

Mobile Robots

The previous chapters deal mainly with articulated manipulators that represent the large majority of robots used in industrial settings. However, *mobile robots* are becoming increasingly important in advanced applications, in view of their potential for autonomous intervention. This chapter presents techniques for modelling, planning and control of *wheeled* mobile robots. The structure of the kinematic constraints arising from the pure rolling of the wheels is first analyzed; it is shown that such constraints are in general *non-holonomic* and consequently reduce the local mobility of the robot. The *kinematic model* associated with the constraints is introduced to describe the instantaneous admissible motions, and conditions are given under which it can be put in *chained form*. The *dynamic model*, that relates the admissible motions to the generalized forces acting on the robot DOFs, is then derived. The peculiar nature of the kinematic model, and in particular the existence of *flat outputs*, is exploited to devise *trajectory planning* methods that guarantee that the nonholonomic constraints are satisfied. The structure of *minimum-time trajectories* is also analyzed. The *motion control* problem for mobile robots is then discussed, with reference to two basic motion tasks, i.e., *trajectory tracking* and *posture regulation*. The chapter concludes by surveying some techniques for *odometric localization* that is necessary to implement feedback control schemes.

11.1 Nonholonomic Constraints

Wheels are by far the most common mechanism to achieve locomotion in mobile robots. Any wheeled vehicle is subject to kinematic constraints that reduce in general its local mobility, while leaving intact the possibility of reaching arbitrary configurations by appropriate manoeuvres. For example, any driver knows by experience that, while it is impossible to move instantaneously a car in the direction orthogonal to its heading, it is still possible to

park it arbitrarily, at least in the absence of obstacles. It is therefore important to analyze in detail the structure of these constraints.

In accordance with the terminology introduced in Sect. B.4, consider a mechanical system whose *configuration* $\mathbf{q} \in \mathcal{C}$ is described by a vector of *generalized coordinates*, and assume that the *configuration space* \mathcal{C} (i.e., the space of all possible robot configurations) coincides¹ with \mathbb{R}^n . The motion of the system that is represented by the evolution of \mathbf{q} over time may be subject to constraints that can be classified under various criteria. For example, they may be expressed as equalities or inequalities (respectively, *bilateral* or *unilateral* constraints), and they may depend explicitly on time or not (*rheonomic* or *scleronomic* constraints). In this chapter, only *bilateral scleronomic* constraints will be considered.

Constraints that can be put in the form

$$h_i(\mathbf{q}) = 0 \quad i = 1, \dots, k < n \quad (11.1)$$

are called *holonomic* (or *integrable*). In the following, it is assumed that the functions $h_i : \mathcal{C} \mapsto \mathbb{R}$ are of class C^∞ (*smooth*) and independent. The effect of holonomic constraints is to reduce the space of accessible configurations to a subset of \mathcal{C} with dimension $n - k$. A mechanical system for which all the constraints can be expressed in the form (11.1) is called *holonomic*.

In the presence of holonomic constraints, the implicit function theorem can be used in principle to solve the equations in (11.1) by expressing k generalized coordinates as a function of the remaining $n - k$, so as to eliminate them from the formulation of the problem. However, in general this procedure is only valid locally, and may introduce singularities. A convenient alternative is to replace the original generalized coordinates with a reduced set of $n - k$ new coordinates that are directly defined on the accessible subspace, in such a way that the available DOFs are effectively characterized. The mobility of the reduced system thus obtained is completely equivalent to that of the original mechanism.

Holonomic constraints are generally the result of mechanical interconnections between the various bodies of the system. For example, prismatic and revolute joints used in robot manipulators are a typical source of such constraints, and joint variables are an example of reduced sets of coordinates in the above sense. Constraints of the form (11.1) may also arise in particular operating conditions; for example, one may mention the case of a kinematically redundant manipulator that moves while keeping the end-effector fixed at a certain pose (*self-motion*).

Constraints that involve generalized coordinates and velocities

$$a_i(\mathbf{q}, \dot{\mathbf{q}}) = 0 \quad i = 1, \dots, k < n$$

¹ This assumption is taken for simplicity. In the general case, the configuration space \mathcal{C} may be identified with a Euclidean space only on a local basis, because its global geometric structure is more complex; this will be further discussed in Chap. 12. The material presented in this chapter is, however, still valid.

are called *kinematic*. They constrain the instantaneous admissible motion of the mechanical system by reducing the set of generalized velocities that can be attained at each configuration. Kinematic constraints are generally expressed in *Pfaffian form*, i.e., they are linear in the generalized velocities:

$$\mathbf{a}_i^T(\mathbf{q})\dot{\mathbf{q}} = 0 \quad i = 1, \dots, k < n, \quad (11.2)$$

or, in *matrix form*

$$\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}. \quad (11.3)$$

Vectors $\mathbf{a}_i : \mathcal{C} \mapsto \mathbb{R}^n$ are assumed to be smooth as well as linearly independent.

Clearly, the existence of k holonomic constraints (11.1) implies that of an equal number of kinematic constraints:

$$\frac{dh_i(\mathbf{q})}{dt} = \frac{\partial h_i(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = 0 \quad i = 1, \dots, k.$$

However, the converse is not true in general. A system of kinematic constraints in the form (11.3) may or may not be integrable to the form (11.1). In the negative case, the kinematic constraints are said to be *nonholonomic* (or *non-integrable*). A mechanical system that is subject to at least one such constraint is called *nonholonomic*.

Nonholonomic constraints reduce the mobility of the mechanical system in a completely different way with respect to holonomic constraints. To appreciate this fact, consider a single Pfaffian constraint

$$\mathbf{a}^T(\mathbf{q})\dot{\mathbf{q}} = 0. \quad (11.4)$$

If the constraint is holonomic, it can be integrated and written as

$$h(\mathbf{q}) = c, \quad (11.5)$$

where $\partial h / \partial \mathbf{q} = \gamma(\mathbf{q}) \mathbf{a}^T(\mathbf{q})$, with $\gamma(\mathbf{q}) \neq 0$ an *integrating factor* and c an *integration constant*. Therefore, there is a loss of *accessibility* in the configuration space, because the motion of the mechanical system in \mathcal{C} is confined to a particular *level surface* of the scalar function h . This surface, which depends on the initial configuration \mathbf{q}_0 through the value of $h(\mathbf{q}_0) = c$, has dimension $n - 1$.

Assume instead that the constraint (11.4) is *nonholonomic*. In this case, generalized velocities are indeed constrained to belong to a subspace of dimension $n - 1$, i.e., the null space of matrix $\mathbf{a}^T(\mathbf{q})$. Nevertheless, the fact that the constraint is *non-integrable* means that there is *no loss of accessibility in \mathcal{C} for the system*. In other words, while the number of DOFs decreases to $n - 1$ due to the constraint, the *number of generalized coordinates cannot be reduced*, not even locally.

The conclusion just drawn for the case of a single constraint is general. An n -dimensional mechanical system subject to k nonholonomic constraints

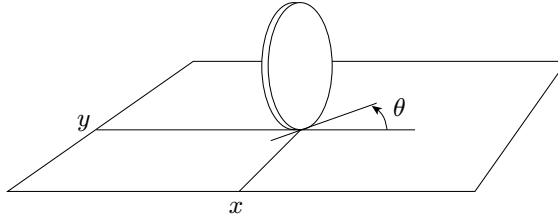


Fig. 11.1. Generalized coordinates for a disk rolling on a plane

can access its whole configuration space \mathcal{C} , although at any configuration its generalized velocities must belong to an $(n - k)$ -dimensional subspace.

The following is a classical example of nonholonomic mechanical system, that is particularly relevant in the study of mobile robots.

Example 11.1

Consider a disk that rolls without slipping on the horizontal plane, while keeping its *sagittal plane* (i.e., the plane that contains the disk) in the vertical direction (Fig. 11.1). Its configuration is described by three² generalized coordinates: the Cartesian coordinates (x, y) of the contact point with the ground, measured in a fixed reference frame, and the angle θ characterizing the orientation of the disk with respect to the x axis. The configuration vector is therefore $\mathbf{q} = [x \ y \ \theta]^T$.

The *pure rolling* constraint for the disk is expressed in the Pfaffian form as

$$\dot{x} \sin \theta - \dot{y} \cos \theta = [\sin \theta \quad -\cos \theta \quad 0] \dot{\mathbf{q}} = 0, \quad (11.6)$$

and entails that, in the absence of slipping, the velocity of the contact point has zero component in the direction orthogonal to the sagittal plane. The angular velocity of the disk around the vertical axis instead is unconstrained.

Constraint (11.6) is **nonholonomic**, because it implies no loss of accessibility in the configuration space of the disk. To substantiate this claim, consider that the disk can be driven from any initial configuration $\mathbf{q}_i = [x_i \ y_i \ \theta_i]^T$ to any final configuration $\mathbf{q}_f = [x_f \ y_f \ \theta_f]^T$ through the following sequence of movements that do not violate constraint (11.6):

1. rotate the disk around its vertical axis so as to reach the orientation θ_v for which the *sagittal axis* (i.e., the intersection of the sagittal plane and the horizontal plane) goes through the final contact point (x_f, y_f) ;
2. roll the disk on the plane at a constant orientation θ_v until the contact point reaches its final position (x_f, y_f) ;
3. rotate again the disk around its vertical axis to change the orientation from θ_v to θ_f .

² One could add to this description an angle ϕ measuring the rotation of the disk around the horizontal axis passing through its centre. Such a coordinate is however irrelevant for the analysis presented in this chapter, and is therefore ignored in the following.

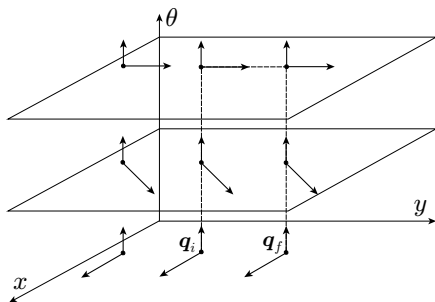


Fig. 11.2. A local representation of the configuration space for the rolling disk with an example manoeuvre that transfers the configuration from \mathbf{q}_i to \mathbf{q}_f (dashed line)

An example of this manoeuvre is shown in Fig. 11.2. Two possible directions of instantaneous motion are shown at each configuration: the first, that is aligned with the sagittal axis, moves the contact point while keeping the orientation constant (rolling); the second varies the orientation while keeping the contact point fixed (rotation around the vertical axis).

It is interesting to note that, in addition to wheeled vehicles, there exist other robotic systems that are nonholonomic in nature. For example, the pure rolling constraint also arises in manipulation problems with round-fingered robot hands. Another kind of nonholonomic behaviour is found in multibody systems that ‘float’ freely (i.e., without a fixed base), such as manipulators used in space operations. In fact, in the absence of external generalized forces, the conservation of the angular momentum represents a non-integrable Pfaffian constraint for the system.

11.1.1 Integrability Conditions

In the presence of Pfaffian kinematic constraints, integrability conditions can be used to decide whether the system is holonomic or nonholonomic.

Consider first the case of a *single* Pfaffian constraint:

$$\mathbf{a}^T(\mathbf{q})\dot{\mathbf{q}} = \sum_{j=1}^n a_j(\mathbf{q})\dot{q}_j = 0. \quad (11.7)$$

For this constraint to be integrable, there must exist a scalar function $h(\mathbf{q})$ and an integrating factor $\gamma(\mathbf{q}) \neq 0$ such that the following condition holds:

$$\gamma(\mathbf{q})a_j(\mathbf{q}) = \frac{\partial h(\mathbf{q})}{\partial q_j} \quad j = 1, \dots, n. \quad (11.8)$$

The converse is also true: if there exists an integrating factor $\gamma(\mathbf{q}) \neq 0$ such that $\gamma(\mathbf{q})\mathbf{a}(\mathbf{q})$ is the gradient of a scalar function $h(\mathbf{q})$, constraint (11.7) is integrable. By using Schwarz theorem on the symmetry of second derivatives, the integrability condition (11.8) may be replaced by the following system of partial differential equations:

$$\frac{\partial(\gamma a_k)}{\partial q_j} = \frac{\partial(\gamma a_j)}{\partial q_k} \quad j, k = 1, \dots, n, \quad j \neq k, \quad (11.9)$$

that does not contain the unknown function $h(\mathbf{q})$. Note that condition (11.9) implies that a Pfaffian constraint with constant coefficients a_j is always holonomic.

Example 11.2

Consider the following kinematic constraint in $\mathcal{C} = \mathbb{R}^3$:

$$\dot{q}_1 + q_1 \dot{q}_2 + \dot{q}_3 = 0.$$

The holonomy condition (11.9) gives

$$\begin{aligned} \frac{\partial \gamma}{\partial q_2} &= \gamma + q_1 \frac{\partial \gamma}{\partial q_1} \\ \frac{\partial \gamma}{\partial q_3} &= \frac{\partial \gamma}{\partial q_1} \\ q_1 \frac{\partial \gamma}{\partial q_3} &= \frac{\partial \gamma}{\partial q_2}. \end{aligned}$$

By substituting the second and third equations into the first, it is easy to conclude that the only solution is $\gamma = 0$. Therefore, the constraint is nonholonomic.

Example 11.3

Consider the pure rolling constraint (11.6). In this case, the holonomy condition (11.9) gives

$$\begin{aligned} \sin \theta \frac{\partial \gamma}{\partial y} &= -\cos \theta \frac{\partial \gamma}{\partial x} \\ \cos \theta \frac{\partial \gamma}{\partial \theta} &= \gamma \sin \theta \\ \sin \theta \frac{\partial \gamma}{\partial \theta} &= -\gamma \cos \theta. \end{aligned}$$

Squaring and adding the last two equations gives $\partial \gamma / \partial \theta = \pm \gamma$. Assume for example $\partial \gamma / \partial \theta = \gamma$. Using this in the above equations leads to

$$\begin{aligned} \gamma \cos \theta &= \gamma \sin \theta \\ \gamma \sin \theta &= -\gamma \cos \theta \end{aligned}$$

whose only solution is $\gamma = 0$. The same conclusion is reached by letting $\partial \gamma / \partial \theta = -\gamma$. This confirms that constraint (11.6) is nonholonomic.

The situation becomes more complicated when dealing with a system of $k > 1$ kinematic constraints of the form (11.3). In fact, in this case it may happen that the single constraints are not integrable if taken separately, but the whole system is integrable. In particular, if $p \leq k$ independent linear combinations of the constraints

$$\sum_{i=1}^k \gamma_{ji}(\mathbf{q}) \mathbf{a}_i^T(\mathbf{q}) \dot{\mathbf{q}} \quad j = 1, \dots, p$$

are integrable, there exist p independent scalar functions $h_j(\mathbf{q})$ such that

$$\text{span} \left\{ \frac{\partial h_1(\mathbf{q})}{\partial \mathbf{q}}, \dots, \frac{\partial h_p(\mathbf{q})}{\partial \mathbf{q}} \right\} \subset \text{span} \{ \mathbf{a}_1^T(\mathbf{q}), \dots, \mathbf{a}_k^T(\mathbf{q}) \} \quad \forall \mathbf{q} \in \mathcal{C}.$$

Therefore, the configurations that are accessible for the mechanical system belong to the $(n - p)$ -dimensional subspace consisting of the particular level surfaces of the functions h_j :

$$\{ \mathbf{q} \in \mathcal{C} : h_1(\mathbf{q}) = c_1, \dots, h_p(\mathbf{q}) = c_p \}$$

on which the motion is started (see Problem 11.2). In the case $p = k$, the system of kinematic constraints (11.3) is completely integrable, and hence holonomic.

Example 11.4

Consider the system of Pfaffian constraints

$$\begin{aligned} \dot{q}_1 + q_1 \dot{q}_2 + \dot{q}_3 &= 0 \\ \dot{q}_1 + \dot{q}_2 + q_1 \dot{q}_3 &= 0. \end{aligned}$$

Taken separately, these constraints are found to be non-integrable (in particular, the first is the nonholonomic constraint of Example 11.2). However, subtracting the second from the first gives

$$(q_1 - 1)(\dot{q}_2 - \dot{q}_3) = 0$$

so that $\dot{q}_2 = \dot{q}_3$, because the constraints must be satisfied for any value of \mathbf{q} . The assigned system of constraints is then equivalent to

$$\begin{aligned} \dot{q}_2 &= \dot{q}_3 \\ \dot{q}_1 + (1 + q_1)\dot{q}_2 &= 0, \end{aligned}$$

which can be integrated as

$$\begin{aligned} q_2 - q_3 &= c_1 \\ \log(q_1 + 1) + q_2 &= c_2 \end{aligned}$$

with integration constants c_1, c_2 .

The integrability conditions of a system of Pfaffian kinematic constraints are quite complex and derive from a fundamental result of differential geometry known as *Frobenius theorem*. However, as shown later in the chapter, it is possible to derive such conditions more directly from a different perspective.

11.2 Kinematic Model

The system of k Pfaffian constraints (11.3) entails that the admissible generalized velocities at each configuration \mathbf{q} belong to the $(n - k)$ -dimensional null space of matrix $\mathbf{A}^T(\mathbf{q})$. Denoting by $\{\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_{n-k}(\mathbf{q})\}$ a basis of $\mathcal{N}(\mathbf{A}^T(\mathbf{q}))$, the admissible trajectories for the mechanical system can then be characterized as the solutions of the nonlinear dynamic system

$$\dot{\mathbf{q}} = \sum_{j=1}^m \mathbf{g}_j(\mathbf{q}) u_j = \mathbf{G}(\mathbf{q}) \mathbf{u} \quad m = n - k, \quad (11.10)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the state vector and $\mathbf{u} = [u_1 \ \dots \ u_m]^T \in \mathbb{R}^m$ is the input vector. System (11.10) is said to be *driftless* because one has $\dot{\mathbf{q}} = \mathbf{0}$ if the input is zero.

The choice of the *input vector fields* $\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_m(\mathbf{q})$ (and thus of matrix $\mathbf{G}(\mathbf{q})$) in (11.10) is not unique. Correspondingly, the components of \mathbf{u} may have different meanings. In general, it is possible to choose the basis of $\mathcal{N}(\mathbf{A}^T(\mathbf{q}))$ in such a way that the u_j s have a physical interpretation, as will be shown later for some examples of mobile robots. In any case, vector \mathbf{u} may not be directly related to the actual control inputs, that are in general forces and/or torques. For this reason, Eq. (11.10) is referred to as the *kinematic model* of the constrained mechanical system.

The holonomy or nonholonomy of constraints (11.3) can be established by analyzing the controllability³ properties of the associated kinematic model (11.10). In fact, two cases are possible:

1. If system (11.10) is controllable, given two arbitrary configurations \mathbf{q}_i and \mathbf{q}_f in \mathcal{C} , there exists a choice of $\mathbf{u}(t)$ that steers the system from \mathbf{q}_i to \mathbf{q}_f , i.e., there exists a trajectory that joins the two configurations and satisfies the kinematic constraints (11.3). Therefore, these do not affect in any way the accessibility of \mathcal{C} , and they are (completely) nonholonomic.
2. If system (11.10) is not controllable, the kinematic constraints (11.3) reduce the set of accessible configurations in \mathcal{C} . Hence, the constraints are partially or completely integrable depending on the dimension $\nu < n$ of the accessible configuration space. In particular:

³ Refer to Appendix D for a short survey of nonlinear controllability theory, including the necessary tools from differential geometry.

- 2a. If $m < \nu < n$, the loss of accessibility is not maximal, and thus constraints (11.3) are only partially integrable. The mechanical system is still **nonholonomic**.
- 2b. If $\nu = m$, the loss of accessibility is maximal, and constraints (11.3) are completely integrable. Therefore, the mechanical system is **holonomic**.

Note how this particular viewpoint, i.e., the equivalence between controllability and nonholonomy, was already implicitly adopted in Example 11.1, where the controllability of the kinematic system was proven *constructively*, i.e., by exhibiting a reconfiguration manoeuvre. A more systematic approach is to take advantage of the controllability conditions for nonlinear driftless systems. In particular, **controllability may be verified using the accessibility rank condition**

$$\dim \Delta_{\mathcal{A}}(\mathbf{q}) = n, \quad (11.11)$$

where $\Delta_{\mathcal{A}}$ is the **accessibility distribution** associated with system (11.10), i.e., the involutive closure of distribution $\Delta = \text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_m\}$. The following cases may occur:

1. If (11.11) holds, system (11.10) is controllable and the kinematic constraints (11.3) are (completely) nonholonomic.
2. If (11.11) does not hold, system (11.10) is not controllable and the kinematic constraints (11.3) are at least partially integrable. In particular, let

$$\dim \Delta_{\mathcal{A}}(\mathbf{q}) = \nu < n.$$

Then

- 2a. If $m < \nu < n$, constraints (11.3) are only partially integrable.
- 2b. If $\nu = m$, constraints (11.3) are completely integrable, and hence holonomic. This happens when $\Delta_{\mathcal{A}}$ coincides with $\Delta = \text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_m\}$, i.e., when the latter distribution is involutive.

It is easy to verify that, in the case of a single kinematic constraint (11.7), the integrability condition given by (11.9) is equivalent to the involutivity of $\Delta = \text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_{n-1}\}$. Another remarkable situation is met when the number of Pfaffian constraints is $k = n - 1$; in this case, the associated kinematic model (11.10) consists of a single vector field \mathbf{g} ($m = 1$). Hence, $n - 1$ Pfaffian constraints are always integrable, because the distribution associated with a single vector field is always involutive. For example, a mechanical system with two generalized coordinates that is subject to a scalar Pfaffian constraint is always holonomic.

In the following, the kinematic models of two wheeled vehicles of particular interest will be analyzed in detail. A large part of the existing mobile robots have a kinematic model that is equivalent to one of these two.

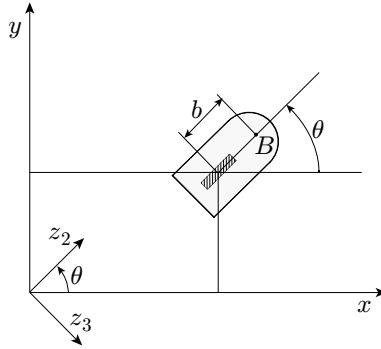


Fig. 11.3. Generalized coordinates for a unicycle

11.2.1 Unicycle

A *unicycle* is a vehicle with a single orientable wheel. Its configuration is completely described by $\mathbf{q} = [x \ y \ \theta]^T$, where (x, y) are the Cartesian coordinates of the contact point of the wheel with the ground (or equivalently, of the wheel centre) and θ is the orientation of the wheel with respect to the x axis (see Fig. 11.3).

As already seen in Example 11.1, the pure rolling constraint for the wheel is expressed as

$$\dot{x} \sin \theta - \dot{y} \cos \theta = [\sin \theta \quad -\cos \theta \quad 0] \dot{\mathbf{q}} = 0, \quad (11.12)$$

entailing that the velocity of the contact point is zero in the direction orthogonal to the sagittal axis of the vehicle. The line passing through the contact point and having such direction is therefore called *zero motion line*. Consider the matrix

$$\mathbf{G}(\mathbf{q}) = [\mathbf{g}_1(\mathbf{q}) \quad \mathbf{g}_2(\mathbf{q})] = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix},$$

whose columns $\mathbf{g}_1(\mathbf{q})$ and $\mathbf{g}_2(\mathbf{q})$ are, for each \mathbf{q} , a basis of the null space of the matrix associated with the Pfaffian constraint. All the admissible generalized velocities at \mathbf{q} are therefore obtained as a linear combination of $\mathbf{g}_1(\mathbf{q})$ and $\mathbf{g}_2(\mathbf{q})$. The kinematic model of the unicycle is then

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad (11.13)$$

where the inputs v and ω have a clear physical interpretation. In particular, v is the *driving velocity*, i.e., the modulus⁴ (with sign) of the contact point

⁴ Note that v is given by the angular speed of the wheel around its horizontal axis multiplied by the wheel radius.

velocity vector, whereas the *steering velocity* ω is the wheel angular speed around the vertical axis.

The Lie bracket of the two input vector fields is

$$[\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}) = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix},$$

that is always linearly independent from $\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q})$. Therefore, the iterative procedure (see Sect. D.2) for building the accessibility distribution $\Delta_{\mathcal{A}}$ ends with

$$\dim \Delta_{\mathcal{A}} = \dim \Delta_2 = \dim \text{span}\{\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]\} = 3.$$

This indicates that the unicycle is controllable with degree of nonholonomy $\kappa = 2$, and that constraint (11.12) is nonholonomic — the same conclusion reached in Example 11.3 by applying the integrability condition.

A unicycle in the strict sense (i.e., a vehicle equipped with a single wheel) is a robot with a serious problem of balance in static conditions. However, there exist vehicles that are kinematically equivalent to a unicycle but more stable from a mechanical viewpoint. Among these, the most important are the *differential drive* and the *synchro drive* vehicles, already introduced in Sect. 1.2.2.

For the differential drive mobile robot of Fig. 1.13, denote by (x, y) the Cartesian coordinates of the midpoint of the segment joining the two wheel centres, and by θ the common orientation of the fixed wheels (hence, of the vehicle body). Then, the kinematic model (11.13) of the unicycle also applies to the differential drive vehicle, provided that the driving and steering velocities v and ω are expressed as a function of the actual velocity inputs, i.e., the angular speeds ω_R and ω_L of the right and left wheel, respectively. Simple arguments (see Problem 11.6) can be used to show that there is a one-to-one correspondence between the two sets of inputs:

$$v = \frac{r(\omega_R + \omega_L)}{2} \quad \omega = \frac{r(\omega_R - \omega_L)}{d}, \quad (11.14)$$

where r is the radius of the wheels and d is the distance between their centres.

The equivalence with the kinematic model (11.13) is even more straightforward for the *synchro drive* mobile robot of Fig. 1.14, whose control inputs are indeed the driving velocity v and the steering velocity ω , that are common to the three orientable wheels. The Cartesian coordinates (x, y) may represent in this case any point of the robot (for example, its centroid), while θ is the common orientation of the wheels. Note that, unlike a differential drive vehicle, the orientation of the body of a synchro drive vehicle never changes, unless a third actuator is added for this specific purpose.

11.2.2 Bicycle

Consider now a *bicycle*, i.e., a vehicle having an orientable wheel and a fixed wheel arranged as in Fig. 11.4. A possible choice for the generalized coordi-

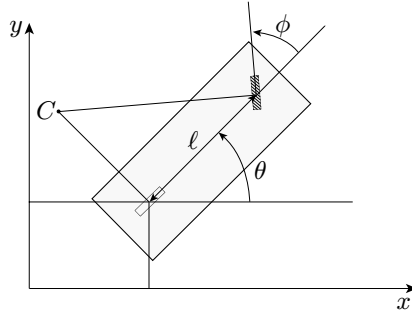


Fig. 11.4. Generalized coordinates and instantaneous centre of rotation for a bicycle

nates is $\mathbf{q} = [x \ y \ \theta \ \phi]^T$, where (x, y) are the Cartesian coordinates of the contact point between the rear wheel and the ground (i.e., of the rear wheel centre), θ is the orientation of the vehicle with respect to the x axis, and ϕ is the steering angle of the front wheel with respect to the vehicle.

The motion of the vehicle is subject to two pure rolling constraints, one for each wheel:

$$\dot{x}_f \sin(\theta + \phi) - \dot{y}_f \cos(\theta + \phi) = 0 \quad (11.15)$$

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0, \quad (11.16)$$

where (x_f, y_f) is the Cartesian position of the centre of the front wheel. The geometric meaning of these constraints is obvious: the velocity of the centre of the front wheel is zero in the direction orthogonal to the wheel itself, while the velocity of the centre of the rear wheel is zero in the direction orthogonal to the sagittal axis of the vehicle. The zero motion lines of the two wheels meet at a point C called *instantaneous centre of rotation* (Fig. 11.4), whose position depends only on (and changes with) the configuration \mathbf{q} of the bicycle. Each point of the vehicle body then moves instantaneously along an arc of circle with centre in C (see also Problem 11.7).

Using the rigid body constraint

$$\begin{aligned} x_f &= x + \ell \cos \theta \\ y_f &= y + \ell \sin \theta, \end{aligned}$$

where ℓ is the distance between the wheels, constraint (11.15) can be rewritten as

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \ell \dot{\theta} \cos \phi = 0. \quad (11.17)$$

The matrix associated with the Pfaffian constraints (11.16), (11.17) is then

$$\mathbf{A}^T(\mathbf{q}) = \begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 \\ \sin(\theta + \phi) & -\cos(\theta + \phi) & -\ell \cos \phi & 0 \end{bmatrix},$$

with constant rank $k = 2$. The dimension of its null space is $n - k = 2$, and all the admissible velocities at \mathbf{q} may be written as a linear combination of a basis of $\mathcal{N}(\mathbf{A}^T(\mathbf{q}))$, for example

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi \\ \sin \theta \cos \phi \\ \sin \phi / \ell \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2.$$

Since the front wheel is orientable, it is immediate to set $u_2 = \omega$, where ω is the *steering* velocity. The expression of u_1 depends instead on how the vehicle is driven.

If the bicycle has *front-wheel drive*, one has directly $u_1 = v$, where v is the *driving* velocity of the front wheel. The corresponding kinematic model is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi \\ \sin \theta \cos \phi \\ \sin \phi / \ell \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega. \quad (11.18)$$

Denoting by $\mathbf{g}_1(\mathbf{q})$ and $\mathbf{g}_2(\mathbf{q})$ the two input vector fields, simple computations give

$$\mathbf{g}_3(\mathbf{q}) = [\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}) = \begin{bmatrix} \cos \theta \sin \phi \\ \sin \theta \sin \phi \\ -\cos \phi / \ell \\ 0 \end{bmatrix} \quad \mathbf{g}_4(\mathbf{q}) = [\mathbf{g}_1, \mathbf{g}_3](\mathbf{q}) = \begin{bmatrix} -\sin \theta / \ell \\ \cos \theta / \ell \\ 0 \\ 0 \end{bmatrix},$$

both linearly independent from $\mathbf{g}_1(\mathbf{q})$ and $\mathbf{g}_2(\mathbf{q})$. Hence, the iterative procedure for building the accessibility distribution $\Delta_{\mathcal{A}}$ ends with

$$\dim \Delta_{\mathcal{A}} = \dim \Delta_3 = \dim \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4\} = 4.$$

This means that the front-wheel drive bicycle is controllable with degree of nonholonomy $\kappa = 3$, and constraints (11.15), (11.16) are (completely) non-holonomic.

The kinematic model of a bicycle with *rear-wheel drive* can be derived by noting that in this case the first two equations must coincide with those of the unicycle model (11.13). It is then sufficient to set $u_1 = v / \cos \phi$ to obtain

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \phi / \ell \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad (11.19)$$

where v is the *driving* velocity of the rear wheel.⁵ In this case, one has

$$\mathbf{g}_3(\mathbf{q}) = [\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\ell \cos^2 \phi} \\ 0 \end{bmatrix} \quad \mathbf{g}_4(\mathbf{q}) = [\mathbf{g}_1, \mathbf{g}_3](\mathbf{q}) = \begin{bmatrix} -\frac{\sin \theta}{\ell \cos^2 \phi} \\ \frac{\cos \theta}{\ell \cos^2 \phi} \\ 0 \\ 0 \end{bmatrix},$$

again linearly independent from $\mathbf{g}_1(\mathbf{q})$ and $\mathbf{g}_2(\mathbf{q})$. Hence, the rear-wheel drive bicycle is also controllable with degree of nonholonomy $\kappa = 3$.

Like the unicycle, the bicycle is also unstable in static conditions. Kinetically equivalent vehicles that are mechanically balanced are the *tricycle* and the *car-like* robot, introduced in Sect. 1.2.2 and shown respectively in Fig. 1.15 and 1.16. In both cases, the kinematic model is given by (11.18) or by (11.19) depending on the wheel drive being on the front or the rear wheels. In particular, (x, y) are the Cartesian coordinates of the midpoint of the rear wheel axle, θ is the orientation of the vehicle, and ϕ is the steering angle.

11.3 Chained Form

The possibility of transforming the kinematic model (11.10) of a mobile robot in a canonical form is of great interest for solving planning and control problems with efficient, systematic procedures. Here, the analysis is limited to systems with two inputs, like the unicycle and bicycle models.

A $(2, n)$ **chained form** is a two-input driftless system

$$\dot{\mathbf{z}} = \gamma_1(\mathbf{z})v_1 + \gamma_2(\mathbf{z})v_2,$$

whose equations are expressed as

$$\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1 \\ &\vdots \\ \dot{z}_n &= z_{n-1} v_1. \end{aligned} \tag{11.20}$$

Using the following notation for a ‘repeated’ Lie bracket:

$$\text{ad}_{\gamma_1} \gamma_2 = [\gamma_1, \gamma_2] \quad \text{ad}_{\gamma_1}^k \gamma_2 = [\gamma_1, \text{ad}_{\gamma_1}^{k-1} \gamma_2],$$

⁵ Note that the kinematic model (11.19) is no longer valid for $\phi = \pm\pi/2$, where the first vector field is not defined. This corresponds to the mechanical jam in which the front wheel is orthogonal to the sagittal axis of the vehicle. This singularity does not arise in the front-wheel drive bicycle (11.18), that in principle can still pivot around the rear wheel contact point in such a situation.

one has for system (11.20)

$$\gamma_1 = \begin{bmatrix} 1 \\ 0 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-1} \end{bmatrix} \quad \gamma_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow \text{ad}_{\gamma_1}^k \gamma_2 = \begin{bmatrix} 0 \\ \vdots \\ (-1)^k \\ \vdots \\ 0 \end{bmatrix},$$

where $(-1)^k$ is the $(k+2)$ -th component. This implies that the system is controllable, because the accessibility distribution

$$\Delta_{\mathcal{A}} = \text{span} \{ \gamma_1, \gamma_2, \text{ad}_{\gamma_1} \gamma_2, \dots, \text{ad}_{\gamma_1}^{n-2} \gamma_2 \}$$

has dimension n . In particular, the degree of nonholonomy is $\kappa = n - 1$.

There exist necessary and sufficient conditions for transforming a generic two-input driftless system

$$\dot{\mathbf{q}} = \mathbf{g}_1(\mathbf{q})u_1 + \mathbf{g}_2(\mathbf{q})u_2 \quad (11.21)$$

in the chained form (11.20) via coordinate and input transformations

$$\mathbf{z} = \mathbf{T}(\mathbf{q}) \quad \mathbf{v} = \boldsymbol{\beta}(\mathbf{q})\mathbf{u}. \quad (11.22)$$

In particular, it can be shown that systems like (11.21) with dimension n not larger than 4 can *always* be put in chained form. This applies, for example, to the kinematic models of the unicycle and the bicycle.

There also exist sufficient conditions for transformability in chained form that are relevant because they are constructive. Define the distributions

$$\begin{aligned} \Delta_0 &= \text{span} \{ \mathbf{g}_1, \mathbf{g}_2, \text{ad}_{\mathbf{g}_1} \mathbf{g}_2, \dots, \text{ad}_{\mathbf{g}_1}^{n-2} \mathbf{g}_2 \} \\ \Delta_1 &= \text{span} \{ \mathbf{g}_2, \text{ad}_{\mathbf{g}_1} \mathbf{g}_2, \dots, \text{ad}_{\mathbf{g}_1}^{n-2} \mathbf{g}_2 \} \\ \Delta_2 &= \text{span} \{ \mathbf{g}_2, \text{ad}_{\mathbf{g}_1} \mathbf{g}_2, \dots, \text{ad}_{\mathbf{g}_1}^{n-3} \mathbf{g}_2 \}. \end{aligned}$$

Assume that, in a certain set, it is $\dim \Delta_0 = n$, Δ_1 and Δ_2 are involutive, and there exists a scalar function $h_1(\mathbf{q})$ whose differential $d\mathbf{h}_1$ satisfies

$$d\mathbf{h}_1 \cdot \Delta_1 = 0 \quad d\mathbf{h}_1 \cdot \mathbf{g}_1 = 1,$$

where the symbol \cdot denotes the inner product between a row vector and a column vector — in particular, $\cdot \Delta_1$ is the inner product with any vector generated by distribution Δ_1 . In this case, system (11.21) can be put in the form (11.20) through the coordinate transformation⁶

$$z_1 = h_1$$

⁶ This transformation makes use of the Lie derivative (see Appendix D).

$$\begin{aligned}
z_2 &= L_{g_1}^{n-2} h_2 \\
&\vdots \\
z_{n-1} &= L_{g_1} h_2 \\
z_n &= h_2,
\end{aligned}$$

where h_2 must be chosen independent of h_1 and such that $dh_2 \cdot \Delta_2 = 0$. The input transformation is given by

$$\begin{aligned}
v_1 &= u_1 \\
v_2 &= \left(L_{g_1}^{n-1} h_2 \right) u_1 + \left(L_{g_2} L_{g_1}^{n-2} h_2 \right) u_2.
\end{aligned}$$

In general, the coordinate and input transformations are not unique.

Consider the kinematic model (11.13) of the unicycle. With the change of coordinates

$$\begin{aligned}
z_1 &= \theta \\
z_2 &= x \cos \theta + y \sin \theta \\
z_3 &= x \sin \theta - y \cos \theta
\end{aligned} \tag{11.23}$$

and the input transformation

$$\begin{aligned}
v &= v_2 + z_3 v_1 \\
\omega &= v_1,
\end{aligned} \tag{11.24}$$

one obtains the (2,3) chained form

$$\begin{aligned}
\dot{z}_1 &= v_1 \\
\dot{z}_2 &= v_2 \\
\dot{z}_3 &= z_2 v_1.
\end{aligned} \tag{11.25}$$

Note that, while z_1 is simply the orientation θ , coordinates z_2 and z_3 represent the position of the unicycle in a moving reference frame whose z_2 axis is aligned with the sagittal axis of the vehicle (see Fig. 11.3).

As for mobile robots with bicycle-like kinematics, consider for example the model (11.19) corresponding to the rear-wheel drive case. Using the change of coordinates

$$\begin{aligned}
z_1 &= x \\
z_2 &= \frac{1}{\ell} \sec^3 \theta \tan \phi \\
z_3 &= \tan \theta \\
z_4 &= y
\end{aligned}$$

and the input transformation

$$v = \frac{v_1}{\cos \theta}$$

$$\omega = -\frac{3}{\ell} v_1 \sec \theta \sin^2 \phi + \frac{1}{\ell} v_2 \cos^3 \theta \cos^2 \phi,$$

the (2,4) chained form is obtained:

$$\begin{aligned}\dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1 \\ \dot{z}_4 &= z_3 v_1.\end{aligned}$$

This transformation is defined everywhere in the configuration space, with the exception of points where $\cos \theta = 0$. The equivalence between the two models is then subject to the condition $\theta \neq \pm k\pi/2$, with $k = 1, 2, \dots$.

11.4 Dynamic Model

The derivation of the dynamic model of a mobile robot is similar to the manipulator case, the main difference being the presence of nonholonomic constraints on the generalized coordinates. An important consequence of nonholonomy is that exact linearization of the dynamic model via feedback is no longer possible. In the following, the Lagrange formulation is used to obtain the dynamic model of an n -dimensional mechanical system subject to $k < n$ kinematic constraints in the form (11.3), and it is shown how this model can be partially linearized via feedback.

As usual, define the Lagrangian \mathcal{L} of the mechanical system as the difference between its kinetic and potential energy:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}(\mathbf{q}) \dot{\mathbf{q}} - \mathcal{U}(\mathbf{q}), \quad (11.26)$$

where $\mathbf{B}(\mathbf{q})$ is the (symmetric and positive definite) inertia matrix of the mechanical system. The Lagrange equations are in this case

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right)^T - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}} \right)^T = \mathbf{S}(\mathbf{q}) \boldsymbol{\tau} + \mathbf{A}(\mathbf{q}) \boldsymbol{\lambda}, \quad (11.27)$$

where $\mathbf{S}(\mathbf{q})$ is an $(n \times m)$ matrix mapping the $m = n - k$ external inputs $\boldsymbol{\tau}$ to generalized forces performing work on \mathbf{q} , $\mathbf{A}(\mathbf{q})$ is the transpose of the $(k \times n)$ matrix characterizing the kinematic constraints (11.3), and $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of *Lagrange multipliers*. The term $\mathbf{A}(\mathbf{q}) \boldsymbol{\lambda}$ represents the vector of reaction forces at the generalized coordinate level. It has been assumed that the number of available inputs matches the number of DOFs (*full actuation*),

that is, in turn, equal to the number n of generalized coordinates minus the number k of constraints.

Using (11.26), (11.27), the *dynamic model* of the constrained mechanical system is expressed as

$$B(q)\ddot{q} + n(q, \dot{q}) = S(q)\tau + A(q)\lambda \quad (11.28)$$

$$A^T(q)\dot{q} = 0, \quad (11.29)$$

where

$$n(q, \dot{q}) = \dot{B}(q)\dot{q} - \frac{1}{2} \left(\frac{\partial}{\partial q} (\dot{q}^T B(q) \dot{q}) \right)^T + \left(\frac{\partial U(q)}{\partial q} \right)^T.$$

Consider now a matrix $G(q)$ whose columns are a basis for the null space of $A^T(q)$, so that $A^T(q)G(q) = 0$. One can replace the constraint given by (11.29) with the kinematic model

$$\dot{q} = G(q)v = \sum_{i=1}^m g_i(q) v_i, \quad (11.30)$$

where $v \in \mathbb{R}^m$ is the vector of *pseudo-velocities*;⁷ for example, in the case of a unicycle the components of this vector are the driving velocity v and the steering velocity ω . Moreover, the Lagrange multipliers in (11.28) can be eliminated premultiplying both sides of the equation by $G^T(q)$. This leads to the *reduced dynamic model*

$$G^T(q) (B(q)\ddot{q} + n(q, \dot{q})) = G^T(q)S(q)\tau, \quad (11.31)$$

a system of m differential equations.

Differentiation of (11.30) with respect to time gives

$$\ddot{q} = \dot{G}(q)v + G(q)\dot{v}.$$

Premultiplying this by $G^T(q)B(q)$ and using the reduced dynamic model (11.31), one obtains

$$M(q)\dot{v} + m(q, v) = G^T(q)S(q)\tau, \quad (11.32)$$

where

$$\begin{aligned} M(q) &= G^T(q)B(q)G(q) \\ m(q, v) &= G^T(q)B(q)\dot{G}(q)v + G^T(q)n(q, G(q)v), \end{aligned}$$

⁷ In the dynamic modeling context, the use of this term emphasizes the difference between v and \dot{q} , that are the actual (generalized) velocities of the mechanical system.

with $\mathbf{M}(\mathbf{q})$ positive definite and

$$\dot{\mathbf{G}}(\mathbf{q})\mathbf{v} = \sum_{i=1}^m \left(v_i \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}}(\mathbf{q}) \right) \mathbf{G}(\mathbf{q})\mathbf{v}.$$

This finally leads to the *state-space reduced model*

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{v} \quad (11.33)$$

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{q})\mathbf{m}(\mathbf{q}, \mathbf{v}) + \mathbf{M}^{-1}(\mathbf{q})\mathbf{G}^T(\mathbf{q})\mathbf{S}(\mathbf{q})\boldsymbol{\tau}, \quad (11.34)$$

that represents in a compact form the kinematic and dynamic models of the constrained system as a set of $n + m$ differential equations.

Suppose now that

$$\det \left(\mathbf{G}^T(\mathbf{q})\mathbf{S}(\mathbf{q}) \right) \neq 0,$$

an assumption on the ‘control availability’ that is satisfied in many cases of interest. It is then possible to perform a *partial linearization via feedback* of (11.33), (11.34) by letting

$$\boldsymbol{\tau} = \left(\mathbf{G}^T(\mathbf{q})\mathbf{S}(\mathbf{q}) \right)^{-1} (\mathbf{M}(\mathbf{q})\mathbf{a} + \mathbf{m}(\mathbf{q}, \mathbf{v})), \quad (11.35)$$

where $\mathbf{a} \in \mathbb{R}^m$ is the *pseudo-acceleration* vector. The resulting system is

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{v} \quad (11.36)$$

$$\dot{\mathbf{v}} = \mathbf{a}. \quad (11.37)$$

Note the structure of this system: the first n equations are the kinematic model, of which the last m — that represent the inclusion of m integrators on the input channels — are a *dynamic extension*. If the system is unconstrained and fully actuated, it is $\mathbf{G}(\mathbf{q}) = \mathbf{S}(\mathbf{q}) = \mathbf{I}_n$; then, the feedback law (11.35) simply reduces to an inverse dynamics control analogous to (8.57), and correspondingly the closed-loop system is equivalent to n decoupled double integrators.

The implementation of the feedback control (11.35) in principle requires the measurement of \mathbf{v} , and this may not be available. However, pseudo-velocities can be computed via the kinematic model as

$$\mathbf{v} = \mathbf{G}^\dagger(\mathbf{q})\dot{\mathbf{q}} = \left(\mathbf{G}^T(\mathbf{q})\mathbf{G}(\mathbf{q}) \right)^{-1} \mathbf{G}^T(\mathbf{q})\dot{\mathbf{q}}, \quad (11.38)$$

provided that \mathbf{q} and $\dot{\mathbf{q}}$ are measured. Note that the left pseudo-inverse of $\mathbf{G}(\mathbf{q})$ has been used here.

By defining the state $\mathbf{x} = (\mathbf{q}, \mathbf{v}) \in \mathbb{R}^{n+m}$ and the input $\mathbf{u} = \mathbf{a} \in \mathbb{R}^m$, system (11.36), (11.37) can be expressed as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} = \begin{bmatrix} \mathbf{G}(\mathbf{q})\mathbf{v} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \end{bmatrix} \mathbf{u}, \quad (11.39)$$

i.e., a nonlinear system with drift also known as the *second-order kinematic model* of the constrained mechanical system. Accordingly, Eq. (11.36) is sometimes called *first-order kinematic model*. In view of the results recalled in Appendix D, the controllability of the latter guarantees the controllability of system (11.39).

Summarizing, in nonholonomic mechanical systems — such as wheeled mobile robots — it is possible to ‘cancel’ the dynamic effects via nonlinear state feedback, provided that the dynamic parameters are exactly known and the complete state of the system (generalized coordinates and velocities \mathbf{q} and $\dot{\mathbf{q}}$) is measured.

Under these assumptions, the control problem can be directly addressed at the (pseudo-)velocity level, i.e., by choosing \mathbf{v} in such a way that the kinematic model

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{v}$$

behaves as desired. From \mathbf{v} , it is possible to derive the actual control inputs at the generalized force level through (11.35). Since $\mathbf{a} = \dot{\mathbf{v}}$ appears in this equation, the pseudo-velocities \mathbf{v} must be differentiable with respect to time.

Example 11.5

For illustration, the above procedure for deriving, reducing and partially linearizing the dynamic model is now applied to the unicycle. Let m be the mass of the unicycle, I its moment of inertia around the vertical axis through its centre, τ_1 the driving force and τ_2 the steering torque. With the kinematic constraint expressed as (11.12), the dynamic model (11.28), (11.29) takes on the form

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \lambda$$

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0.$$

In this case one has

$$\begin{aligned} \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{0} \\ \mathbf{G}(\mathbf{q}) &= \mathbf{S}(\mathbf{q}) \\ \mathbf{G}^T(\mathbf{q})\mathbf{S}(\mathbf{q}) &= \mathbf{I} \\ \mathbf{G}^T(\mathbf{q})\mathbf{B}\dot{\mathbf{G}}(\mathbf{q}) &= \mathbf{0}, \end{aligned}$$

and thus the reduced model in state-space is obtained as

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{M}^{-1}(\mathbf{q})\boldsymbol{\tau} \end{aligned}$$

where

$$\mathbf{M}^{-1}(\mathbf{q}) = \begin{bmatrix} 1/m & 0 \\ 0 & 1/I \end{bmatrix}.$$

By using the input transformation

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{u} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \mathbf{u},$$

the second-order kinematic model is obtained as

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \omega \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2$$

with the state vector $\boldsymbol{\xi} = [x \ y \ \theta \ v \ \omega]^T \in \mathbb{R}^5$.

11.5 Planning

As with manipulators, the problem of planning a trajectory for a mobile robot can be broken down in finding a *path* and defining a *timing law* on the path. However, if the mobile robot is subject to nonholonomic constraints, the first of these two subproblems becomes more difficult than in the case of manipulators. In fact, in addition to meeting the boundary conditions (interpolation of the assigned points and continuity of the desired degree) the path must also satisfy the nonholonomic constraints at all points.

11.5.1 Path and Timing Law

Assume that one wants to plan a trajectory $\mathbf{q}(t)$, for $t \in [t_i, t_f]$, that leads a mobile robot from an initial configuration $\mathbf{q}(t_i) = \mathbf{q}_i$ to a final configuration $\mathbf{q}(t_f) = \mathbf{q}_f$ in the absence of obstacles. The trajectory $\mathbf{q}(t)$ can be broken down into a geometric path $\mathbf{q}(s)$, with $d\mathbf{q}(s)/ds \neq 0$ for any value of s , and a timing law $s = s(t)$, with the parameter s varying between $s(t_i) = s_i$ and $s(t_f) = s_f$ in a monotonic fashion, i.e., with $\dot{s}(t) \geq 0$, for $t \in [t_i, t_f]$. A possible choice for s is the arc length along the path; in this case, it would be $s_i = 0$ and $s_f = L$, where L is the length of the path.

The above space-time separation implies that

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = \frac{d\mathbf{q}}{ds} \dot{s} = \mathbf{q}' \dot{s},$$

where the prime symbol denotes differentiation with respect to s . The generalized velocity vector is then obtained as the product of the vector \mathbf{q}' , which is directed as the tangent to the path in configuration space, by the scalar \dot{s} , that varies its modulus. Note that the vector $[x' \ y']^T \in \mathbb{R}^2$ is directed as

the tangent to the Cartesian path, and has unit norm if s is the cartesian arc length (see Sect. 5.3.1).

Nonholonomic constraints of the form (11.3) can then be rewritten as

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{A}(\mathbf{q})\mathbf{q}'\dot{s} = \mathbf{0}.$$

If $\dot{s}(t) > 0$, for $t \in [t_i, t_f]$, one has

$$\mathbf{A}(\mathbf{q})\mathbf{q}' = \mathbf{0}. \quad (11.40)$$

This condition, that must be verified at all points by the tangent vector on the configuration space path, characterizes the notion of *geometric* path admissibility induced by the kinematic constraint (11.3) that actually affects generalized velocities. Similar to what has been done for trajectories in Sect. 11.2, geometrically admissible paths can be explicitly defined as the solutions of the nonlinear system

$$\mathbf{q}' = \mathbf{G}(\mathbf{q})\tilde{\mathbf{u}}, \quad (11.41)$$

where $\tilde{\mathbf{u}}$ is a vector of *geometric* inputs that are related to the velocity inputs \mathbf{u} by the relationship $\mathbf{u}(t) = \tilde{\mathbf{u}}(s)\dot{s}(t)$. Once the geometric inputs $\tilde{\mathbf{u}}(s)$ are assigned for $s \in [s_i, s_f]$, the path of the robot in configuration space is uniquely determined. The choice of a timing law $s = s(t)$, for $t \in [t_i, t_f]$, will then identify a particular trajectory along this path.

For example, in the case of a mobile robot with unicycle-like kinematics, the pure rolling constraint (11.6) entails the following condition for geometric admissibility of the path:

$$\begin{bmatrix} \sin \theta & -\cos \theta & 0 \end{bmatrix} \mathbf{q}' = x' \sin \theta - y' \cos \theta = 0,$$

that simply expresses the fact that the tangent to the Cartesian path must be aligned with the robot sagittal axis. As a consequence, a path whose tangent is discontinuous (e.g., a broken line) is *not* admissible, unless the unicycle is allowed to stop at discontinuity points by setting $\dot{s} = 0$ for the time necessary to rotate on the spot so as to align with the new tangent.

Geometrically admissible paths for the unicycle are the solutions of the system

$$\begin{aligned} x' &= \tilde{v} \cos \theta \\ y' &= \tilde{v} \sin \theta \\ \theta' &= \tilde{\omega}, \end{aligned} \quad (11.42)$$

where $\tilde{v}, \tilde{\omega}$ are related to v, ω by

$$v(t) = \tilde{v}(s)\dot{s}(t) \quad (11.43)$$

$$\omega(t) = \tilde{\omega}(s)\dot{s}(t). \quad (11.44)$$

11.5.2 Flat Outputs

Many kinematic models of mobile robots, including the unicycle and the bicycle, exhibit a property known as *differential flatness*, that is particularly relevant in planning problems. A nonlinear dynamic system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}$ is *differentially flat* if there exists a set of outputs \mathbf{y} , called *flat* outputs, such that the state \mathbf{x} and the control inputs \mathbf{u} can be expressed algebraically as a function of \mathbf{y} and its time derivatives up to a certain order:

$$\begin{aligned}\mathbf{x} &= \mathbf{x}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots, \mathbf{y}^{(r)}) \\ \mathbf{u} &= \mathbf{u}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots, \mathbf{y}^{(r)}).\end{aligned}$$

As a consequence, once an output trajectory is assigned for \mathbf{y} , the associated trajectory of the state \mathbf{x} and history of control inputs \mathbf{u} are uniquely determined.

In the unicycle and bicycle cases, the Cartesian coordinates are indeed flat outputs. In the following, this property is established for the unicycle. This can be done with reference to either the kinematic model (11.13) or the geometric model (11.42). For simplicity, refer to the latter. Its first two equations imply that, given a Cartesian path $(x(s), y(s))$, the associated state trajectory is $\mathbf{q}(s) = [x(s) \ y(s) \ \theta(s)]^T$ where

$$\theta(s) = \text{Atan2}(y'(s), x'(s)) + k\pi \quad k = 0, 1. \quad (11.45)$$

The two possible choices for k account for the fact that the same Cartesian path may be followed moving forward ($k = 0$) or backward ($k = 1$). If the initial orientation of the robot is assigned, only one of the choices for k is correct. The geometric inputs that drive the robot along the Cartesian path are easily obtained from (11.42), (11.45) as

$$\tilde{v}(s) = \pm \sqrt{(x'(s))^2 + (y'(s))^2} \quad (11.46)$$

$$\tilde{\omega}(s) = \frac{y''(s)x'(s) - x''(s)y'(s)}{(x'(s))^2 + (y'(s))^2}. \quad (11.47)$$

These equations deserve some comments:

- The choice of the sign of $\tilde{v}(s)$ depends on the type of motion (forward or backward).
- If $x'(\bar{s}) = y'(\bar{s}) = 0$ for some $\bar{s} \in [s_i, s_f]$, one has $\tilde{v}(\bar{s}) = 0$. This happens, for example, in correspondence of cusps (motion inversions) in the Cartesian path. In these points, Eq. (11.45) does not define the orientation, that can however be derived by continuity, i.e., as the limit of its right-hand side for $s \rightarrow \bar{s}^-$. Similar arguments can be repeated for the steering velocity $\tilde{\omega}$ given by (11.47).
- The possibility of reconstructing θ and $\tilde{\omega}$ is lost when the Cartesian trajectory degenerates to a point, because in this case it is $x'(s) = y'(s) = 0$ identically.

It is interesting to note that for driftless dynamic systems — like the kinematic models of mobile robots — differential flatness is a necessary and sufficient condition for transformability in the chained form introduced in Sect. 11.3. In particular, it is easy to prove that the flat outputs of a $(2, n)$ chained form are z_1 and z_n , from which it is possible to compute all the other state variables as well as the associated control inputs. For example, in the case of the $(2, 3)$ chained form (11.25) it is

$$z_2 = \frac{\dot{z}_3}{\dot{z}_1} \quad v_1 = \dot{z}_1 \quad v_2 = \frac{\dot{z}_1 \ddot{z}_3 - \ddot{z}_1 \dot{z}_3}{\dot{z}_1^2}.$$

Note that z_2 and v_2 can be actually reconstructed only if $\dot{z}_1(t) \neq 0$, for $t \in [t_i, t_f]$.

11.5.3 Path Planning

Whenever a mobile robot admits a set of flat outputs \mathbf{y} , these may be exploited to solve planning problems efficiently. In fact, one may use any interpolation scheme to plan the path of \mathbf{y} in such a way as to satisfy the appropriate boundary conditions. The evolution of the other configuration variables, together with the associated control inputs, can then be computed algebraically from $\mathbf{y}(s)$. The resulting configuration space path will *automatically* satisfy the nonholonomic constraints (11.40).

In particular, consider the problem of planning a path that leads a unicycle from an initial configuration $\mathbf{q}(s_i) = \mathbf{q}_i = [x_i \ y_i \ \theta_i]^T$ to a final configuration $\mathbf{q}(s_f) = \mathbf{q}_f = [x_f \ y_f \ \theta_f]^T$.

Planning via Cartesian polynomials

As mentioned above, the problem can be solved by interpolating the initial values x_i, y_i and the final values x_f, y_f of the flat outputs x, y . Letting $s_i = 0$ and $s_f = 1$, one may use the following cubic polynomials:

$$\begin{aligned} x(s) &= s^3 x_f - (s-1)^3 x_i + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \\ y(s) &= s^3 y_f - (s-1)^3 y_i + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2, \end{aligned}$$

that automatically satisfy the boundary conditions on x, y . The orientation at each point being related to x', y' by (11.45), it is also necessary to impose the additional boundary conditions

$$\begin{aligned} x'(0) &= k_i \cos \theta_i & x'(1) &= k_f \cos \theta_f \\ y'(0) &= k_i \sin \theta_i & y'(1) &= k_f \sin \theta_f, \end{aligned}$$

where $k_i \neq 0, k_f \neq 0$ are free parameters that must however have the same sign. This condition is necessary to guarantee that the unicycle arrives in \mathbf{q}_f with the same kind of motion (forward or backward) with which it leaves \mathbf{q}_i ;

in fact, since $x(s)$ and $y(s)$ are cubic polynomials, the Cartesian path does not contain motion inversions in general.

For example, by letting $k_i = k_f = k > 0$, one obtains

$$\begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} k \cos \theta_f - 3x_f \\ k \sin \theta_f - 3y_f \end{bmatrix} \quad \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} k \cos \theta_i + 3x_i \\ k \sin \theta_i + 3y_i \end{bmatrix}.$$

The choice of k_i and k_f has a precise influence on the obtained path. In fact, by using (11.46) it is easy to verify that

$$\tilde{v}(0) = k_i \quad \tilde{v}(1) = k_f.$$

The evolution of the robot orientation along the path and the associated geometric inputs can then be computed by using Eqs. (11.45) and (11.46), (11.47), respectively.

Planning via the chained form

Another technique, which can be immediately generalized to other kinematic models of mobile robots (e.g., the bicycle), is planning the path in the chained form coordinates \mathbf{z} . To this end, it is first necessary to compute the initial and final values \mathbf{z}_i and \mathbf{z}_f that correspond to \mathbf{q}_i and \mathbf{q}_f , by using the change of coordinates (11.23). It is then sufficient to interpolate the initial and final values of z_1 and z_3 (the flat outputs) with the appropriate boundary conditions on the remaining variable $z_2 = z'_3/z'_1$.

Again, it is possible to adopt a cubic polynomial to solve the problem. As an alternative, one may use polynomials of different degree for x and y in order to reduce the number of unknown coefficients to be computed. For example, under the assumption $z_{1,i} \neq z_{1,f}$, consider the following interpolation scheme:

$$\begin{aligned} z_1(s) &= z_{1,f}s - (s-1)z_{1,i} \\ z_3(s) &= s^3 z_{3,f} - (s-1)^3 z_{3,i} + \alpha_3 s^2 (s-1) + \beta_3 s (s-1)^2, \end{aligned}$$

with $s \in [0, 1]$. Note that $z'_1(s)$ is constant and equal to $z_{1,f} - z_{1,i} \neq 0$. The unknowns α_3, β_3 must be determined by imposing the boundary conditions on z_2 :

$$\frac{z'_3(0)}{z'_1(0)} = z_{2i} \quad \frac{z'_3(1)}{z'_1(1)} = z_{2f},$$

from which

$$\begin{aligned} \alpha_3 &= z_{2,f}(z_{1,f} - z_{1,i}) - 3z_{3,f} \\ \beta_3 &= z_{2,i}(z_{1,f} - z_{1,i}) + 3z_{3,i}. \end{aligned}$$

This scheme cannot be directly applied when $z_{1,i} = z_{1,f}$, i.e., when $\theta_i = \theta_f$. To handle this singular case, one may introduce a *via point* $\mathbf{q}_v = [x_v \ y_v \ \theta_v]^T$

such that $\theta_v \neq \theta_i$, and solve the original planning problem using two consecutive paths, the first from \mathbf{q}_i to \mathbf{q}_v and the second from \mathbf{q}_v to \mathbf{q}_f . Another possibility, which avoids the introduction of the via point, is to let $z_{1,f} = z_{1,i} + 2\pi$ (i.e., to replace θ_f with $\theta_f + 2\pi$); this obviously corresponds to the same final configuration of the unicycle. With the resulting manoeuvre, the robot will reach its destination while performing a complete rotation of orientation along the path.

Once the path has been planned for the chained form, the path $\mathbf{q}(s)$ in the original coordinates and the associated geometric inputs $\tilde{\mathbf{u}}(s)$ are reconstructed by inverting the change of coordinates (11.23) and of inputs (11.24), respectively.

Planning via parameterized inputs

A conceptually different approach to path planning consists of writing the inputs — rather than the path — in parameterized form, and computing the value of the parameters so as to drive the robot from \mathbf{q}_i to \mathbf{q}_f . Again, it is convenient to work on the chained form, whose equations are easily integrable in closed form under appropriate inputs. For the sake of generality, refer to the $(2, n)$ chained form (11.20), whose geometric version is

$$\begin{aligned} z'_1 &= \tilde{v}_1 \\ z'_2 &= \tilde{v}_2 \\ z'_3 &= z_2 \tilde{v}_1 \\ &\vdots \\ z'_n &= z_{n-1} \tilde{v}_1. \end{aligned}$$

Let the geometric input be chosen as

$$\tilde{v}_1 = \text{sgn}(\Delta) \quad (11.48)$$

$$\tilde{v}_2 = c_0 + c_1 s + \dots + c_{n-2} s^{n-2}, \quad (11.49)$$

with $\Delta = z_{1,f} - z_{1,i}$ and $s \in [s_i, s_f] = [0, |\Delta|]$. Parameters c_0, \dots, c_{n-2} must be chosen so as to give $\mathbf{z}(s_f) = \mathbf{z}_f$. It is possible to verify that such condition is expressed as a linear system of equations

$$\mathbf{D}(\Delta) \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-2} \end{bmatrix} = \mathbf{d}(\mathbf{z}_i, \mathbf{z}_f, \Delta) \quad (11.50)$$

where matrix $\mathbf{D}(\Delta)$ is invertible if $\Delta \neq 0$. For example, in the case of the $(2, 3)$ chained form, one obtains

$$\mathbf{D} = \begin{bmatrix} |\Delta| & \frac{\Delta^2}{2} \\ \text{sgn}(\Delta) \frac{\Delta^2}{2} & \frac{\Delta^3}{6} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} z_{2,f} - z_{2,i} \\ z_{3,f} - z_{3,i} - z_{2,i} \Delta \end{bmatrix}. \quad (11.51)$$

If $z_{1,i} = z_{1,f}$ a singular case is met that can be handled as before.

Both $\mathbf{z}(s)$ and $\tilde{\mathbf{v}}(s)$ must then be converted to $\mathbf{q}(s)$ and $\tilde{\mathbf{u}}(s)$ using the inverse coordinate and input transformations that apply to the specific case.

The above method does not make explicit use of the flat outputs, but relies on the closed-form integrability of the chained form, whose existence, as already mentioned, is equivalent to differential flatness. Note also that in this planning scheme, as in the previous two, parameter s does not represent the arc length on the path.

Other classes of parameterized inputs that can be used in place of (11.48), (11.49) are sinusoidal and piecewise constant functions.

Numerical results

For illustration, some numerical results of the planning methods so far described are now presented. The considered vehicle is a unicycle that must perform various ‘parking’ manoeuvres.

Two typical paths produced by the planner that uses cubic Cartesian polynomials are shown in Fig. 11.5. As already noticed, the unicycle never inverts its motion, that is forward in these two manoeuvres because $k = 5 > 0$. For $k < 0$, the manoeuvres would have been performed in backward motion with different paths.

In Fig. 11.6, the same planner is used to solve a *parallel parking* problem, in which the difference between the initial and the final configuration of the unicycle is a pure displacement in the direction orthogonal to the sagittal axis. Note how the path changes as $k_i = k_f = k$ is changed; in particular, an increase in k leads to elongated ‘take-off’ (from \mathbf{q}_i) and ‘landing’ (on \mathbf{q}_f) phases.

Figure 11.7 refers to the case in which \mathbf{q}_i and \mathbf{q}_f differ only for the value of θ (a pure reorientation); the unicycle leaves the initial position and follows a path that leads back to it with the correct orientation. This behaviour is to be expected, because with this planner a Cartesian path of nonzero length is needed to achieve any kind of reconfiguration.

To allow a comparison, the same planning problems have also been solved with the method based on the use of parameterized inputs in conjunction with the chained form.

Figure 11.8 shows the path obtained with this planner for the same two parking problems of Fig. 11.5. While in the first case the obtained manoeuvre is similar to the one obtained before, in the second the path contains a cusp, corresponding to a motion inversion. In fact, in view of its nature, this planner can only generate paths along which the robot orientation stays between its initial value θ_i and its final value θ_f .

In Fig. 11.9, two different solutions produced by this planner are reported for the parallel parking problem of Fig. 11.6. The singularity due to $\theta_i = \theta_f$ has been solved in two different ways: by adding a via point \mathbf{q}_v , and redefining θ_f as $\theta_f = \theta_i + 2\pi$. Note that in the latter case the path produced by the

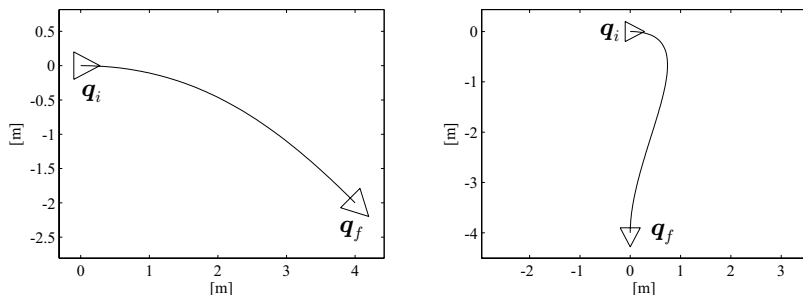


Fig. 11.5. Two parking manoeuvres planned via cubic Cartesian polynomials; in both cases $k = 5$ has been used

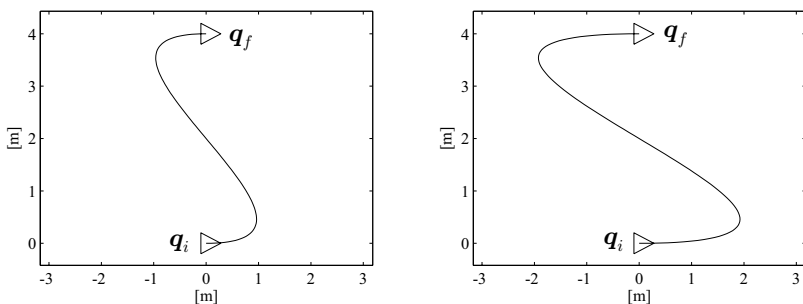


Fig. 11.6. Planning a parallel parking manoeuvre via cubic Cartesian polynomials; *left*: with $k = 10$, *right*: with $k = 20$

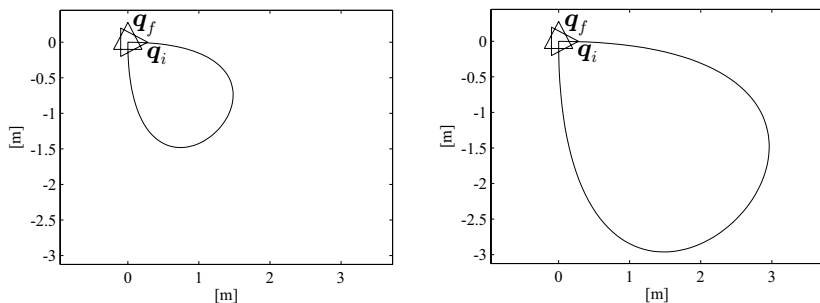


Fig. 11.7. Planning a pure reorientation manoeuvre via cubic Cartesian polynomials; *left*: with $k = 10$, *right*: with $k = 20$

planner leads the robot to the destination with a complete rotation of its orientation.

Finally, the same pure reorientation manoeuvre of Fig. 11.7 has been considered in Fig. 11.10. The path on the left has been obtained as outlined before, i.e., exploiting the transformations of coordinates (11.23) and of in-

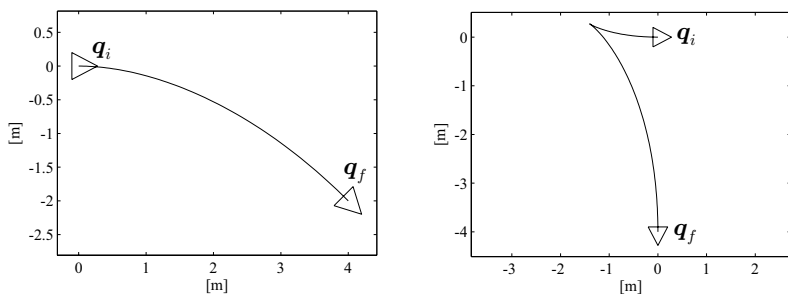


Fig. 11.8. Two parking manoeuvres planned via the chained form

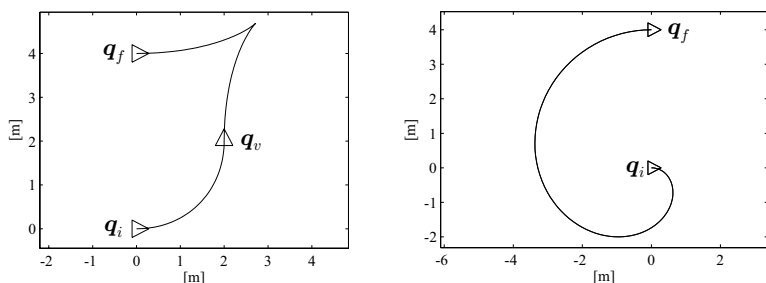


Fig. 11.9. Planning a parallel parking manoeuvre via the chained form; *left*: adding a via point q_v , *right*: letting $\theta_f = \theta_i + 2\pi$

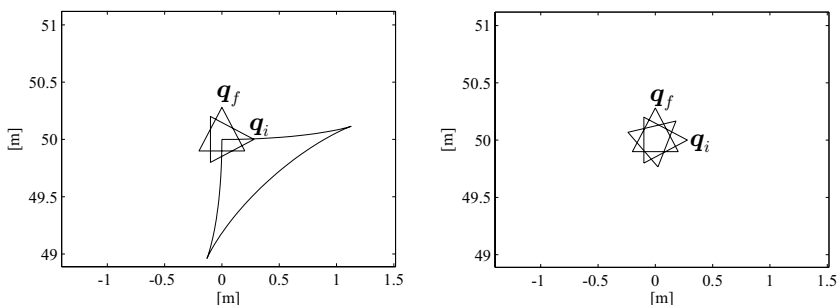


Fig. 11.10. Planning a pure reorientation manoeuvre via the chained form; *left*: with the coordinate transformation (11.23), *right*: with the coordinate transformation (11.52)

puts (11.24) to put the system in chained form, and then using the parameterized inputs (11.48), (11.49). As in the previous case, the required reorientation is realized by a Cartesian path. This is a consequence of the structure of (11.23), for which $\theta_i \neq \theta_f$ implies in general $z_{2,i} \neq z_{2,f}$ and $z_{3,i} \neq z_{3,f}$, even when $x_i = x_f$, $y_i = y_f$.

The manoeuvre on the right of Fig. 11.10, which is a rotation on the spot, was achieved by using a different change of coordinates to put the system in (2,3) chained form. In particular, the following transformation has been used

$$\begin{aligned} z_1 &= \theta - \theta_f \\ z_2 &= (x - x_i) \cos \theta + (y - y_i) \sin \theta \\ z_3 &= (x - x_i) \sin \theta - (y - y_i) \cos \theta, \end{aligned} \quad (11.52)$$

which places the origin of the (z_2, z_3) reference frame in correspondence of the initial Cartesian position of the unicycle. With this choice one has $z_{2,i} = z_{2,f}$ and $z_{3,i} = z_{3,f}$ for a pure reorientation, and thus the manoeuvre is efficiently obtained as a simple rotation.

As a matter of fact, using (11.52) in place of (11.23) is always recommended. In fact, the analysis of (11.51) shows that in general the magnitude of the coefficients of \tilde{v}_2 — and therefore, the length of the obtained path — depends not only on the amount of reconfiguration required for z_2 and z_3 , but also on the value of $z_{2,i}$ itself. The adoption of (11.52), which implies $z_{2,i} = 0$, makes the size of the manoeuvre invariant with respect to the Cartesian position of the unicycle.

11.5.4 Trajectory Planning

Once a path $\mathbf{q}(s)$, $s \in [s_i, s_f]$, has been determined, it is possible to choose a timing law $s = s(t)$ with which the robot should follow it. In this respect, considerations similar to those of Sect. 5.3.1 apply. For example, if the velocity inputs of the unicycle are subject to bounds of the form⁸

$$|v(t)| \leq v_{\max} \quad |\omega(t)| \leq \omega_{\max} \quad \forall t, \quad (11.53)$$

it is necessary to verify whether the velocities along the planned trajectory are admissible. In the negative case, it is possible to slow down the timing law via *uniform scaling*. To this end, it is convenient to rewrite the timing law by replacing t with the normalized time variable $\tau = t/T$, with $T = t_f - t_i$. From (11.43), (11.44) one has

$$v(t) = \tilde{v}(s) \frac{ds}{d\tau} \frac{d\tau}{dt} = \tilde{v}(s) \frac{ds}{d\tau} \frac{1}{T} \quad (11.54)$$

$$\omega(t) = \tilde{\omega}(s) \frac{ds}{d\tau} \frac{d\tau}{dt} = \tilde{\omega}(s) \frac{ds}{d\tau} \frac{1}{T}, \quad (11.55)$$

and therefore it is sufficient to increase T (i.e., the duration of the trajectory) to reduce uniformly v and ω , so as to stay within the given bounds.

⁸ For a differential drive unicycle, the actual bounds affect the wheel angular speeds ω_L and ω_R . Through Eqs. (11.14), these bounds can be mapped to constraints on v and ω (see Problem 11.9).

It is also possible to plan directly a trajectory without separating the geometric path from the timing law. To this end, all the techniques presented before can be used with the time variable t directly in place of the path parameter s . A drawback of this approach is that the duration $t_f - t_i = s_f - s_i$ of the trajectory is fixed, and uniform scaling cannot be used to satisfy bounds on the velocity inputs. In fact, an increase (or decrease) of $t_f - t_i$ would modify the geometric path associated with the planned trajectory.

11.5.5 Optimal Trajectories

The planning techniques so far presented can be used to compute trajectories that lead the robot from an initial configuration \mathbf{q}_i to a final configuration \mathbf{q}_f while complying with the nonholonomic constraints and, possibly, bounds on the velocity inputs. Often, other requirements are added, such as limiting the path curvature, avoiding workspace obstacles or reducing energy consumption. In general, these are integrated in the design procedure as the optimization of a suitable cost criterion along the trajectory. For example, the previous objectives will be respectively formulated as the minimization of the maximum curvature, the maximization of the minimum distance between the robot and the obstacles, or the minimization of the total energy needed by the mobile robot to follow the path.

A simple technique for attacking the optimal planning problem consists of *over-parameterizing* the adopted interpolation scheme, so as to pursue the optimization — typically, via numerical techniques — of the cost criterion by appropriately choosing the redundant parameters. Clearly, the obtained trajectory will be optimal only with respect to the set of trajectories that can be generated by the chosen scheme, and will be a *suboptimal* solution for the original planning problem; this may or may not lead to the fulfilment of the original specifications. For example, the planning scheme based on cubic Cartesian polynomials contains two free parameters (k_i and k_f), that may be chosen so as to maximize the minimum distance along the path between the unicycle and certain obstacles. However, depending on the placement of the obstacles with respect to \mathbf{q}_i and \mathbf{q}_f , a collision-free path (i.e., a path for which the above distance is always positive) may or may not⁹ exist within the chosen family of cubic polynomials.

A more systematic approach to the problem relies on the use of *optimal control* theory. The basic problem considered in this discipline is in fact the determination of a control law that transfers a dynamic system between two assigned states so as to minimize a chosen cost functional along the trajectory. A powerful tool for solving this problem is the *Pontryagin minimum principle* that provides *necessary* conditions for optimality. By exploiting these conditions in conjunction with the analysis of the specific characteristics of the

⁹ The complexity of the problem of planning collision-free motions in the presence of obstacles is such that specific solution techniques are needed; these are presented in Chap. 12.

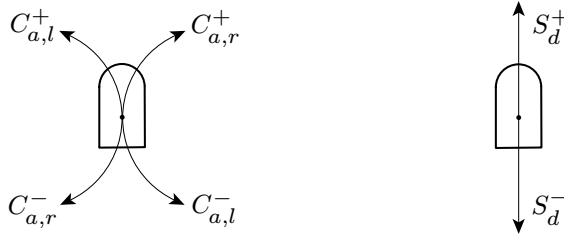


Fig. 11.11. The elementary arcs that constitute the trajectories of the sufficient family for the minimum-time planning problem for the unicycle

considered problem, it is often possible to identify a reduced set of candidate trajectories, also referred to as a *sufficient family*, among which there is certainly the desired optimal solution (if it exists).

In any case, each optimal planning problem must be formulated in the appropriate context. When minimizing curvature or avoiding static obstacles, the timing law part of the trajectory is irrelevant, and the problem can be solved by planning a path for the geometric model (11.41). If the cost criterion depends on the path as well as on the timing law, it is necessary to plan directly on the kinematic model (11.10). A particularly important example of the latter situation is met when minimum-time trajectories are sought in the presence of bounds on the velocity inputs.

Minimum-time trajectories

Consider the problem of transferring the unicycle (11.13) from the initial configuration \mathbf{q}_i to the final configuration \mathbf{q}_f while minimizing the functional

$$J = t_f - t_i = \int_{t_i}^{t_f} dt,$$

under the assumption that the driving and steering velocity inputs v and ω are bounded as in (11.53). By combining the conditions provided by the minimum principle with geometrical arguments, it is possible to determine a sufficient family for the solution to this problem. This family consists of trajectories obtained by concatenating *elementary arcs* of two kinds only:

- arcs of circle of variable length, covered with velocities $v(t) = \pm v_{\max}$ and $\omega(t) = \pm \omega_{\max}$ (the radius of the circle is always v_{\max}/ω_{\max});
- line segments of variable length, covered with velocities $v(t) = \pm v_{\max}$ and $\omega(t) = 0$.

These elementary arcs are shown in Fig. 11.11, where a compact notation is also defined for identifying them. In particular, C_a and S_d indicate an arc of circle of duration a and a line segment of duration d , respectively (in the

particular case $v_{\max} = 1$, a and d are also the lengths of these arcs). The superscript indicates forward (+) or backward (−) motion, while for circular arcs the second subscript indicates a rotation in the clockwise (r) or counterclockwise (l) direction. With this notation, and considering for simplicity the case $v_{\max} = 1$ and $\omega_{\max} = 1$, the trajectories of the sufficient family (also called *Reeds-Shepp curves*) can be classified in the following nine groups:

$$\begin{array}{lll}
 \text{I} & C_a|C_b|C_e & a \geq 0, b \geq 0, e \geq 0, a + b + e \leq \pi \\
 \text{II} & C_a|C_bC_e & 0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2 \\
 \text{III} & C_aC_b|C_e & 0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2 \\
 \text{IV} & C_aC_b|C_bC_e & 0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2 \\
 \text{V} & C_a|C_bC_b|C_e & 0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2 \\
 \text{VI} & C_a|C_{\pi/2}S_eC_{\pi/2}|C_b & 0 \leq a \leq \pi/2, 0 \leq b \leq \pi/2, e \geq 0 \\
 \text{VII} & C_a|C_{\pi/2}S_eC_b & 0 \leq a \leq \pi, 0 \leq b \leq \pi/2, e \geq 0 \\
 \text{VIII} & C_aS_eC_{\pi/2}|C_b & 0 \leq a \leq \pi/2, 0 \leq b \leq \pi, e \geq 0 \\
 \text{IX} & C_aS_eC_b & 0 \leq a \leq \pi/2, 0 \leq b \leq \pi/2, e \geq 0,
 \end{array} \tag{11.56}$$

where the symbol “|” between two elementary arcs indicates the presence of a cusp (motion inversion) on the path. Each group contains trajectories consisting of a *sequence* of no more than five elementary arcs; the duration of circular arcs is bounded by either $\pi/2$ or π , while the duration of line segments depends on the Cartesian distance between \mathbf{q}_i and \mathbf{q}_f . The number of possible sequences produced by each group is finite; they are obtained by instantiating the elementary arcs depending on the direction of motion and of rotation. For example, it is easy to show that group IX generates eight sequence types, each corresponding to a trajectory entirely covered in forward or backward motion:

$$\begin{aligned}
 & C_{a,r}^+ S_e^+ C_{a,r}^+, C_{a,r}^+ S_e^+ C_{a,l}^+, C_{a,l}^+ S_e^+ C_{a,r}^+, C_{a,l}^+ S_e^+ C_{a,l}^+ \\
 & C_{a,r}^- S_e^- C_{a,r}^-, C_{a,r}^- S_e^- C_{a,l}^-, C_{a,l}^- S_e^- C_{a,r}^-, C_{a,l}^- S_e^- C_{a,l}^-.
 \end{aligned}$$

By this kind of argument, it may be verified that the above groups generate a total number of 48 different sequences.

In practice, one may use an exhaustive algorithm to identify the minimum-time trajectory that leads the unicycle from \mathbf{q}_i to \mathbf{q}_f :

- Determine all the trajectories belonging to the sufficient family that join \mathbf{q}_i and \mathbf{q}_f .
- Compute the value of the cost criterion $t_f - t_i$ along these trajectories, and choose the one to which the minimum value is associated.

The first is clearly the most difficult step. Essentially, for each of the aforementioned 48 sequences it is necessary to identify — if it exists — the corresponding trajectory going from \mathbf{q}_i to \mathbf{q}_f , and to compute the duration of the associated elementary arcs. To this end, for each sequence, it is possibly

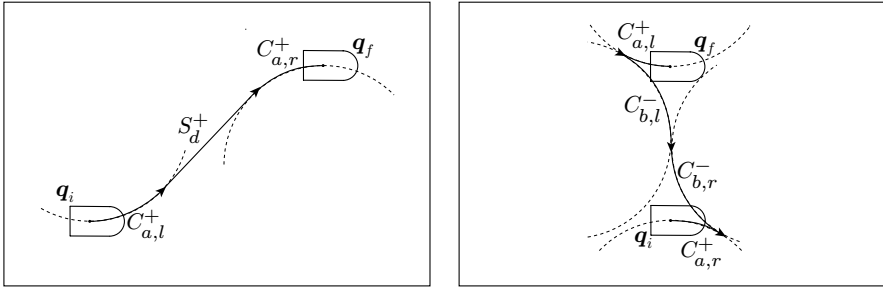


Fig. 11.12. Two examples of minimum-time trajectories for the unicycle

to express and invert in closed form the relationship between the duration of the arcs and the obtained change in configuration. By doing so, the first step can also be completed very quickly.

Figure 11.12 shows two examples of minimum-time trajectories for the unicycle. The first (on the left) belongs to group IX and contains three elementary arcs without inversions of motion. The second (on the right) is a group V trajectory consisting of four arcs of circle, along which there are two motion inversions.

11.6 Motion Control

The motion control problem for wheeled mobile robots is generally formulated with reference to the kinematic model (11.10), i.e., by assuming that the control inputs determine directly the generalized velocities $\dot{\mathbf{q}}$. For example, in the case of the unicycle (11.13) and of the bicycle (11.18) or (11.19) this means that the inputs are the driving and steering velocity inputs v and ω . There are essentially two reasons for taking this simplifying assumption. First, as already seen in Sect. 11.4, under suitable assumptions it is possible to cancel the dynamic effects via state feedback, so that the control problem is actually transferred to the second-order kinematic model, and from the latter to the first-order kinematic model. Second, in the majority of mobile robots it is not possible to command directly the wheel torques, because there are *low-level* control loops that are integrated in the hardware or software architecture. These loops accept as input a reference value for the wheel angular speed, that is reproduced as accurately as possible by standard regulation actions (e.g., PID controllers). In this situation, the actual inputs available for high-level controls are precisely the reference velocities.

In this section, a unicycle-like vehicle is again considered, although some of the presented control schemes may be extended to other kinds of mobile robots. Two basic control problems, illustrated in Fig. 11.13, will be considered for system (11.13):

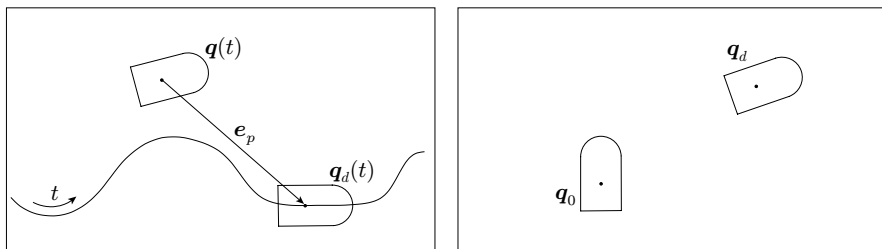


Fig. 11.13. Control problems for a unicycle; *left*: trajectory tracking, *right*: posture regulation

- **Trajectory tracking:** the robot must asymptotically track a desired Cartesian trajectory $(x_d(t), y_d(t))$, starting from an initial configuration $\mathbf{q}_0 = [x_0 \ y_0 \ \theta_0]^T$ that may or may not be ‘matched’ with the trajectory.
- **Posture regulation:** the robot must asymptotically reach a given posture, i.e., a desired configuration \mathbf{q}_d , starting from an initial configuration \mathbf{q}_0 .

From a practical point of view, the most relevant of these problems is certainly the first. This is because, unlike industrial manipulators, mobile robots must invariably operate in unstructured workspaces containing obstacles. Clearly, forcing the robot to move along (or close to) a trajectory planned in advance considerably reduces the risk of collisions. On the other hand, a preliminary planning step is not required when posture regulation is performed, but the Cartesian trajectory along which the robot approaches \mathbf{q}_d cannot be specified by the user.

11.6.1 Trajectory Tracking

For the tracking problem to be soluble, it is necessary that the desired Cartesian trajectory $(x_d(t), y_d(t))$ is admissible for the kinematic model (11.13), i.e., it must satisfy the equations

$$\begin{aligned}\dot{x}_d &= v_d \cos \theta_d \\ \dot{y}_d &= v_d \sin \theta_d \\ \dot{\theta}_d &= \omega_d\end{aligned}\tag{11.57}$$

for some choice of the reference inputs v_d and ω_d . For example, this is certainly true if the trajectory has been produced using one of the planning schemes of the previous section. In any case, as seen in Sect. 11.5.2, since the unicycle coordinates x and y are flat outputs, the orientation along the desired trajectory $(x_d(t), y_d(t))$ can be computed as

$$\theta_d(t) = \text{Atan2}(\dot{y}_d(t), \dot{x}_d(t)) + k\pi \quad k = 0, 1, \tag{11.58}$$

as well as the reference inputs

$$v_d(t) = \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (11.59)$$

$$\omega_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}. \quad (11.60)$$

Note that (11.58) and (11.59), (11.60), correspond respectively to (11.45) and (11.46), (11.47) with $s = t$. In the following, it is assumed that the value of k in (11.58) — and correspondingly, the sign of v_d in (11.59) — has been chosen.

By comparing the desired state $\mathbf{q}_d(t) = [x_d(t) \ y_d(t) \ \theta_d(t)]^T$ with the current measured state $\mathbf{q}(t) = [x(t) \ y(t) \ \theta(t)]^T$ it is possible to compute an error vector that can be fed to the controller. However, rather than using directly the difference between \mathbf{q}_d and \mathbf{q} , it is convenient to define the tracking error as

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}.$$

The positional part of \mathbf{e} is the Cartesian error $\mathbf{e}_p = [x_d - x \ y_d - y]^T$ expressed in a reference frame aligned with the current orientation θ of the robot (see Fig. 11.13). By differentiating \mathbf{e} with respect to time, and using (11.13) and (11.57), one easily finds

$$\begin{aligned} \dot{e}_1 &= v_d \cos e_3 - v + e_2 \omega \\ \dot{e}_2 &= v_d \sin e_3 - e_1 \omega \\ \dot{e}_3 &= \omega_d - \omega. \end{aligned} \quad (11.61)$$

Using the input transformation

$$v = v_d \cos e_3 - u_1 \quad (11.62)$$

$$\omega = \omega_d - u_2, \quad (11.63)$$

which is clearly invertible, the following expression is obtained for the tracking error dynamics:

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (11.64)$$

Note that the first term of this dynamics is linear, while the second and third are nonlinear. Moreover, the first and second terms are in general time-varying, due to the presence of the reference inputs $v_d(t)$ and $\omega_d(t)$.

Control based on approximate linearization

The simplest approach to designing a tracking controller consists of using the approximate linearization of the error dynamics around the reference trajectory, on which clearly $\mathbf{e} = \mathbf{0}$. This approximation, whose accuracy increases as the tracking error \mathbf{e} decreases, is obtained from (11.64) simply setting $\sin e_3 = e_3$ and evaluating the input matrix on the trajectory. The result is

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (11.65)$$

Note that the approximate system is still time-varying. Consider now the linear feedback

$$u_1 = -k_1 e_1 \quad (11.66)$$

$$u_2 = -k_2 e_2 - k_3 e_3 \quad (11.67)$$

that leads to the following closed-loop linearized dynamics:

$$\dot{\mathbf{e}} = \mathbf{A}(t) \mathbf{e} = \begin{bmatrix} -k_1 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & -k_2 & -k_3 \end{bmatrix} \mathbf{e}. \quad (11.68)$$

The characteristic polynomial of matrix \mathbf{A} is

$$p(\lambda) = \lambda(\lambda + k_1)(\lambda + k_3) + \omega_d^2(\lambda + k_3) + v_d k_2(\lambda + k_1).$$

At this point, it is sufficient to let

$$k_1 = k_3 = 2\zeta a \quad k_2 = \frac{a^2 - \omega_d^2}{v_d}, \quad (11.69)$$

with $\zeta \in (0, 1)$ and $a > 0$, to obtain

$$p(\lambda) = (\lambda + 2\zeta a)(\lambda^2 + 2\zeta a\lambda + a^2).$$

The closed-loop linearized error dynamics is then characterized by three constant eigenvalues: one real negative eigenvalue in $-2\zeta a$ and a pair of complex eigenvalues with negative real part, damping coefficient ζ and natural frequency a . However, in view of its time-varying nature, there is no guarantee that system (11.68) is asymptotically stable.

A notable exception is when v_d and ω_d are constant, as in the case of circular or rectilinear trajectories. In fact, the linearized system (11.68) is then time-invariant and therefore asymptotically stable with the choice of gains in (11.69). Hence, by using the control law (11.66), (11.67) with the same gains, the origin of the original error system (11.64) is also asymptotically stable, although this result is not guaranteed to hold globally. For sufficiently

small initial errors, the unicycle will certainly converge to the desired Cartesian trajectory (either circular or rectilinear).

In general, the feedback controller (11.66), (11.67) is linear but also time-varying in view of the expression of k_2 in (11.69). The actual velocity inputs v and ω should be reconstructed from u_1 and u_2 through (11.62), (11.63). In particular, it is easy to verify that v and ω tend to coincide with the reference inputs v_d and ω_d (i.e., they reduce to a pure feedforward action) as the tracking error e vanishes.

Finally, note that k_2 in (11.69) diverges when v_d goes to zero, i.e., when the reference Cartesian trajectory tends to stop. Therefore, the above control scheme can only be used for *persistent* Cartesian trajectories, i.e., trajectories such that $|v_d(t)| \geq \bar{v} > 0, \forall t \geq 0$. This also means that motion inversions (from forward to backward motion, or vice versa) on the reference trajectory are not allowed.

Nonlinear control

Consider again the exact expression (11.64) of the tracking error dynamics, now rewritten for convenience in the ‘mixed’ form:

$$\begin{aligned}\dot{e}_1 &= e_2 \omega + u_1 \\ \dot{e}_2 &= v_d \sin e_3 - e_1 \omega \\ \dot{e}_3 &= u_2,\end{aligned}\tag{11.70}$$

and the following nonlinear version of the control law (11.66), (11.67):

$$u_1 = -k_1(v_d, \omega_d) e_1 \tag{11.71}$$

$$u_2 = -k_2 v_d \frac{\sin e_3}{e_3} e_2 - k_3(v_d, \omega_d) e_3, \tag{11.72}$$

where $k_1(\cdot, \cdot) > 0$ and $k_3(\cdot, \cdot) > 0$ are bounded functions with bounded derivatives, and $k_2 > 0$ is constant. If the reference inputs v_d and ω_d are also bounded with bounded derivatives, and they do not both converge to zero, the tracking error e converges to zero globally, i.e., for any initial condition.

A sketch is now given of the proof of this result. Consider the closed-loop error dynamics

$$\begin{aligned}\dot{e}_1 &= e_2 \omega - k_1(v_d, \omega_d) e_1 \\ \dot{e}_2 &= v_d \sin e_3 - e_1 \omega \\ \dot{e}_3 &= -k_2 v_d \frac{\sin e_3}{e_3} e_2 - k_3(v_d, \omega_d) e_3,\end{aligned}\tag{11.73}$$

and the candidate Lyapunov function

$$V = \frac{k_2}{2} (e_1^2 + e_2^2) + \frac{e_3^2}{2},$$

whose time derivative along the system trajectories

$$\dot{V} = -k_1(v_d, \omega_d)k_2 e_1^2 - k_3(v_d, \omega_d)e_3^2$$

is negative semi-definite. This means that V (which is bounded below) tends to a limit value, and also that the norm of \mathbf{e} is bounded. As system (11.73) is time-varying, it is not possible to use La Salle theorem to gain further insight. However, under the above assumptions, one may verify that \dot{V} is limited, and therefore \dot{V} is uniformly continuous. Hence, Barbalat lemma (see Sect. C.3) may be used to conclude that \dot{V} tends to zero, i.e., that e_1 and e_3 tend to zero. From this and the system equations it is possible to prove that

$$\lim_{t \rightarrow \infty} (v_d^2 + \omega_d^2)e_2^2 = 0,$$

and thus e_2 tends to zero as well, provided that at least one of the reference inputs is persistent.

Again, the actual velocity inputs v and ω must be computed from u_1 and u_2 using (11.62), (11.63). Note that the control law (11.71), (11.72) requires the persistency of the state trajectory $\mathbf{q}(t)$, but not of the Cartesian trajectory. In particular, the reference velocity $v_d(t)$ can converge to zero as long as $\omega_d(t)$ does not, and vice versa. For example, this controller can be used to track a Cartesian trajectory that degenerates to a simple rotation on the spot.

Input/output linearization

A well-known systematic approach to the design of trajectory tracking controllers is based on input/output linearization via feedback (see Sect. 8.5.2). In the case of the unicycle, consider the following outputs:

$$\begin{aligned} y_1 &= x + b \cos \theta \\ y_2 &= y + b \sin \theta, \end{aligned}$$

with $b \neq 0$. They represent the Cartesian coordinates of a point B located along the sagittal axis of the unicycle at a distance $|b|$ from the contact point of the wheel with the ground (see Fig. 11.3). In particular, B is ‘ahead’ of the contact point if b is positive and ‘behind’ if it is negative.

The time derivatives of y_1 and y_2 are

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -b \sin \theta \\ \sin \theta & b \cos \theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{T}(\theta) \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (11.74)$$

Matrix $\mathbf{T}(\theta)$ has determinant b , and is therefore invertible under the assumption that $b \neq 0$. It is then sufficient to use the following input transformation

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{T}^{-1}(\theta) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta/b & \cos \theta/b \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

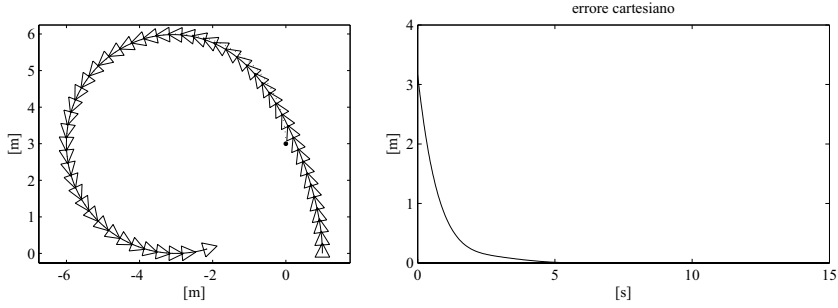


Fig. 11.14. Tracking a circular reference trajectory (*dotted*) with the controller based on approximate linearization; *left*: Cartesian motion of the unicycle, *right*: time evolution of the norm of the Cartesian error e_p

to put the equations of the unicycle in the form

$$\begin{aligned}\dot{y}_1 &= u_1 \\ \dot{y}_2 &= u_2 \\ \dot{\theta} &= \frac{u_2 \cos \theta - u_1 \sin \theta}{b}.\end{aligned}\tag{11.75}$$

An input/output linearization via feedback has therefore been obtained. At this point, a simple linear controller of the form

$$u_1 = \dot{y}_{1d} + k_1(y_{1d} - y_1)\tag{11.76}$$

$$u_2 = \dot{y}_{2d} + k_2(y_{2d} - y_2),\tag{11.77}$$

with $k_1 > 0$, $k_2 > 0$, guarantees exponential convergence to zero of the Cartesian tracking error, with decoupled dynamics on its two components. Note that the orientation, whose evolution is governed by the third equation in (11.75), is not controlled. In fact, this tracking scheme does not use the orientation error; hence, it is based on the output error rather than the state error.

It should be emphasized that the reference Cartesian trajectory for point B can be arbitrary, and in particular the associated path may exhibit isolated points with discontinuous geometric tangent (like in a broken line) without requiring the robot to stop and reorient itself at those points. This is true as long as $b \neq 0$, and therefore such a possibility does not apply to the contact point between the wheel and the ground, whose velocity, as already discussed, cannot have a component in the direction orthogonal to the sagittal axis of the vehicle.

Simulations

An example of application of the trajectory tracking controller (11.66), (11.67) based on linear approximation is shown in Fig. 11.14. The desired trajectory

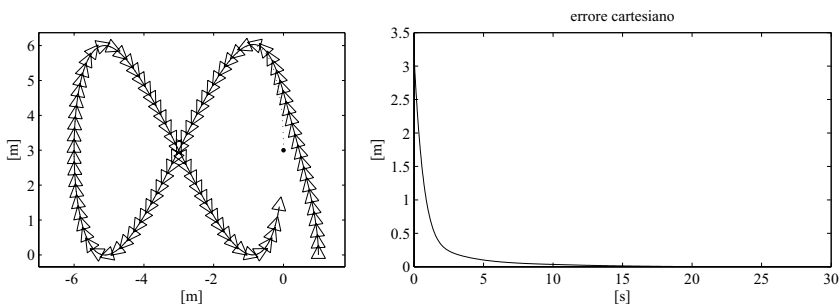


Fig. 11.15. Tracking a figure of eight-shaped reference trajectory (*dotted*) with the nonlinear controller; *left*: Cartesian motion of the unicycle, *right*: time evolution of the norm of the Cartesian error e_p

in this case is the circle of centre (x_c, y_c) and radius R described by the parametric equations

$$\begin{aligned} x_d(t) &= x_c + R \cos(\omega_d t) \\ y_d(t) &= y_c + R \sin(\omega_d t), \end{aligned}$$

with $R = 3$ m and $\omega_d = 1/3$ rad/s. Hence, the reference driving velocity on the circle is constant and equal to $v_d = R\omega_d = 1$ m/s. The controller gains have been chosen according to (11.69) with $\zeta = 0.7$ and $a = 1$. Note the exponential convergence to zero of the Cartesian error.

In the second simulation (Fig. 11.15) the reference trajectory is figure of eight-shaped, with centre in (x_c, y_c) , and is described by the parametric equations

$$\begin{aligned} x_d(t) &= x_c + R_1 \sin(2\omega_d t) \\ y_d(t) &= y_c + R_2 \sin(\omega_d t), \end{aligned}$$

with $R_1 = R_2 = 3$ m and $\omega_d = 1/15$ rad/s. The reference driving velocity $v_d(t)$ in this case varies over time and must be computed using (11.59). The results shown have been obtained with the nonlinear controller (11.71), (11.72), with k_1 and k_3 again given by (11.69) in which $\zeta = 0.7$ and $a = 1$, while k_2 has been set to 1. Also in this case, the error converges to zero very quickly.

The third Cartesian reference trajectory is a square with a side of 4 m, to be traced with constant velocity. This requirement means that the unicycle cannot stop at the vertices in order to reorient itself, and therefore the representative point (x, y) of the unicycle cannot follow the reference trajectory. As a consequence, the tracking scheme based on input/output linearization has been adopted. In particular, two simulations are presented, both obtained using the control law (11.76), (11.77) with $k_1 = k_2 = 2$. In the first, the point B that tracks the reference trajectory is located at a distance $b = 0.75$ m from the contact point between the wheel and the ground. As shown in Fig. 11.16,

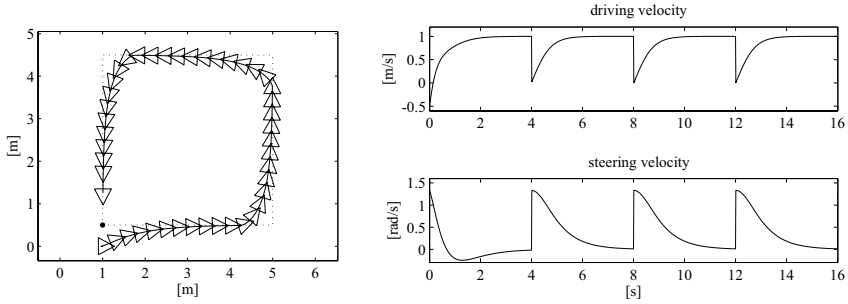


Fig. 11.16. Tracking a square reference trajectory (*dotted*) with the controller based on input/output linearization, with $b = 0.75$; *left*: Cartesian motion of the unicycle, *right*: time evolution of the velocity inputs v and ω

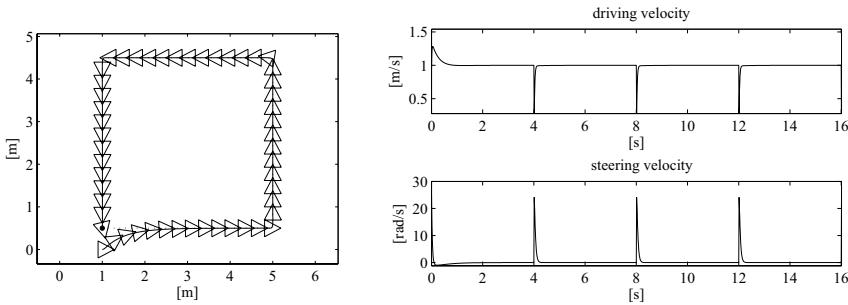


Fig. 11.17. Tracking a square reference trajectory (*dotted*) with the controller based on input/output linearization, with $b = 0.2$; *left*: Cartesian motion of the unicycle, *right*: time evolution of the velocity inputs v and ω

the unicycle actually moves on a ‘smoothed’ trajectory; therefore, an obstacle-free channel around the trajectory must be available to account for the area swept by the unicycle in correspondence of the square vertices.

The second simulation (Fig. 11.17) shows what can happen when b is reduced in order to achieve more accurate tracking for the unicycle representative point; in particular, $b = 0.2$ m was chosen in this case. While it is true that the unicycle tracks the square more closely with respect to the first simulation, the steering velocity is much higher in correspondence of the vertices, and this might be a problem in the presence of saturations on the velocity inputs. This situation is consistent with the fact that matrix \mathbf{T} in (11.74) tends to become singular when b approaches zero.

11.6.2 Regulation

Consider now the problem of designing a **feedback control** law that drives the unicycle (11.13) to a desired configuration \mathbf{q}_d . A reasonable approach could

be to plan first a trajectory that stops in \mathbf{q}_d , and then track it via feedback. However, none of the tracking schemes so far described can be used to this end. In fact, the controller based on approximate linearization and the nonlinear controller both require a persistent state trajectory. The scheme based on input/output linearization can also track non-persistent trajectories, but will drive point B to the destination rather than the representative point of the unicycle. Besides, the final orientation at the destination will not be controlled.

Actually, the difficulty of identifying feedback control laws for tracking non-persistent trajectories is structural. It is in fact possible to prove that, due to the nonholonomy of the system, the unicycle does not admit any *universal* controller, i.e., a controller that can asymptotically stabilize arbitrary state trajectories, either persistent or not. This situation is completely different from the case of manipulators, for which the scheme based on inverse dynamics is an example of universal controller. As a consequence, the posture regulation problem in nonholonomic mobile robots must be addressed using purposely designed control laws.

Cartesian regulation

Consider first the problem of designing a feedback controller for a *partial* regulation task, in which the objective is to drive the unicycle to a given Cartesian position, without specifying the final orientation. This simplified version of the regulation problem is of practical interest. For example, a mobile robot exploring an unknown environment must visit a sequence of Cartesian positions (*view points*) from where it perceives the characteristics of the surrounding area using its on-board sensors. If these are isotropically distributed on the robot (as in the case of a ring of equispaced ultrasonic sensors, a rotating laser range finder, or a panoramic camera), the orientation of the robot at the view point is irrelevant.

Without loss of generality, assume that the desired Cartesian position is the origin; the Cartesian error \mathbf{e}_p is then simply $[-x \ -y]^T$. Consider the following control law

$$v = -k_1(x \cos \theta + y \sin \theta) \quad (11.78)$$

$$\omega = k_2(\text{Atan2}(y, x) - \theta + \pi), \quad (11.79)$$

where $k_1 > 0$, $k_2 > 0$. These two commands have an immediate geometric interpretation: the driving velocity v is proportional to the projection of the Cartesian error \mathbf{e}_p on the sagittal axis of the unicycle, whereas the steering velocity ω is proportional to the difference between the orientation of the unicycle and that of vector \mathbf{e}_p (*pointing error*).

Consider the following ‘Lyapunov-like’ function:

$$V = \frac{1}{2}(x^2 + y^2),$$

that is only positive semi-definite at the origin, because it is zero in all configurations such that $x = y = 0$, independently from the value of the orientation θ . Using the unicycle equations in (11.13) and the control inputs (11.78), (11.79) one obtains

$$\dot{V} = -k_1(x \cos \theta + y \sin \theta)^2,$$

that is negative semi-definite at the origin. This indicates that V , which is bounded below, tends to a limit value, and also that the position error \mathbf{e}_p is bounded in norm. It is easy to verify that \dot{V} is also bounded, and thus V is uniformly continuous. Barbalat lemma¹⁰ implies that \dot{V} tends to zero. Hence

$$\lim_{t \rightarrow \infty} (x \cos \theta + y \sin \theta) = 0,$$

i.e., the projection of the Cartesian error vector \mathbf{e}_p on the sagittal axis of the unicycle tends to vanish. This cannot happen in a point different from the origin, because the steering velocity (11.79) would then force the unicycle to rotate so as to align with \mathbf{e}_p . One may then conclude that the Cartesian error tends to zero for any initial configuration.

Posture regulation

To design a feedback controller that is able to regulate the whole configuration vector (Cartesian position and vehicle orientation) of the unicycle, it is convenient to formulate the problem in polar coordinates. It is again assumed, without loss of generality, that the desired configuration is the origin $\mathbf{q}_d = [0 \ 0 \ 0]^T$.

With reference to Fig. 11.18, let ρ be the distance between the representative point (x, y) of the unicycle and the origin of the Cartesian plane, γ the angle between vector \mathbf{e}_p and the sagittal axis of the vehicle, and δ the angle between the same vector and the x axis. In formulae:

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ \gamma &= \text{Atan2}(y, x) - \theta + \pi \\ \delta &= \gamma + \theta. \end{aligned}$$

In these coordinates, the kinematic model of the unicycle is expressed as

$$\begin{aligned} \dot{\rho} &= -v \cos \gamma \\ \dot{\gamma} &= \frac{\sin \gamma}{\rho} v - \omega \\ \dot{\delta} &= \frac{\sin \gamma}{\rho} v. \end{aligned} \tag{11.80}$$

Note that the input vector field associated with v is singular for $\rho = 0$.

¹⁰ La Salle theorem cannot be used because V is not positive definite at the origin.

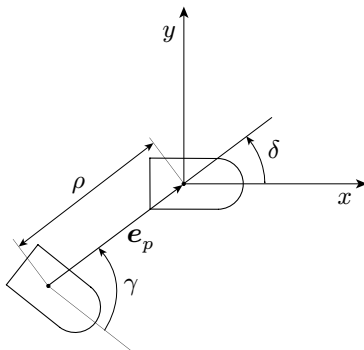


Fig. 11.18. Definition of polar coordinates for the unicycle

Define the feedback control as¹¹

$$v = k_1 \rho \cos \gamma \quad (11.81)$$

$$\omega = k_2 \gamma + k_1 \frac{\sin \gamma \cos \gamma}{\gamma} (\gamma + k_3 \delta), \quad (11.82)$$

with $k_1 > 0$, $k_2 > 0$. The kinematic model (11.80) under the action of the control law (11.81), (11.82) asymptotically converges to the desired configuration $[\rho \ \gamma \ \delta]^T = [0 \ 0 \ 0]^T$.

The proof of this result is based on the following Lyapunov candidate:

$$V = \frac{1}{2} (\rho^2 + \gamma^2 + k_3 \delta^2),$$

whose time derivative along the closed-loop system trajectories

$$\dot{V} = -k_1 \cos^2 \gamma \rho^2 - k_2 \gamma^2$$

is **negative semi-definite**. As a consequence, V tends to a limit value and the system state is bounded. It can also be shown that \ddot{V} is bounded, so that \dot{V} is uniformly continuous. In view of Barbalat lemma, it can be inferred that \dot{V} tends to zero, and likewise do ρ and γ . Further analysis of the closed-loop system leads to concluding that δ converges to zero as well.

Note that angles γ and δ are undefined for $x = y = 0$. They are, however, always well-defined during the motion of the unicycle, and asymptotically tend to the desired zero value.

¹¹ It is easy to verify that the expression (11.81) for v coincides with (11.78), the driving velocity prescribed by the Cartesian regulation scheme, except for the presence of ρ , whose effect is to modulate v according to the distance of the robot from the destination. As for the steering velocity, (11.82) differs from (11.79) for the presence of the second term, that contains the orientation error θ (through δ) in addition to the pointing error.

It should be noted that the control law (11.81), (11.82), once mapped back to the original coordinates, is discontinuous at the origin of the configuration space \mathcal{C} . As a matter of fact, it can be proven that any feedback law that can regulate the posture of the unicycle must be necessarily discontinuous with respect to the state and/or time-varying.¹²

Simulations

To illustrate the characteristics of the above regulation schemes, simulation results are now presented for two parking manoeuvres performed in feedback by a unicycle mobile robot.

Figure 11.19 shows the robot trajectories produced by the Cartesian regulator (11.78), (11.79), with $k = 1$ and $k_2 = 3$, for two different initial configurations. Note that the final orientation of the robot varies with the approach direction, and that the unicycle reaches the destination in forward motion, after inverting its motion at most once (like in the second manoeuvre). It is possible to prove that such behaviour is general with this controller.

The results of the application of the posture regulator (11.81), (11.82) starting from the same initial conditions are shown in Fig. 11.20. The gains have been set to $k_1 = 1$, $k_2 = 2.5$ and $k_3 = 3$. The trajectories obtained are quite similar to the previous ones, but as expected the orientation is driven to zero as well. As before, the final approach to the destination is always in forward motion, with at most one motion inversion in the transient phase.

11.7 Odometric Localization

The implementation of any feedback controller requires the availability of the robot configuration at each time instant. Unlike the case of manipulators, in which the joint encoders provide a direct measurement of the configuration, mobile robots are equipped with incremental encoders that measure the rotation of the wheels, but not directly the position and orientation of the vehicle with respect to a fixed world frame. It is therefore necessary to devise a *localization* procedure that estimates in real time the robot configuration.

Consider a unicycle moving under the action of velocity commands v and ω that are constant within each sampling interval. This assumption, which

¹² This result, which actually applies to all nonholonomic robots, derives from the application of a necessary condition (*Brockett theorem*) for the smooth stabilizability of control systems. In the particular case of a driftless system of the form (11.10), in which there are fewer inputs than states and the input vector fields are linearly independent, such a condition is violated and no control law that is continuous in the state \mathbf{q} can asymptotically stabilize an equilibrium point. Brockett theorem does not apply to time-varying stabilizing controllers that may thus be continuous in \mathbf{q} .

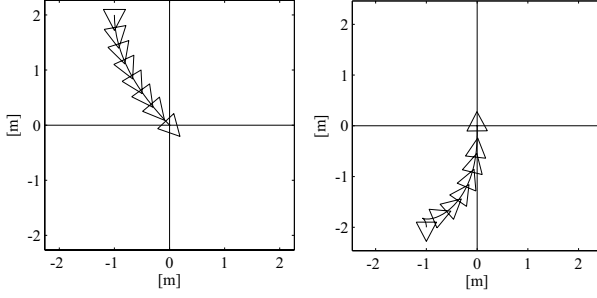


Fig. 11.19. Regulation to the origin of the Cartesian position of the unicycle with the controller (11.78), (11.79), for two different initial configurations

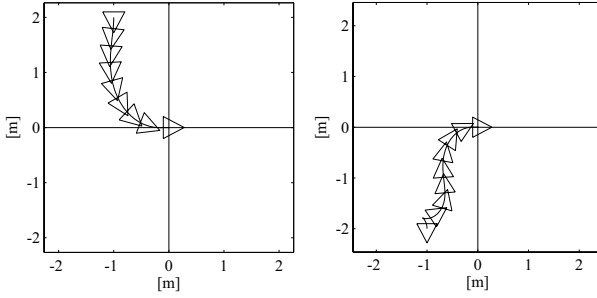


Fig. 11.20. Regulation to the origin of the posture of the unicycle with the controller (11.81), (11.82), for two different initial configurations

is generally satisfied¹³ in digital control implementations, implies that during the interval the robot moves along an arc of circle of radius $R = v_k/\omega_k$, which degenerates to a line segment if $\omega_k = 0$. Assume that the robot configuration $\mathbf{q}(t_k) = \mathbf{q}_k$ at the sampling time t_k is known, together with the value of the velocity inputs v_k and ω_k applied in the interval $[t_k, t_{k+1})$. The value of the configuration variables \mathbf{q}_{k+1} at the sampling time t_{k+1} can then be reconstructed by forward integration of the kinematic model (11.13).

A first possibility is to use the approximate formulae

$$\begin{aligned} x_{k+1} &= x_k + v_k T_s \cos \theta_k \\ y_{k+1} &= y_k + v_k T_s \sin \theta_k \\ \theta_{k+1} &= \theta_k + \omega_k T_s, \end{aligned} \tag{11.83}$$

where $T_s = t_{k+1} - t_k$ is the duration of the sampling interval. These equations, which correspond to the use of the Euler method for numerical integration

¹³ In particular, this is certainly true if the velocity commands computed by the digital controller are converted to control inputs for the robot through a zero-order hold (ZOH).

of (11.13), introduce an error in the computation of x_{k+1} and y_{k+1} , that is performed as if the orientation θ_k remained constant throughout the interval. This error becomes smaller as T_s is decreased. The third formula instead is exact.

If the accuracy of the Euler method proves to be inadequate, one may use the following estimate with the same T_s :

$$\begin{aligned}x_{k+1} &= x_k + v_k T_s \cos \left(\theta_k + \frac{\omega_k T_s}{2} \right) \\y_{k+1} &= y_k + v_k T_s \sin \left(\theta_k + \frac{\omega_k T_s}{2} \right) \\\theta_{k+1} &= \theta_k + \omega_k T_s,\end{aligned}\tag{11.84}$$

corresponding to the adoption of the second-order Runge–Kutta integration method. Note how the first two formulae use the average value of the unicycle orientation in the sampling interval.

To obtain an exact reconstruction of \mathbf{q}_{k+1} under the assumption of constant velocity inputs within the sampling interval one may use simple geometric arguments or exploit the transformability of the unicycle kinematic model in the chained form (11.25). As already seen, this form is easily integrable in closed form, leading to an exact expression for \mathbf{z}_{k+1} . The configuration \mathbf{q}_{k+1} can then be computed by inverting the coordinate and input transformations (11.23) and (11.24). This procedure, which can be generalized to any mobile robot that can be put in chained form, provides the formulae:

$$\begin{aligned}x_{k+1} &= x_k + \frac{v_k}{\omega_k} (\sin \theta_{k+1} - \sin \theta_k) \\y_{k+1} &= y_k - \frac{v_k}{\omega_k} (\cos \theta_{k+1} - \cos \theta_k) \\\theta_{k+1} &= \theta_k + \omega_k T_s.\end{aligned}\tag{11.85}$$

Note that the first two are still defined for $\omega_k = 0$; in this case, they coincide with the corresponding formulae of Euler and Runge–Kutta methods (which are exact over line segments). In the implementation, however, it is necessary to handle this situation with a conditional instruction.

Figure 11.21 allows a comparison among the configurations \mathbf{q}_{k+1} reconstructed via the three aforementioned integration methods. In practice, the difference is obviously much less dramatic, and tends to disappear as the duration T_s of the sampling interval is decreased.

In the previous formulae it has been assumed that the velocity inputs v_k and ω_k applied in the sampling interval are available. In view of the non-ideality of any actuation system, rather than relying on the ‘commanded’ values, it is convenient to reconstruct v_k and ω_k using the robot proprioceptive sensors. First of all, note that

$$v_k T_s = \Delta s \quad \omega_k T_s = \Delta \theta \quad \frac{v_k}{\omega_k} = \frac{\Delta s}{\Delta \theta},\tag{11.86}$$

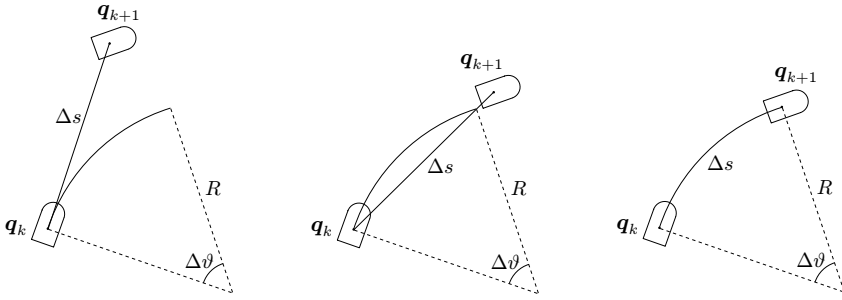


Fig. 11.21. Odometric localization for a unicycle moving along an elementary tract corresponding to an arc of circle; *left*: integration via Euler method, *centre*: integration via Runge–Kutta method, *right*: exact integration

where Δs is the length of the elementary tract travelled by the robot and $\Delta\theta$ is the total variation of orientation along the tract. For example, in the case of a differential drive unicycle, denote by $\Delta\phi_R$ and $\Delta\phi_L$ the rotation of the right and left wheel, respectively, as measured by the incremental encoders during the sampling interval. From (11.14) one easily finds

$$\Delta s = \frac{r}{2} (\Delta\phi_R + \Delta\phi_L) \quad \Delta\theta = \frac{r}{d} (\Delta\phi_R - \Delta\phi_L)$$

that, used in (11.86), allow the implementation of all the previous formulae for the reconstruction of \mathbf{q}_{k+1} .

The forward integration of the kinematic model using the velocity commands reconstructed via the proprioceptive sensors — the encoders of the wheel actuators — is referred to as *odometric localization* or *passive localization* or *dead reckoning* — the latter is a term of uncertain etymology used in marine navigation. This method, relying on the iterated use of the previous formulae starting from an estimate of the initial configuration, provides an estimate whose accuracy cannot be better than that of \mathbf{q}_0 . In any case, odometric localization — independently from the adopted integration method — is subject in practice to an error that grows over time (*drift*) and quickly becomes significant over sufficiently long paths. This error is the result of several causes, that include wheel slippage, inaccuracy in the calibration of kinematic parameters (e.g., the radius of the wheels), as well as the numerical error introduced by the integration method, if Euler or Runge–Kutta methods are used. It should also be noted that, once an odometric localization technique has been chosen, its performance also depends on the specific kinematic arrangement of the robot; for example, differential drive is usually better than synchro drive in this respect.

A more robust solution is represented by *active localization* methods. For example, this kind of approach can be adopted when the robot is equipped with proximity exteroceptive sensors (like a laser range finder) and knows a

map of the workspace, either given in advance or built by the robot itself during the motion. It is then possible to correct the estimate provided by the passive localization methods by comparing the expected measures of the exteroceptive sensors with the actual readings. These techniques, which make use of tools from *Bayesian estimation theory* such as the *Extended Kalman Filter* or the *Particle Filter*, provide greater accuracy than pure odometric localization and are therefore essential in navigation tasks over long paths.

Bibliography

The literature on mobile robots is very rich and includes some comprehensive treatises, among which one of the most recent is [210].

Many books deal with the realization of mobile robots, with particular emphasis on electro-mechanical design and sensor equipment, e.g., [106, 72, 21]. For the modelling of systems subject to nonholonomic constraints, see [164]. A general classification of wheeled mobile robots based on the number, placement and type of wheels is proposed in [18], where the derivation of the kinematic and dynamic models is also presented.

Conditions for transforming driftless systems in chained form are discussed in detail in [159], while a general reference on flat outputs is [80]. The presented schemes for Cartesian trajectory tracking based on linear and nonlinear control are taken from [34], while the method for posture regulation based on polar coordinates was proposed in [2]. Complete (input/state) linearization of the unicycle kinematic model can be obtained using a dynamic feedback law, as shown for example in [174]. The non-existence of a universal controller for nonholonomic robots is proven in [135].

A detailed extension of some of the planning and control techniques described in this chapter to the case of bicycle-like kinematics is given in [58]. The design of time-varying and/or discontinuous feedback control laws for posture regulation in mobile robots was addressed in many works, including [193] and [153].

The reader interested in sensor-based robot localization and map building can refer, e.g., to [231].

Problems

11.1. Consider the mobile robot obtained by connecting N trailers to a rear-wheel drive tricycle. Each trailer is a rigid body with an axle carrying two fixed wheels, that can be assimilated to a single wheel located at the midpoint of the axle, and is hinged to the midpoint of the preceding axle through a revolute joint. Find a set of generalized coordinates for the robot.

11.2. Consider an omnidirectional mobile robot having three Mecanum wheels placed at the vertices of an equilateral triangle, each oriented in the direction orthogonal to the bisectrix of its angle. Let q_1 and q_2 be the Cartesian coordinates of the centre of the robot, q_3 the vehicle orientation, while q_4, q_5 , and q_6 represent the angle of rotation of each wheel around its axis. Also, denote by r the radius of the wheels and by ℓ the distance between the centre of the robot and the centre of each wheel. This mechanical system is subject to the following Pfaffian constraints:

$$\mathbf{A}_1^T(\mathbf{q}) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \mathbf{A}_2^T \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = 0,$$

where

$$\mathbf{A}_1(q) = \begin{bmatrix} \frac{\sqrt{3}}{2} \cos q_3 - \frac{1}{2} \sin q_3 & \sin q_3 & -\frac{1}{2} \sin q_3 - \frac{\sqrt{3}}{2} \cos q_3 \\ \frac{1}{2} \cos q_3 + \frac{\sqrt{3}}{2} \sin q_3 & -\cos q_3 & \frac{1}{2} \cos q_3 - \frac{\sqrt{3}}{2} \sin q_3 \\ \ell & \ell & \ell \end{bmatrix}$$

and $\mathbf{A}_2 = r \mathbf{I}_3$. Show that this set of constraints is partially integrable, and that in particular the orientation of the vehicle is a linear function of the wheel rotation angles. [*Hint*: add the kinematic constraints side-by-side.]

11.3. Show that, for a single kinematic constraint in the form (11.7), the integrability condition expressed by (11.9) coincides with the involutivity of the distribution $\Delta = \text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_{n-1}\}$ associated with the input vector fields of the corresponding kinematic model.

11.4. Using the controllability condition (11.11), show that a set of Pfaffian constraints that does not depend on the generalized coordinates is always integrable.

11.5. With reference to the kinematic model (11.13) of the unicycle, consider the following sequence of velocity inputs:

$$\begin{aligned} v(t) &= 1 & \omega(t) &= 0, & t &\in [0, \varepsilon) \\ v(t) &= 0 & \omega(t) &= 1 & t &\in [\varepsilon, 2\varepsilon) \\ v(t) &= -1 & \omega(t) &= 0, & t &\in [2\varepsilon, 3\varepsilon) \\ v(t) &= 0 & \omega &= -1 & t &\in [3\varepsilon, 4\varepsilon). \end{aligned}$$

By forward integration of the model equations, show that when ε is infinitesimal the displacement of the unicycle in configuration space is aligned with the Lie bracket of the input vector fields.

11.6. Prove the relationships (11.14) that allow the transformation of the velocity inputs of a differential drive vehicle to those of the equivalent unicycle.

[*Hint:* for the velocity of the midpoint of the segment joining the two wheels, it is sufficient to differentiate with respect to time the Cartesian coordinates of such a point expressed as a function of the coordinates of the wheel centres. As for ω , use the formula (B.4) for the composition of the velocities of a rigid body.]

11.7. For a generic configuration of a bicycle mobile robot, determine the Cartesian position of the instantaneous centre of rotation, and derive the expression of the angular velocity of the body as a function of the robot configuration \mathbf{q} and of the modulus of the velocity of rear wheel centre. In the particular case of the rear-wheel drive bicycle, show that such expression is consistent with the evolution of θ predicted by the kinematic model (11.19). Moreover, compute the velocity v_P of a generic point P on the robot chassis.

11.8. Derive the kinematic model of the tricycle robot towing N trailers considered in Problem 11.1. Denote by ℓ the distance between the front wheel and rear wheel axle of the tricycle, and by ℓ_i the joint-to-joint length of the i -th trailer.

11.9. Consider a differential drive robot whose angular speed inputs — one for the right wheel and one for the left wheel — are subject to the following bounds:

$$|\omega_R(t)| \leq \omega_{RL} \quad |\omega_L(t)| \leq \omega_{RL} \quad \forall t,$$

that correspond to a square admissible region in the ω_R, ω_L plane. Derive the expression of the resulting constraints for the driving and steering velocity v and ω of the equivalent unicycle model. In particular, show that the admissible region in the v, ω plane is a rhombus.

11.10. Compute the expression of matrix $\mathbf{D}(\Delta)$ and vector $\mathbf{d}(\mathbf{z}_i, \mathbf{z}_f, \Delta)$ in (11.50) for the case of the (2,4) chained form.

11.11. Modify the path planning scheme based on parameterized inputs so as to obtain $s_f = 1$.

11.12. Show that the path planning schemes based on polynomials of different degree and on parameterized inputs give exactly the same result in the case of the (2,3) chained form.

11.13. Implement in a computer program the path planning method for a unicycle based on cubic Cartesian polynomials. Use the program to plan a path leading the robot from the configuration $\mathbf{q}_i = [0 \ 0 \ 0]^T$ [m,m,rad] to the configuration $\mathbf{q}_f = [2 \ 1 \ \pi/2]^T$ [m,m,rad]. Then, determine a timing law over the path so as to satisfy the following velocity bounds:

$$|v(t)| \leq 1 \text{ m/s} \quad |\omega(t)| \leq 1 \text{ rad/s} \quad \forall t.$$

11.14. Formulate the trajectory tracking problem for a (2,3) chained form and derive the corresponding error dynamics. Derive a feedback controller using the linear approximation around the reference trajectory.

11.15. Consider the kinematic model (11.19) of the rear-wheel drive bicycle. In analogy to the case of the unicycle, identify two outputs y_1, y_2 for which it is possible to perform a static input/output linearization. [*Hint*: consider a point P on the line passing through the centre of the front wheel and oriented as the wheel itself.]

11.16. Implement in a computer program the Cartesian regulator (11.78), (11.79), including a modification to allow the unicycle to reach the origin either in forward or in backward motion. [*Hint*: modify (11.79) so as to force the robot to orient itself as vector \mathbf{e}_p or vector $-\mathbf{e}_p$, depending on which choice is the most convenient.]

11.17. Prove formulae (11.85) for the exact odometric localization of a unicycle under velocity inputs that are constant in the sampling interval.

11.18. Implement in a computer program the unicycle posture regulator based on polar coordinates, with the state feedback computed through the Runge–Kutta odometric localization method.