Atlanta, GA, vol. 1, pp. 382–387, 1993.

[22] G. Oriolo, “The reactive vortex fields method for robot motion planning with uncertainty”, *36th ANIPLA Annual Conf.*, Genoa, I, pp. 584–597, 1992.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] A. Bemporad, A. De Luca and G. Oriolo, “Local incremental planning for a car-like robot navigating among obstacles,” *1996 IEEE Int. Conf. on Robotics and Automation*, Minneapolis, pp. 1205–1211, 1996.

[28] 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.

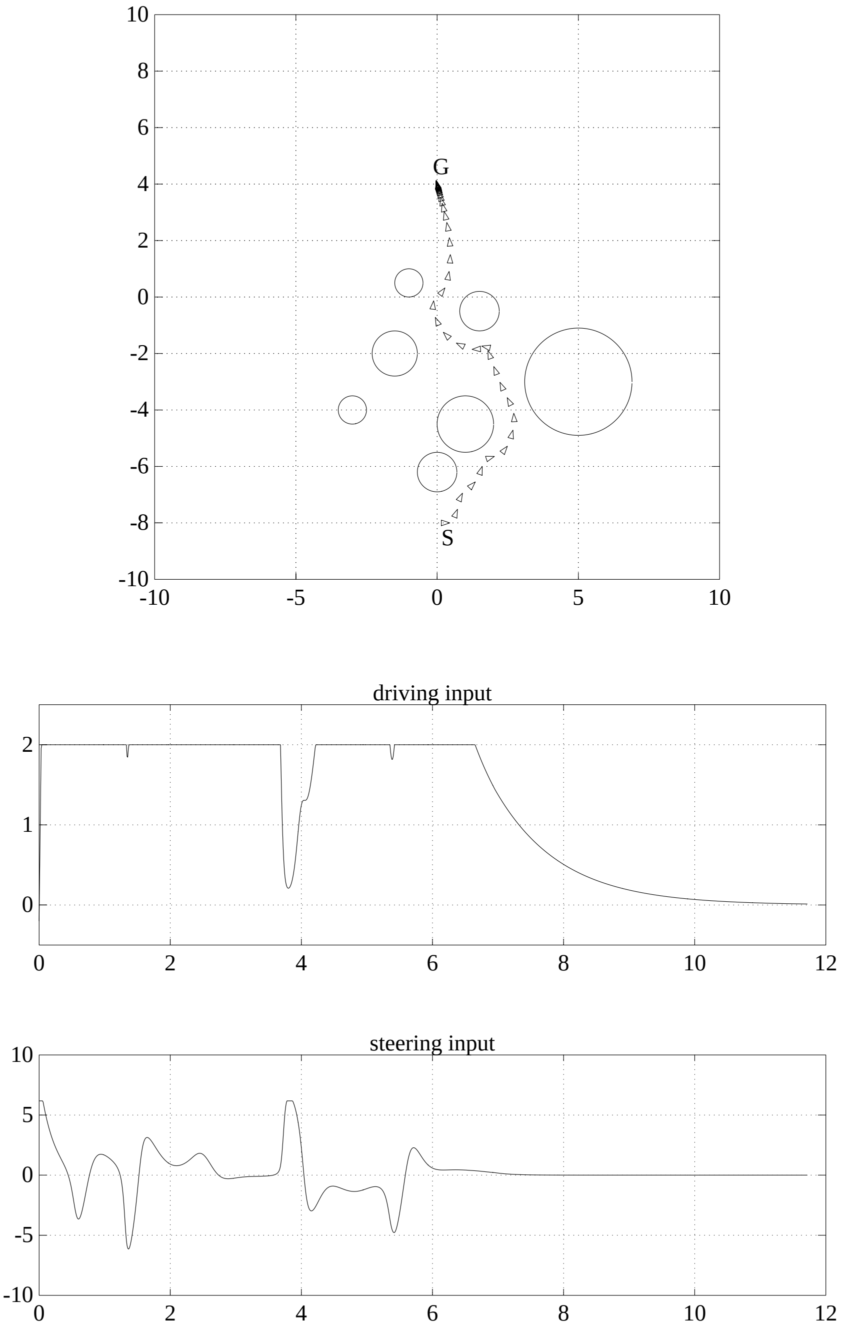

Figure 14: Nonholonomic planning with potential fields (Example I).

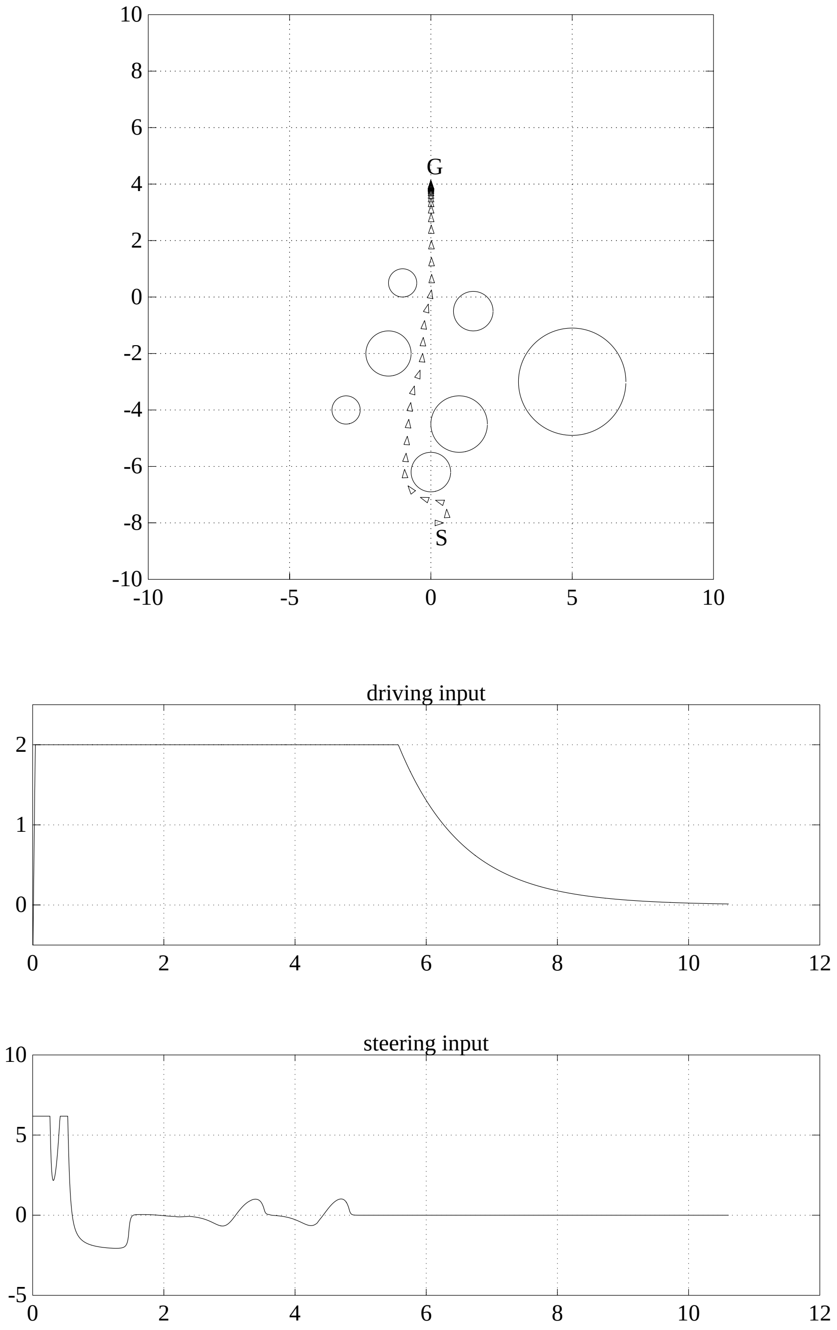

Figure 15: Nonholonomic planning with vortex fields (Example I).

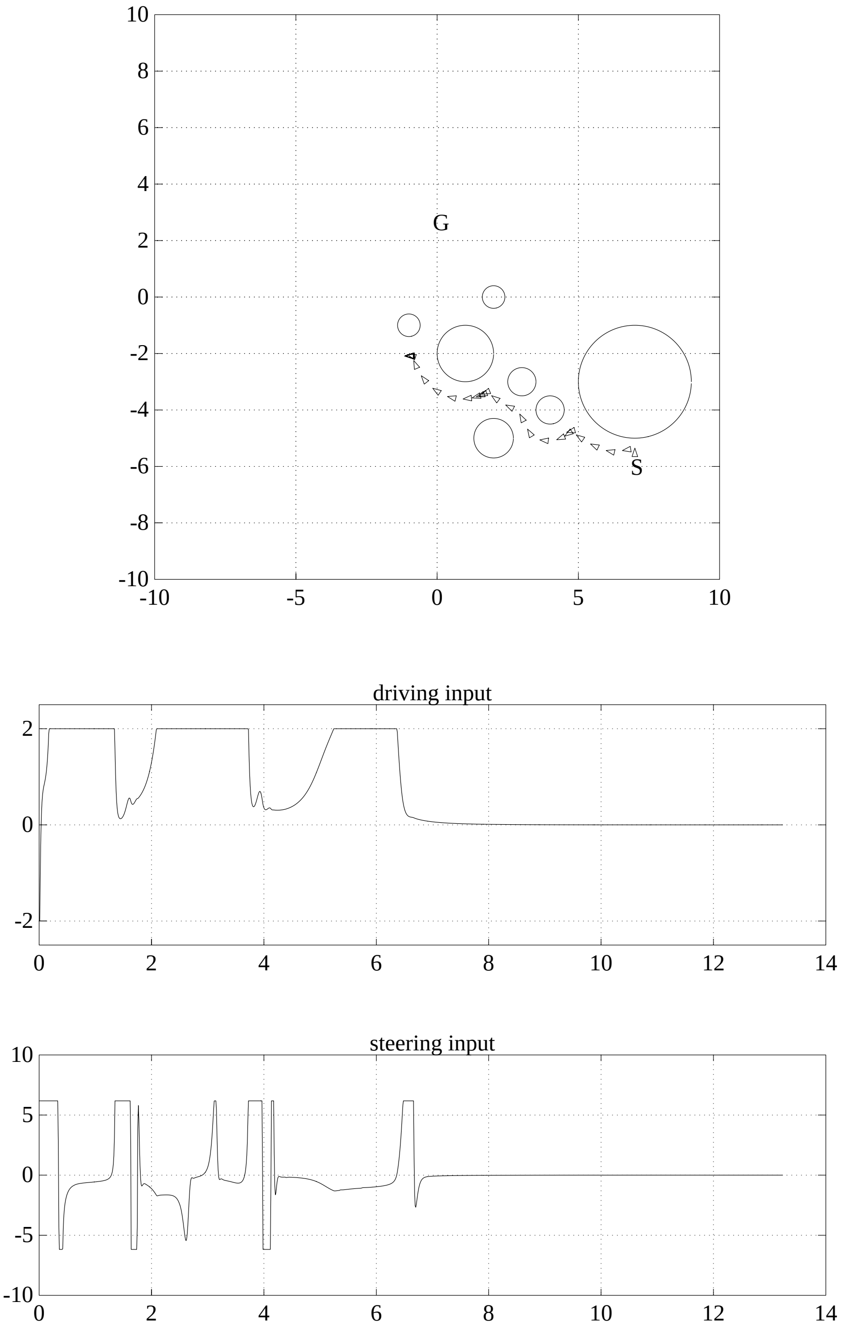

Figure 16: Nonholonomic planning with potential fields (Example II).

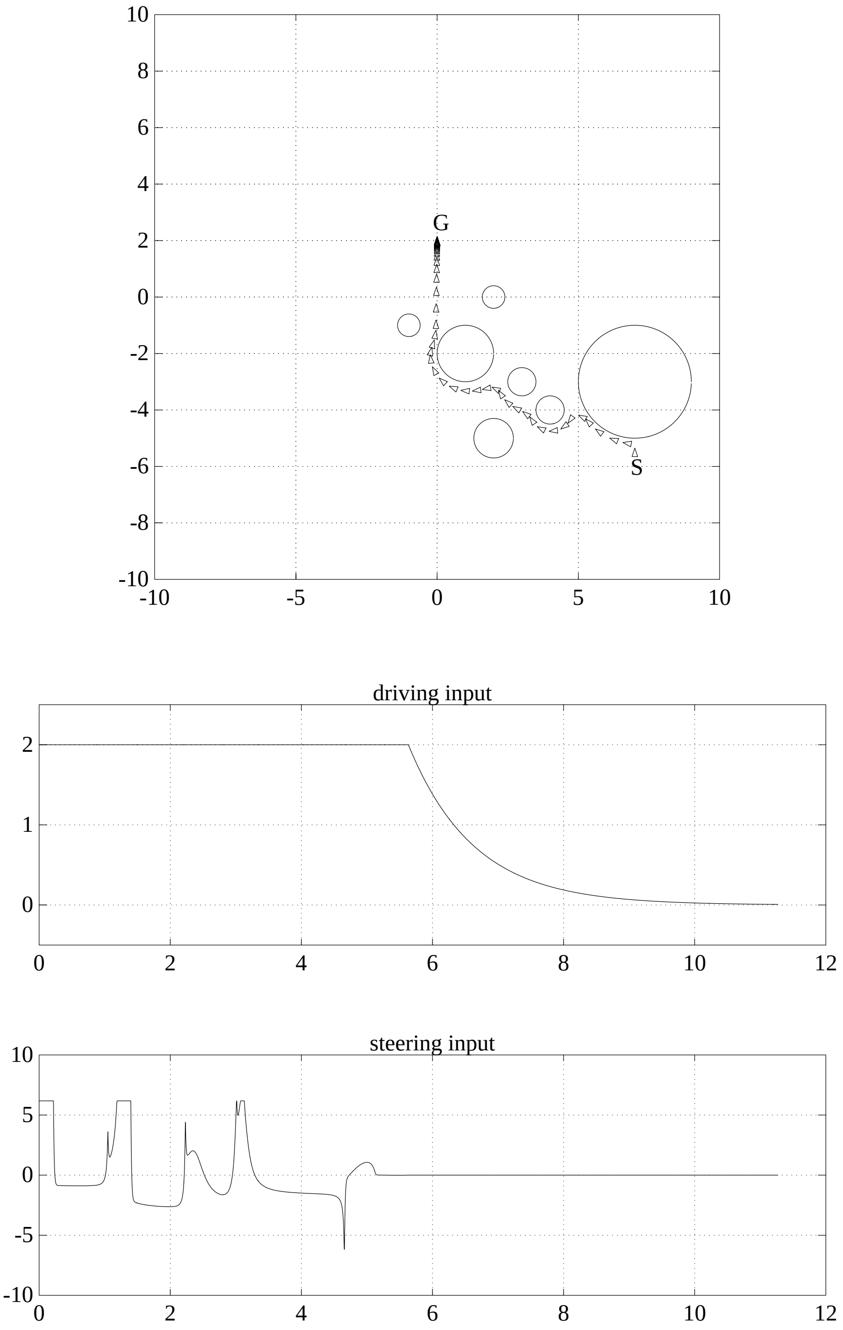

Figure 17: Nonholonomic planning with vortex fields (Example II).

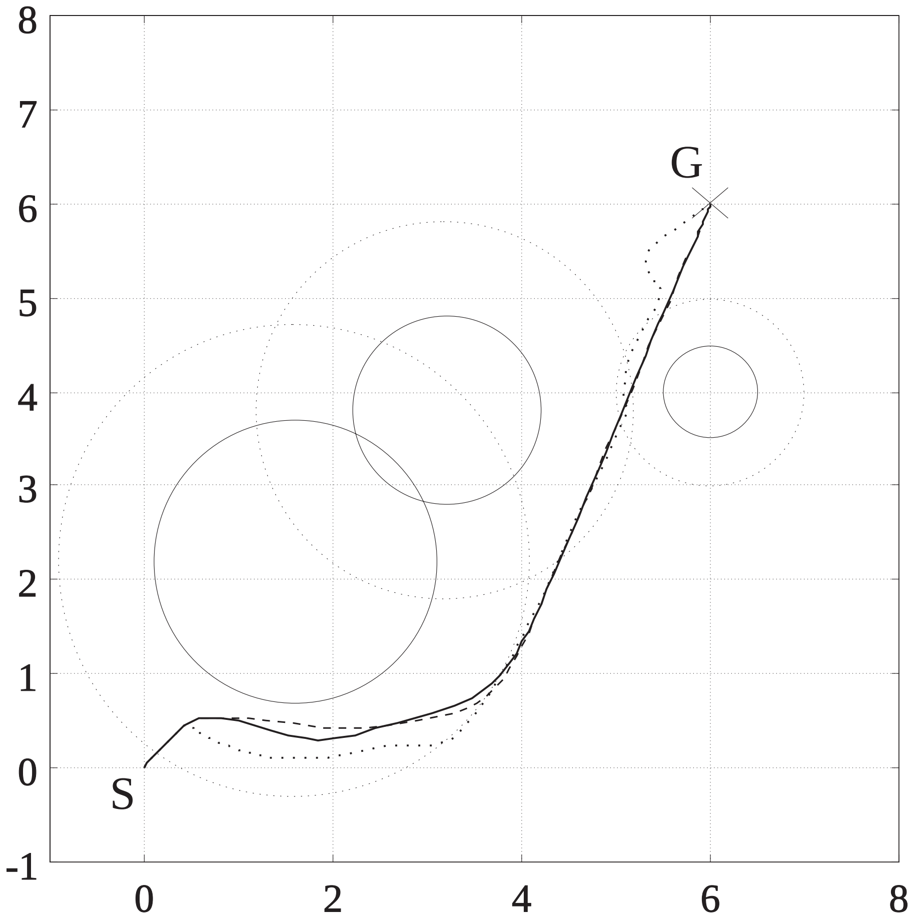

Figure 18: Results with repulsive (dotted), vortex (dashed), and circumventive (solid) fields (Example III).

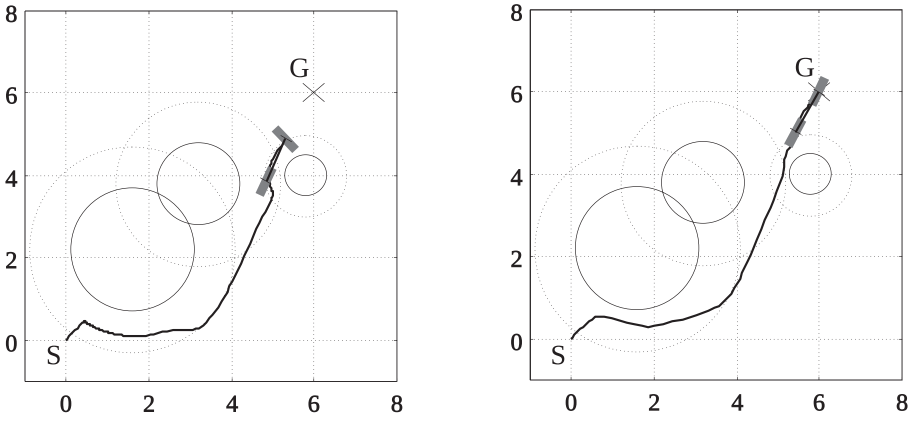

Figure 19: Results with strictly repulsive (left) and circumventive (right) field method (Example IV).

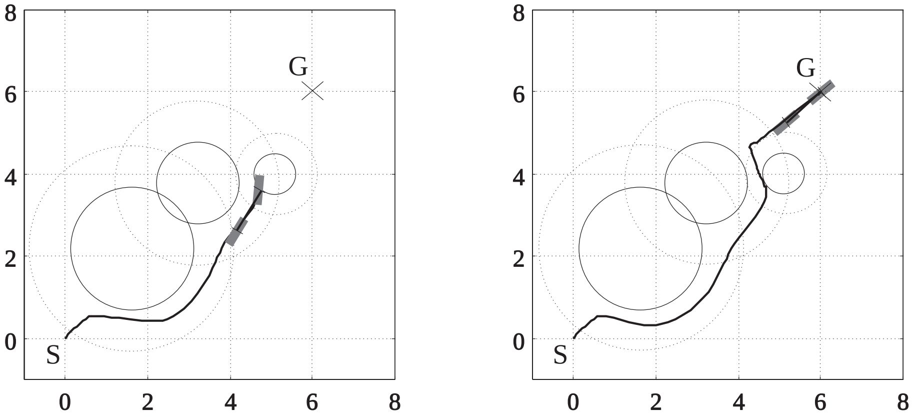

Figure 20: Third simulation: results with vortex field (11) (left) and circumventive (right) field (12) method (Example IV).