12 *Nonholonomic and Underactuated Systems*

EVERY ROBOT system is subject to a variety of motion constraints, but not all of these can be expressed as configuration constraints. A familiar example of such a system is a car. At low speeds, the rear wheels of the car roll freely in the direction they are pointing, but they prevent slipping motion in the perpendicular direction. This constraint implies that the car cannot translate directly to the side. We know by experience, however, that this velocity constraint does not imply a constraint on configurations; the car can reach any position and orientation in the obstacle-free plane. In fact, the prevented sideways translation can be approximated by parallel-parking maneuvers.

This no-slip constraint is a *nonholonomic constraint,* a constraint on the velocity. In addition to rolling without slipping, conservation of angular momentum is a common source of nonholonomic constraints in mechanical systems.

If, instead of viewing the car as a system subject to a motion constraint, we considered the fact that there are only two inputs (speed and steering angle) to control the car's three degrees of freedom, we might call the system *underactuated*. Underactuated systems have fewer controls than degrees of freedom. For second-order mechanical systems, such as those described in the previous chapter, underactuation implies equality constraints on the possible accelerations of the system.

In this section we study motion planning for systems that are underactuated or subject to motion constraints. Our first task is to determine if the constraints actually limit the reachable states of the robot system. This is a controllability question. The next problem is to construct algorithms that find motion plans that satisfy the motion

constraints. A last problem, not addressed in this chapter, is feedback stabilization of the motion plans during execution.

We begin in section 12.1 by providing some background information on vector fields and their Lie (pronounced "lee") algebras. Section 12.2 defines the class of control systems we will consider. Section 12.3 describes different controllability notions and tests for these nonlinear systems. Section 12.4 specializes the discussion to second-order mechanical systems. Finally, section 12.5 describes a number of methods for motion planning for nonholonomic and underactuated systems.

12.1 Preliminaries

First we must decide how generally to define the state spaces of the robotic systems we will consider. For example, we could treat a very general case, allowing the state space of the system to be any smooth manifold. This would allow us to study, e.g., the motion of a spherical pendulum. The configuration space of this system is the sphere S^2. Or we could limit our treatment to systems evolving on Lie groups, particularly matrix Lie groups. This would allow us to model the orientation of a satellite as a point in $SO(3)$.

In this chapter, we restrict our attention even further to systems evolving on vector spaces $\mathcal{M} = \mathbb{R}^n$. This allows us to get to the main results as quickly as possible. Also, any n-dimensional manifold is locally "similar" (diffeomorphic) to $\mathbb{R}^n$, so, equipped with a proper set of local coordinates, any n-dimensional manifold can be treated locally as $\mathbb{R}^n$. By making this simplification, we require the use of a local coordinate system in our computations, and we may lose information about the global structure of the space. As examples, the true configuration space of a 2R robot arm is the torus $T^2 = S^1 \times S^1$, which is doughnut-shaped while $\mathbb{R}^2$ is not; and a global representation of the orientation of a satellite is $SO(3)$, which is different from a local representation using three Euler angles ($\mathbb{R}^3$). See figure 12.1 for another example.

Although we focus on vector state spaces, most of the ideas in this chapter generalize immediately to general manifolds.

In this chapter, $q \in \mathcal{Q}$ denotes the configuration of the system and $x \in \mathcal{M}$ denotes the state of the system. If the system is kinematic, then the state is simply the configuration ($\mathcal{M} = \mathcal{Q}$), and the controls are velocities. If the system is a second-order mechanical system, then x includes both configurations q and velocities $\dot{q}$, and the controls are forces (accelerations). The dimension of the configuration space $\mathcal{Q}$ is $n_{\mathcal{Q}}$, and the dimension of the state space $\mathcal{M}$ is n.

We will carry two examples throughout the chapter: a unicycle, a kinematic system; and a model of a planar spacecraft, a second-order mechanical system. We will treat

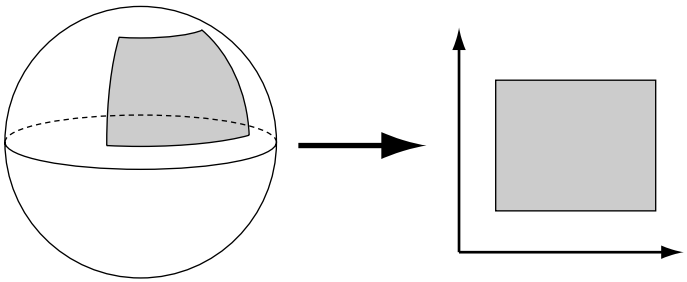

Figure 12.1 Latitude and longitude coordinates allow us to treat a patch of the sphere S^2 as a section of the plane $\mathbb{R}^2$.

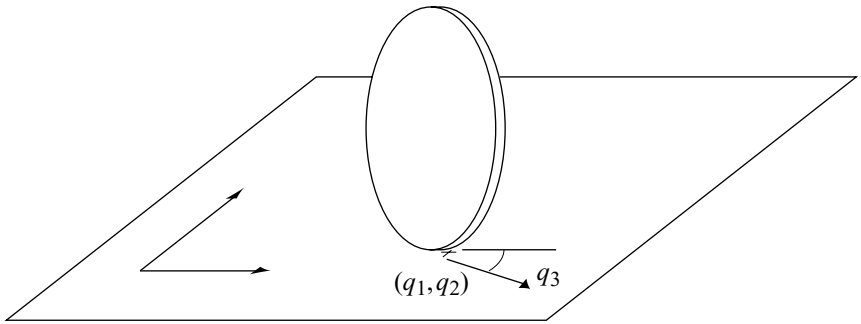

Figure 12.2 The unicycle system. The position of the point of contact is given by (q_1, q_2), and the heading direction is given by q_3.

all systems uniformly, as systems with state x on a state space $\mathcal{M}$. Only in section 12.4 and subsection 12.5.7 will we specialize our study to second-order mechanical systems such as the spacecraft model.

EXAMPLE 12.1.1 Unicycle example. *The unicycle is a wheel that rolls upright on a horizontal plane (figure 12.2). The configuration of the wheel is* $q = [q_1, q_2, q_3]^T$, *describing the contact point of the wheel on the plane* (q_1, q_2) *and the steering angle* q_3 *of the wheel. (We could also include the rolling angle of the wheel, i.e., the location of the air nozzle on the tire, in the description of the configuration, but we will ignore this for now.) The system is kinematic, so* $x = [x_1, x_2, x_3]^T = q$, $\mathcal{M} = \mathcal{Q} = \mathbb{R}^3$, *and* $n_{\mathcal{Q}} = n = 3$. *(Since we are dealing with local coordinates, we are ignoring the fact that the global structure of the space is* $\mathbb{R}^2 \times S^1$. *This will not affect the equations of motion, but requires the use of* $\mathrm{mod}2\pi$ *arithmetic on the third coordinate.) The*

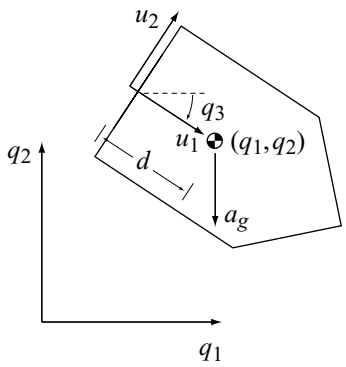

Figure 12.3 The planar body with thrusters (PBWT).

controls are the rolling speed of the wheel and the rate of change of the steering angle. Sideways translation of the wheel is prevented by the no-slip constraint imposed by the wheel. This example is sometimes known as the rolling penny, or the pizza cutter, and it is similar to a model for a car.

EXAMPLE 12.1.2 Planar body with thrusters (PBWT) example. *The body moves in a frictionless, inviscid plane by means of two thrusters fixed to the body (figure 12.3). The mass and inertia of the body (about the center of mass) are unit. The line of action of the thrust u_1 is through the center of mass, and the line of action of the thrust u_2 is perpendicular and a distance d from the center of mass. The configuration is $q = [q_1, q_2, q_3]^T$, describing the location of the center of mass (q_1, q_2) and the angle q_3 of the line of action of the first thruster relative to the world q_1-axis. The system is second-order, so $x = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^T$, $\mathcal{M} = \mathbb{R}^6$, $n_Q = 3$, and $n = 2n_Q = 6$. Gravitational acceleration a_g acts in the $-q_2$-direction, and a_g may be zero.*

The rest of section 12.1 introduces concepts from differential geometry that will be useful in understanding underactuated systems. For the unicycle, e.g., we will see that its instantaneous motions can be described in terms of two "vector fields" associated with the controls to drive and steer the unicycle. Linear combinations of these two vector fields define a "distribution" describing all possible instantaneous motions of the unicycle. The "integral manifold" describes all the states the system can reach by following vector fields in the distribution. We use the "Lie bracket" to show that two vector fields in the distribution can generate a parallel-parking motion for the unicycle, effectively giving it a sideways motion, meaning that the integral

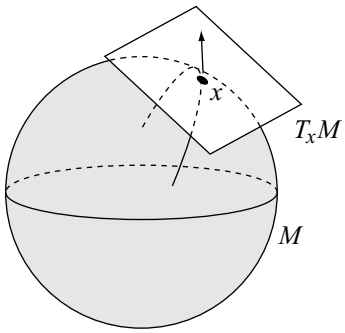

Figure 12.4 A curve on the sphere $\mathcal{M}$, a tangent vector to the curve at x, and the tangent space $T_x\mathcal{M}$ it lives in.

manifold is the entire configuration space—the velocity constraint does not reduce the reachable space.

Section 12.2 describes how a robot system can be expressed as a system of vector fields and controls, and section 12.3 uses the concepts developed in this section to study the set of states reachable by the controls.

12.1.1 Tangent Spaces and Vector Fields

Let $x : \mathbb{R} \to \mathcal{M}$ be a smooth curve on $\mathcal{M}$ parameterized by s. Then dx/ds, evaluated at $x_0 = x(s_0)$, is tangent to the curve at x_0. Call this vector V. The vector V is a *tangent vector* that is tangent to $\mathcal{M}$ at x_0. The tangent vector V lives in $T_{x_0}\mathcal{M}$, the *tangent space* of $\mathcal{M}$ at x_0. This space is an n-dimensional vector space $\mathbb{R}^n$ consisting of the tangents of all possible curves passing through x_0 (figure 12.4). The tangent spaces at different points of $\mathcal{M}$ are different spaces.

The *tangent bundle* of $\mathcal{M}$, written $T\mathcal{M}$, is the $2n$-dimensional manifold that is the union of tangent spaces at all points in $\mathcal{M}$,

$$T\mathcal{M} = \bigcup_{x \in \mathcal{M}} T_x\mathcal{M}.$$

For the systems we study, $T\mathcal{M} = \mathcal{M} \times \mathbb{R}^n = \mathbb{R}^{2n}$.[1]

A smooth *vector field* $g : \mathcal{M} \to T\mathcal{M}$ is a smooth map from points $x \in \mathcal{M}$ to tangent vectors $g(x) \in T_x\mathcal{M}$. It is possible to define C^k vector fields, but we will

1. We note that if $\mathcal{M}$ is a more general manifold, and it is *parallelizable* (e.g., a Lie group), then $T\mathcal{M} = \mathcal{M} \times \mathbb{R}^n$. The reader should be careful not to generalize improperly, however. For example, $TS^2 \neq S^2 \times \mathbb{R}^2$, for reasons beyond the scope of this chapter. For an intuitive discussion of this issue, see [372].

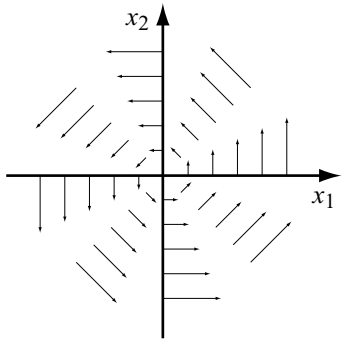

Figure 12.5 The vector field $\frac{1}{2}[-x_2, x_1]^T$.

assume that all vector fields are infinitely differentiable. (For example, the vector field $g(x) = [x_1^2, \sin x_3, x_1 x_2]^T$ is infinitely differentiable, but $[|x_1|, x_2, x_3]^T$ is only C^0.) A picture of the vector field $\frac{1}{2}[-x_2, x_1]^T$ on $\mathbb{R}^2$ is shown in figure 12.5. Tangent vectors are written as column vectors.

In the case of a kinematic system, $\mathcal{M}$ is the configuration space $\mathcal{Q}$, and $T_{x_0}\mathcal{M} = T_{q_0}\mathcal{Q}$ is the set of all possible velocities of the system at $x_0 = q_0$. In the case of a second-order system, $\mathcal{M}$ is the state space $T\mathcal{Q}$, and $T_{x_0}\mathcal{M} = T_{x_0}T\mathcal{Q}$ is the set of all possible velocities and accelerations of the system at $x_0 = [q_0^T, \dot{q}_0^T]^T$. In this case, however, the state x_0 already specifies the velocity portion $\dot{q}_0$ of the tangent vector $[\dot{q}_0^T, \ddot{q}_0^T]^T$. This implies *drift* in second-order systems, as shown in the PBWT example below.

EXAMPLE 12.1.3 Unicycle (cont.) *A tangent vector for the unicycle is given by* $\dot{x} = [\dot{x}_1, \dot{x}_2, \dot{x}_3]^T = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$. *The unicycle is capable of rolling forward and backward and spinning in place. These two vector fields can be written* $g_1^{\text{uni}}(x) = [\cos x_3, \sin x_3, 0]^T$, *rolling forward at unit speed, and* $g_2^{\text{uni}}(x) = [0, 0, 1]^T$, *spinning counterclockwise at unit speed. The vector fields can also be written as* $g_1^{\text{uni}}(x) = (\cos x_3)\partial/\partial x_1 + (\sin x_3)\partial/\partial x_2$ *and* $g_2^{\text{uni}}(x) = \partial/\partial x_3$, *where* $\partial/\partial x_1$, $\partial/\partial x_2$, *and* $\partial/\partial x_3$ *are the canonical unit basis vectors of the tangent space, i.e., unit speed tangent vectors along the coordinates (see figure 12.6).*

EXAMPLE 12.1.4 PBWT (cont.) *A tangent vector for the PBWT is given by* $\dot{x} = [\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, \dot{x}_5, \dot{x}_6]^T = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \ddot{q}_1, \ddot{q}_2, \ddot{q}_3]^T$. *For this system, we can define three vector fields: the* drift *vector field* $g_0^{\text{pbwt}}(x)$ *corresponding to the motion of the body when no thrusters are activated, and the* control *vector fields*

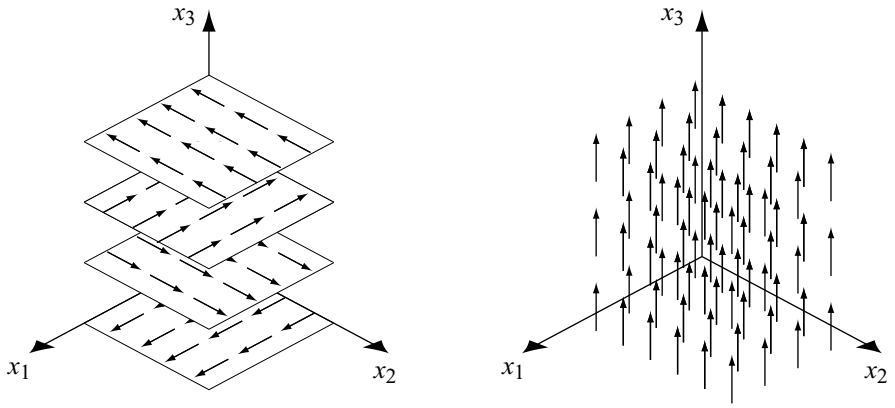

Figure 12.6 The vector fields $g_1^{\text{uni}} = [\cos x_3, \sin x_3, 0]^T$ (shown in constant x_3 layers) and $g_2^{\text{uni}} = [0, 0, 1]^T$.

$g_1^{\text{pbwt}}(x)$ *and* $g_2^{\text{pbwt}}(x)$ *corresponding to the acceleration when thrusters 1 and 2 are fired with unit thrust, respectively. Verify that* $g_0^{\text{pbwt}}(x) = [x_4, x_5, x_6, 0, a_g, 0]^T$, $g_1^{\text{pbwt}}(x) = [0, 0, 0, \cos x_3, \sin x_3, 0]^T$, *and* $g_2^{\text{pbwt}}(x) = [0, 0, 0, -\sin x_3, \cos x_3, -d]^T$, *and write these vector fields in the canonical basis* $\{\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3, \partial/\partial x_4, \partial/\partial x_5, \partial/\partial x_6\}$. *Notice if thruster 1 is fired with thrust* u_1, *the system follows the vector field* $g_0^{\text{pbwt}}(x) + u_1 g_1^{\text{pbwt}}(x)$.

Let ϕ^g denote the *flow* of the vector field g, where $\phi_t^g(x)$ gives the system state after following the flow ϕ^g from x for a time t. The flow satisfies the equation

$$\frac{d}{dt}\phi_t^g(x) = g\left(\phi_t^g(x)\right).$$

The vector field is *complete* if its flow is defined for all x and t.

The curve $\{\phi_t^g(x) \mid t \in \mathbb{R}\}$ is the *integral curve* of g containing x. The integral curve describes the set of reachable points of $\mathcal{M}$ from x by following the vector field forward and backward in time (figure 12.7). This notion can be generalized to the *integral manifold* of a set of vector fields $\mathcal{G}$, a topic for subsection 12.1.3.

12.1.2 Distributions and Constraints

Let $\mathcal{G}$ be a set of vector fields, and let span($\mathcal{G}$) be the linear span of vector fields in $\mathcal{G}$, given by all linear combinations of vector fields in $\mathcal{G}$. At each point $x \in \mathcal{M}$, these vector fields span a linear subspace of $T_x\mathcal{M}$. The set of vector fields $\mathcal{G}$ is

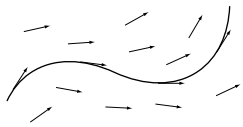

Figure 12.7 An integral curve of a vector field.

said to generate a *distribution* $\mathcal{D} \subseteq T\mathcal{M}$, which is a smooth assignment of a linear subspace of $T_x\mathcal{M}$ for each $x \in \mathcal{M}$. A distribution is *regular* if the dimension of the linear subspace is the same at all x. If the dimension is m, then we say that it is an m-dimensional distribution.

Consider the two-dimensional regular distribution for the unicycle $\mathcal{D} = \text{span}(\{g_1^{\text{uni}}, g_2^{\text{uni}}\}) = u_1 g_1^{\text{uni}}(x) + u_2 g_2^{\text{uni}}(x), u_1, u_2 \in \mathbb{R}$. We might think of this as the "positive" form of the distribution—feasible motions are generated by linear combinations of the vector fields. A "negative" form of the distribution would start with all motions being feasible, then eliminate those that violate motion constraints. For instance, the unicycle distribution could be written

$$\mathcal{D}(x) = \{\dot{x} \in T_x\mathcal{M} \mid \omega(x)\dot{x} = 0\}, \quad \omega(x) = [-\sin x_3, \cos x_3, 0], \tag{12.1}$$

where $\mathcal{D}(x)$ is the linear subspace of $T_x\mathcal{M}$ defined by the distribution $\mathcal{D}$. A row vector $\omega(x)$ is called a *covector* and lives in the *cotangent space* $T_x^*\mathcal{M} = \mathbb{R}^n$, the dual of $T_x\mathcal{M}$ consisting of all linear functionals of elements of $T_x\mathcal{M}$. In other words, a *covector field* ω pairs with a vector field g to yield a real value, $\omega(x)g(x) \in \mathbb{R}$. This is sometimes called the "natural pairing" of a tangent vector and covector. The canonical basis of covector fields is $\{dx_1, \ldots, dx_n\}$, so that the constraint $\omega(x)$ in equation (12.1) can be written as $-\sin x_3 dx_1 + \cos x_3 dx_2$.

A covector field ω is sometimes known as a *one-form,* because it takes a single element of $T_x\mathcal{M}$ and produces a real number, linear in the tangent vector. A *two-form,* as we will see in section 12.4, takes two elements of $T_x\mathcal{M}$ and produces a real number, linear in each of the arguments.

The *cotangent bundle* $T^*\mathcal{M}$ is the union of cotangent spaces $T_x^*\mathcal{M}$ for all $x \in \mathcal{M}$. A set of covector fields $\{\omega_1(x), \ldots, \omega_k(x)\}$ is said to define a *codistribution* $\Omega \subseteq T^*\mathcal{M}$. If the covector fields $\omega_i(x)$, $i = 1 \ldots k$, correspond to motion constraints $\omega_i(x)\dot{x} = 0$, then Ω is called a *constraint* codistribution, and it is said to *annihilate* the distribution $\mathcal{D}$ of feasible motions, and vice versa.

Of special interest are velocity constraints of the form

$$f(q, \dot{q}) = 0 \tag{12.2}$$

that cannot be integrated to yield equivalent configuration constraints. Such constraints are called *nonholonomic*. Nonholonomic constraints of the form

$$a(q)\dot{q} = 0$$

are sometimes called *Pfaffian* constraints, as discussed in chapter 10. Pfaffian constraints arise from rolling without slip [e.g., see equation (12.1)] and conservation of angular momentum. In mechanical systems, the covector field $a(q)$ can be interpreted as a generalized force, so $a(q)\dot{q}$ has units of power, and the constraint $a(q)\dot{q} = 0$ is *passive*—it does no work on the system.

In second-order underactuated systems, the underactuation implies the existence of acceleration constraints of the form

$$f(q, \dot{q}, \ddot{q}) = 0.$$

Constraints of this form that cannot be integrated to equivalent velocity constraints are sometimes referred to as "second-order nonholonomic" constraints, but this terminology is not standard.

In general, it is not easy to determine if an acceleration constraint can be integrated to yield an equivalent velocity constraint, or if a velocity constraint can be integrated to yield an equivalent configuration constraint. In the rest of this chapter, we use the "positive" form of the distribution and study the reachable set by vector fields in $\mathcal{G}$.

12.1.3 Lie Brackets

Let $\mathcal{G}$ be a set of vector fields and $\mathcal{D}$ be the distribution defined by span($\mathcal{G}$). We would like to know the reachable set of $\mathcal{M}$ by following vector fields in $\mathcal{D}$. While this is generally difficult globally, it is possible to learn something about the reachable set *locally* by looking at the *Lie brackets* of vector fields in $\mathcal{D}$. Given two vector fields belonging to $\mathcal{D}$, the Lie bracket tells us if infinitesimal motions along these vector fields can be used to locally generate motion in a direction not contained in $\mathcal{D}$. Perhaps the best-known example is the parallel-parking maneuver for a car or, in our case, a unicycle. Direct sideways motion is prohibited by the no-slip constraint, but sideways motion can be approximated by a series of forward-backward and turning maneuvers. The implication of this is that the locally reachable set of $\mathcal{M}$ is not two-dimensional, as the two-dimensional distribution $\mathcal{D}$ might seem to indicate, but fully three-dimensional. The no-slip velocity constraint does not imply a constraint on reachable configurations.

For two vector fields $g_1, g_2 \in \mathcal{G}$, consider the state reached from $x_0 = x(0)$ by first following g_1 for a small time $\epsilon \ll 1$, then following g_2 for time ϵ, then following $-g_1$ for time ϵ, then following $-g_2$ for time ϵ. This is expressed mathematically as

$$x(4\epsilon) = \phi_\epsilon^{-g_2}\left(\phi_\epsilon^{-g_1}\left(\phi_\epsilon^{g_2}\left(\phi_\epsilon^{g_1}(x_0)\right)\right)\right). \tag{12.3}$$

We can take a Taylor series in ϵ to solve the differential equation (12.3) approximately (see, e.g., [330] and problem 29), yielding

$$x(4\epsilon) = x_0 + \epsilon^2 \left(\frac{\partial g_2}{\partial x} g_1(x_0) - \frac{\partial g_1}{\partial x} g_2(x_0) \right) + O(\epsilon^3), \tag{12.4}$$

where the partial derivatives are evaluated at x_0 and $O(\epsilon^3)$ indicates terms of order ϵ^3, which are dominated by the term of order ϵ^2 when ϵ is small. Note there are no $O(\epsilon)$ terms. The ϵ^2 term represents the approximate net motion of the system, and the term inside the parentheses is the Lie bracket of g_1 and g_2.

The Lie bracket of g_1 and g_2 is written $[g_1, g_2]$ and is given in local coordinates by

$$[g_1, g_2] = \frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2. \tag{12.5}$$

The Lie bracket $[g_1, g_2]$ defines a new vector field, and if it is not contained in span($\mathcal{G}$), then it represents a new motion direction that can be followed approximately. Locally generating motion in this direction is "slower" than following the vector field g_1 or g_2 directly, as the net motion is only $O(\epsilon^2)$ for time $O(\epsilon)$, where $\epsilon \ll 1$. Again, parallel parking is a well-known example, as approximately generating sideways motion by forward-backward and turning motions is tedious and time-consuming. If $[g_1, g_2] = 0$, then no new motion is created, and the two vector fields are said to *commute*.

Since $[g_1, g_2]$ is a vector field, we can calculate its Lie bracket with another vector field. A *Lie product of degree k* is a bracket term where the original vector fields appear k times. For instance, $[[[g_1, g_2], g_1], g_2]$ is a Lie product of degree 4.

EXAMPLE 12.1.5 Unicycle (cont.) *The rolling and turning vector fields for the unicycle are* $g_1^{\text{uni}} = [\cos x_3, \sin x_3, 0]^T$ *and* $g_2^{\text{uni}} = [0, 0, 1]^T$*, respectively. So*

$$\begin{aligned}
[g_1^{\text{uni}}, g_2^{\text{uni}}] &= \frac{\partial g_2^{\text{uni}}}{\partial x} g_1^{\text{uni}} - \frac{\partial g_1^{\text{uni}}}{\partial x} g_2^{\text{uni}} \\
&= \begin{bmatrix} \frac{\partial g_2^{\text{uni}}}{\partial x_1} & \frac{\partial g_2^{\text{uni}}}{\partial x_2} & \frac{\partial g_2^{\text{uni}}}{\partial x_3} \end{bmatrix} g_1^{\text{uni}} - \begin{bmatrix} \frac{\partial g_1^{\text{uni}}}{\partial x_1} & \frac{\partial g_1^{\text{uni}}}{\partial x_2} & \frac{\partial g_1^{\text{uni}}}{\partial x_3} \end{bmatrix} g_2^{\text{uni}} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos x_3 \\ \sin x_3 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -\sin x_3 \\ 0 & 0 & \cos x_3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \sin x_3 \\ -\cos x_3 \\ 0 \end{bmatrix}.
\end{aligned}$$

Note that the Lie bracket motion is to the side, in the direction prevented by the no-slip constraint (see figure 12.8).

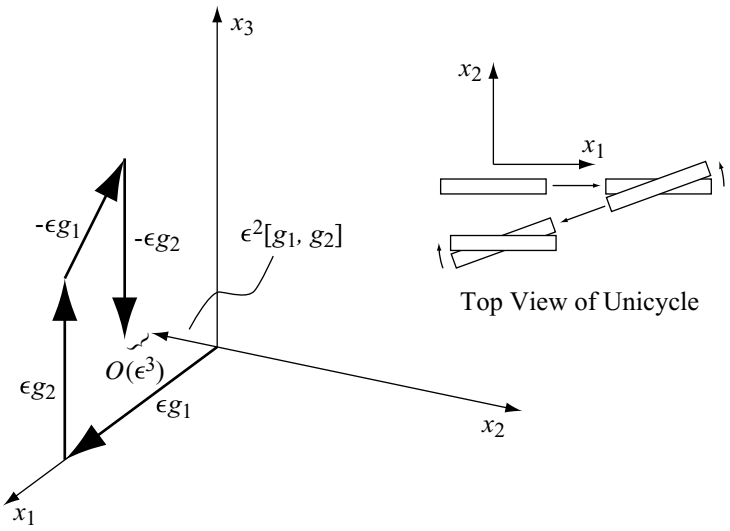

Figure 12.8 Generating a Lie bracket motion for the unicycle, starting from the origin. The net motion is approximately to the side, the Lie bracket direction. It is not exactly to the side, however, due to higher-order terms in ϵ.

The *Lie algebra* of a set of vector fields $\mathcal{G}$, written $\overline{\text{Lie}}(\mathcal{G})$, is the linear span of all Lie products, of all degrees, of vector fields in $\mathcal{G}$. To determine the Lie algebra, define $\mathcal{G}_1 = \mathcal{G}$, and the series

$$\mathcal{G}_{i+1} = \mathcal{G}_i \cup \{[g_j, g_k] \mid \forall g_j \in \mathcal{G}_1, g_k \in \mathcal{G}_i\}.$$

Then $\overline{\text{Lie}}(\mathcal{G})$ is given by the distribution $\text{span}(\mathcal{G}_\infty)$. For example, the series for $\mathcal{G} = \{g_1, g_2\}$ begins

$$\mathcal{G}_1 = \{g_1, g_2\}$$
$$\mathcal{G}_2 = \mathcal{G}_1 \cup \{g_3 = [g_1, g_2]\}$$
$$\mathcal{G}_3 = \mathcal{G}_2 \cup \{g_4 = [g_1, g_3], g_5 = [g_2, g_3]\}$$
$$\mathcal{G}_4 = \mathcal{G}_3 \cup \{g_6 = [g_1, g_4], g_7 = [g_1, g_5], g_8 = [g_2, g_4], g_9 = [g_2, g_5]\}$$
$$\vdots$$

The corresponding series $\mathcal{D}_1 = \text{span}(\mathcal{G}_1), \mathcal{D}_2 = \text{span}(\mathcal{G}_2), \ldots$ is called the *filtration* of the distribution $\mathcal{D}_1$. The filtration is *regular* if each distribution in the filtration is regular. If the filtration is regular, then the dimension of the distribution grows at each step of the construction, or else the construction terminates. (Of course,

$\dim(\mathcal{D}_i) \leq n = \dim(\mathcal{M})$ for all i.) If the filtration is regular, we are guaranteed a finite value of k such that $\mathcal{D}_k = \mathcal{D}_{k+1} = \cdots = \mathcal{D}_\infty$. This distribution is the *involutive closure* $\overline{\mathcal{D}}$ of $\mathcal{D}$, and a distribution $\mathcal{D}$ is *involutive* if $\mathcal{D} = \overline{\mathcal{D}}$.

If the filtration is not regular, then in general there is no way to know a priori a degree k at which $\mathcal{D}_k = \overline{\mathcal{D}}$. If there is a degree k at which all Lie products become zero, then the Lie algebra is called *nilpotent of order k*.

The *integral manifold* of $\mathcal{D}$ containing x_0 is the set of $\mathcal{M}$ that can be reached from x_0 by vector fields in $\mathcal{D}$, and $\mathcal{D}(x)$ is the tangent space of the integral manifold at x. By the well known *Frobenius theorem*, an m-dimensional regular distribution $\mathcal{D}$ can be integrated to yield an m-dimensional integral manifold if and only if $\mathcal{D}$ is involutive.

If a distribution $\mathcal{D}$ does not have the entire space $\mathcal{M}$ as an integral manifold, then $\mathcal{D}$ is said to generate a *foliation* of $\mathcal{M}$, and each distinct integral manifold is called a *leaf* of the foliation. Consider, e.g., the one-dimensional distribution generated by $g_2^{\text{uni}} = [0, 0, 1]^T$ (turning motion) for the unicycle (see figure 12.6). The distribution is one-dimensional, regular, and involutive, and the integral manifolds are lines in x_3 (wrapping around at 2π) with fixed position (x_1, x_2). The unicycle is confined to the same leaf of the foliation for all time if it can only follow this vector field. A more interesting example of a foliation is given in example 12.1.7 below.

The existence of integral manifolds smaller than the whole state space $\mathcal{M}$ indicates that the motion constraints actually limit the reachable state space. For example, velocity constraints on a kinematic system might be integrated to yield configuration constraints, indicating that the original constraints are actually holonomic. Similarly, acceleration constraints on a mechanical system might be integrated to yield velocity or even configuration constraints.

Lie brackets satisfy the following properties:

1. Skew-symmetry:

$$[g_1, g_2] = -[g_2, g_1]$$

2. Jacobi identity:

$$[g_1, [g_2, g_3]] + [g_3, [g_1, g_2]] + [g_2, [g_3, g_1]] = 0$$

Taking these properties into account, the *Philip Hall basis* gives a way to choose the smallest number of Lie products that must be considered at each degree k to generate a basis for the distribution $\mathcal{D}_k$. See the book by Serre [380] for details.

EXAMPLE 12.1.6 Unicycle (cont.) *From before, we have* $g_1^{\text{uni}} = [\cos x_3, \sin x_3, 0]^T$, $g_2^{\text{uni}} = [0, 0, 1]^T$, *and* $g_3^{\text{uni}} = [g_1^{\text{uni}}, g_2^{\text{uni}}] = [\sin x_3, -\cos x_3, 0]^T$. *The dimension of the distribution defined by* $\{g_1^{\text{uni}}, g_2^{\text{uni}}, g_3^{\text{uni}}\}$ *is three at all* $x \in \mathcal{M}$, *implying that the distribution is regular. It is also certainly involutive, since the dimension of* $\mathcal{M}$ *is*

three. To see that the three vector fields are indeed linearly independent, we define the 3×3 *matrix* $[g_1^{\text{uni}}\ g_2^{\text{uni}}\ g_3^{\text{uni}}]$ *obtained by placing the column vectors side by side. The rank of the matrix is 3 at all* $x \in \mathcal{M}$*, which can be verified by the determinant*

$$\det\left[g_1^{\text{uni}}\ g_2^{\text{uni}}\ g_3^{\text{uni}}\right] = \det\begin{bmatrix} \cos x_3 & 0 & \sin x_3 \\ \sin x_3 & 0 & -\cos x_3 \\ 0 & 1 & 0 \end{bmatrix} = 1.$$

Since the distribution is regular and involutive, it has a three-dimensional integral manifold, which is the entire space $\mathcal{M}$*. The distribution* $\mathcal{D}_2$ *is the involutive closure of* $\mathcal{D}_1$*. The filtration is regular.*

EXAMPLE 12.1.7 *Define the vector fields* $g_1(x) = [x_1 \cos x_3, x_2 \sin x_3, 0]^T$ *and* $g_2(x) = [0, 0, 1]^T$ *on* $\mathbb{R}^3$*. The vector field* g_2 *by itself defines a regular one-dimensional involutive distribution. The vector field* g_1 *does not, however, as it vanishes at* $x_1 = x_2 = 0$*. The Lie bracket of these vector fields is* $[g_1, g_2] = [x_1 \sin x_3, -x_2 \cos x_3, 0]^T$*, and*

$$\det[g_1\ g_2\ [g_1, g_2]] = x_1 x_2.$$

This means that the distribution $\text{span}(\{g_1, g_2, [g_1, g_2]\})$ *is rank 3 at points where both* x_1 *and* x_2 *are nonzero. It is not regular, as the rank is less at points where either* x_1 *or* x_2 *is zero. In fact, it is not hard to see that the integral manifold of this distribution is one-dimensional from points* $[0, 0, x_3]^T$*, two-dimensional from points* $[x_1 \neq 0, 0, x_3]^T$ *and* $[0, x_2 \neq 0, x_3]^T$*, and three-dimensional from all other points. The foliation is pictured in figure 12.9.*

EXAMPLE 12.1.8 PBWT (cont.) *As derived previously, we have* $g_0^{\text{pbwt}} = [x_4, x_5, x_6, 0, a_g, 0]^T$*,* $g_1^{\text{pbwt}} = [0, 0, 0, \cos x_3, \sin x_3, 0]^T$*, and* $g_2^{\text{pbwt}} = [0, 0, 0, -\sin x_3, \cos x_3, -d]^T$*. Lie bracket computations show that*

$$\begin{aligned}
g_3^{\text{pbwt}} &= \left[g_0^{\text{pbwt}}, g_1^{\text{pbwt}}\right] \\
&= [-\cos x_3, -\sin x_3, 0, -x_6 \sin x_3, x_6 \cos x_3, 0]^T \\
g_4^{\text{pbwt}} &= \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right] \\
&= [\sin x_3, -\cos x_3, d, -x_6 \cos x_3, -x_6 \sin x_3, 0]^T \\
g_5^{\text{pbwt}} &= \left[g_1^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right] \\
&= [0, 0, 0, d \sin x_3, -d \cos x_3, 0]^T \\
g_6^{\text{pbwt}} &= \left[g_0^{\text{pbwt}}, \left[g_1^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right]\right] \\
&= [-d \sin x_3, d \cos x_3, 0, dx_6 \cos x_3, dx_6 \sin x_3, 0]^T.
\end{aligned}$$

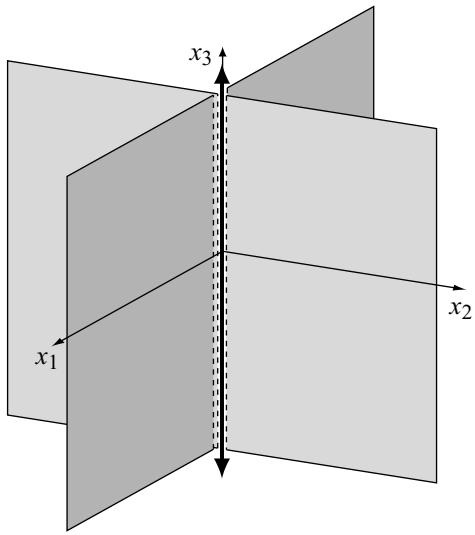

Figure 12.9 The distribution span($\{g_1, g_2, [g_1, g_2]\}$) in example 12.1.7 foliates the state space into nine separate leaves: the line defined by $x_1 = x_2 = 0$, four half-planes, and four three-dimensional quadrants.

A computation shows that

$$\det\left[g_1^{\text{pbwt}}\ g_2^{\text{pbwt}}\ g_3^{\text{pbwt}}\ g_4^{\text{pbwt}}\ g_5^{\text{pbwt}}\ g_6^{\text{pbwt}}\right] = d^4.$$

The dimension of the distribution defined by these vector fields is six at all $x \in \mathcal{M}$ (provided $d \neq 0$), so the distribution is both regular and involutive. The integral manifold is the entire space $\mathcal{M}$. The distribution $\mathcal{D}_4$ is the involutive closure of $\mathcal{D}_1$.

We now apply the ideas of this section to study controllability of underactuated systems, taking into account the fact that *controls* determine how the system vector fields are followed. It may not be possible to follow arbitrary linear combinations of system vector fields. For example, the drift vector field g_0^{pbwt} of the PBWT is fundamentally different from the control vector fields g_1^{pbwt} and g_2^{pbwt}.

12.2 Control Systems

A family of vector fields $\mathcal{G}$ on a manifold $\mathcal{M}$ is sometimes called a *dynamical polysystem*. The system is *symmetric* if for every $g \in \mathcal{G}$, $-g$ is also in $\mathcal{G}$.

The family of dynamical polysystems we will study are *control affine* nonlinear control systems, written

$$\dot{x} = g_0(x) + \sum_{i=1}^{m} g_i(x)u_i, \quad u \in \mathcal{U} \subset \mathbb{R}^m. \tag{12.6}$$

The vector field g_0 is called the *drift vector field,* defining the natural unforced motion of the system, and the g_i, $i = 1 \ldots m$, are linearly independent *control vector fields*. The control vector u belongs to the control set $\mathcal{U}$, and $u(t)$ is piecewise continuous. If $g_0 = 0$, the system is called *drift-free* or *driftless*. Kinematic systems (such as the unicycle) may be drift-free, but second-order systems (such as the PBWT) are not.

EXAMPLE 12.2.1 Unicycle (cont.) *The control system for the unicycle is written* $\dot{x} = g_1^{\text{uni}}(x)u_1 + g_2^{\text{uni}}(x)u_2$, *where* u_1 *is the driving speed and* u_2 *is the steering control.*

EXAMPLE 12.2.2 PBWT (cont.) *The control system for the PBWT is written* $\dot{x} = g_0^{\text{pbwt}}(x) + g_1^{\text{pbwt}}(x)u_1 + g_2^{\text{pbwt}}(x)u_2$, *where* u_1 *is the thrust force at thruster 1 and* u_2 *is the force at thruster 2.*

We will consider two classes of control sets:

- $\mathcal{U}_\pm$: This class of control sets includes any control set $\mathcal{U}$ containing the origin of $\mathbb{R}^m$ in the interior of its convex hull. In other words, the control set *positively* spans $\mathbb{R}^m$—any point in $\mathbb{R}^m$ can be generated by a positive linear combination of elements of $\mathcal{U}$. An example of such a control set is the cube centered at the origin of $\mathbb{R}^m$, $-1 \leq u_i \leq 1$, $i = 1, \ldots, m$. Another example consists of only the vertices of this cube.

- $\mathcal{U}_+$: This class of control sets includes $\mathcal{U}_\pm$ as a subset and includes any control set $\mathcal{U}$ that spans $\mathbb{R}^m$—any point in $\mathbb{R}^m$ can be generated by a linear combination of elements of $\mathcal{U}$. An example of such a control set is the non-negative controls $0 \leq u_i \leq 1$, $i = 1, \ldots, m$.

Examples of the control sets are shown in figure 12.10.

The system (12.6) is symmetric if it is drift-free and the control set is symmetric about the origin, e.g., a cube centered at the origin. We will abuse the term slightly and say that a drift-free system is symmetric for any positive-spanning control set $\mathcal{U} \in \mathcal{U}_\pm$, since the controllability properties we discuss in this chapter are the same for any $\mathcal{U} \in \mathcal{U}_\pm$.

If a system has drift but $g_0 \in \text{span}(\{g_1, \ldots, g_m\})$, then we may be able to choose controls $w(x) \in \mathbb{R}^m$ to always cancel the drift, thereby symmetrizing the system by

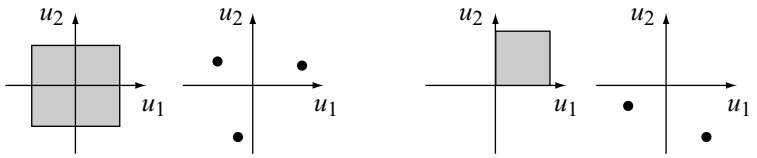

Figure 12.10 For $m = 2$ controls, the two control sets on the left belong to $\mathcal{U}_{\pm}$ and the two control sets on the right belong to $\mathcal{U}_{+}$.

the controls. In this case, the pseudocontrol $u \in \mathcal{U} \in \mathcal{U}_{\pm}$ can be added on top of the drift-canceling control $w(x)$, so the total control vector is $w(x) + u$, and the system is equivalent to the driftless system

$$\dot{x} = \sum_{i=1}^{m} g_i(x)u_i, \quad u \in \mathcal{U} \in \mathcal{U}_{\pm}.$$

As an intuitive example, imagine your motion as you walk on a conveyor. The drift vector field carries you at a constant speed in one direction. You can control your own walking speed, however, to cancel the drift and make progress in the opposite direction.

12.3 Controllability

Let V be a neighborhood of a point $x \in \mathcal{M}$ (i.e., an n-dimensional open set of $\mathcal{M}$ containing x). Let $R^V(x, T)$ indicate the set of reachable points at time T by trajectories remaining inside V and satisfying equation (12.6), and let

$$R^V(x, \leq T) = \bigcup_{0<t\leq T} R^V(x, t).$$

We define the following four versions of nonlinear controllability (see figure 12.11):

- The system is *controllable* from x if, for any $x_{\text{goal}} \in \mathcal{M}$, there exists a $T > 0$ such that $x_{\text{goal}} \in R^{\mathcal{M}}(x, \leq T)$. In other words, any goal state is reachable from x in finite time.

- The system is *accessible* from x if $R^{\mathcal{M}}(x, \leq T)$ contains a full n-dimensional subset of $\mathcal{M}$ for some $T > 0$. See figure 12.11(a).

- The system is *small-time locally accessible* (*STLA*) from x if $R^V(x, \leq T)$ contains a full n-dimensional subset of $\mathcal{M}$ for all neighborhoods V and all $T > 0$. See figure 12.11(b).

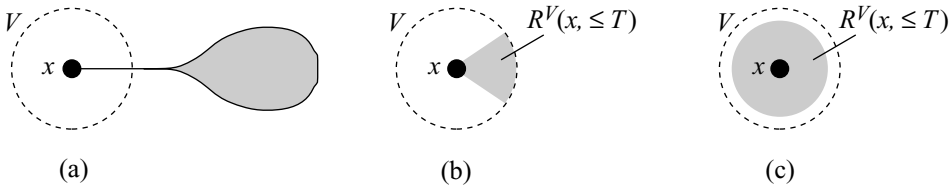

Figure 12.11 Reachable spaces for three systems on $\mathbb{R}^2$. (a) This system is accessible from x, but neither small-time locally accessible (STLA) nor small-time locally controllable (STLC). The reachable set is two-dimensional, but not while confined to the neighborhood V. (b) This system is STLA from x, but not STLC. The reachable set without leaving V does not contain a neighborhood of x. (c) This system is STLC from x.

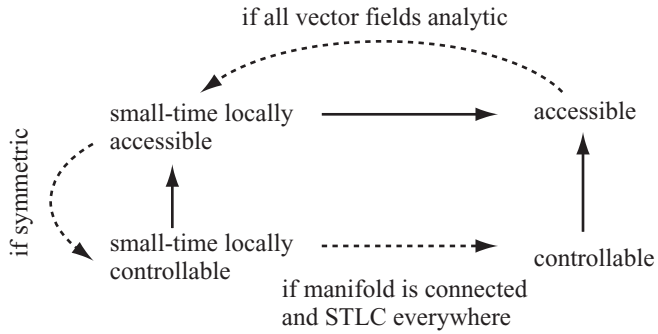

Figure 12.12 Implications among the controllability properties. Dashed arrows are conditional.

- The system is *small-time locally controllable* (*STLC*) from x if $R^V(x, \leq T)$ contains a neighborhood of x for all neighborhoods V and all $T > 0$. See figure 12.11(c).

The phrase "small-time" indicates that the property holds for any time $T > 0$, and "locally" indicates that the property holds for arbitrarily small (but full-dimensional) wiggle room around the initial state. For practical systems, it might take finite time to switch between controls (e.g., putting a car in reverse gear). In this case, we might say a system is locally, but not small-time, controllable. Here we ignore the switch time and retain the standard "small-time locally" terms.

If a property holds for all $x \in \mathcal{M}$, the phrase "from x" can be eliminated. Figure 12.12 shows the implications among the properties. If the vector fields are all analytic, then accessibility implies STLA.

Small-time local controllability is of special interest. STLC implies that the system can locally maneuver in any direction, and if the system is STLC at all $x \in \mathcal{M}$, then the system can follow any curve on $\mathcal{M}$ arbitrarily closely. This allows the system to maneuver through cluttered spaces, since any motion of a system with no motion constraints can be approximated by a system that is STLC everywhere. Also, if $\mathcal{M}$ is connected, then the system is controllable if it is STLC everywhere.

STLA and STLC are local concepts that can be established by looking at the behavior of the system in a neighborhood of a state. Accessibility and controllability, on the other hand, are global concepts. As a result, they may depend on things such as the topology of the space and nonlocal behavior of the system vector fields.

Some physical examples of the various properties:

- Imagine setting the minute and hour hands on a watch by turning a knob that can spin in only one direction. The configuration space of the hands is one-dimensional, since the motion of the hour hand is coupled to the motion of the minute hand. Show that this system is accessible, controllable, and STLA on the configuration space, but not STLC.

- Consider the system on $\mathbb{R}^2$ described by the drift vector field $g_0 = [x_2^2, 0]^T$ and the single control vector field $g_1 = [0, 1]^T$, where $u = u_1 \in [-1, 1]$. Show that the system is accessible and STLA from any x but neither controllable nor STLC.

- Consider the system on $\mathbb{R}^2$ described by the drift vector field $g_0 = [x_2, 0]^T$ and the single control vector field $g_1 = [0, 1]^T$, where $u = u_1 \in [-1, 1]$. This is the linear double-integrator $\ddot{q} = u$ written in the first-order form $\dot{x}_1 = x_2, \dot{x}_2 = u$. Convince yourself that the system is STLC only from zero-velocity states $[*, 0]^T$ (see figure 12.13).

- The unicycle satisfies all the controllability properties if $\mathcal{U} \in \mathcal{U}_{\pm}$.

- Show that the unicycle is accessible, STLA, and controllable in the obstacle-free plane, but not STLC, if $\mathcal{U}$ belongs to the class $\mathcal{U}_+$ but not $\mathcal{U}_{\pm}$.

- Any system confined to a k-dimensional integral manifold, $k < n$, satisfies none of the controllability properties.

As hinted at in the linear double-integrator example, for second-order systems with velocity variables in the state vector, STLC can only hold at zero velocity. States with nonzero velocity result in drift in the configuration variables that cannot be instantaneously compensated by finite actuation forces. *Therefore, when we talk about STLC for second-order systems, we implicitly mean STLC at zero velocity.*

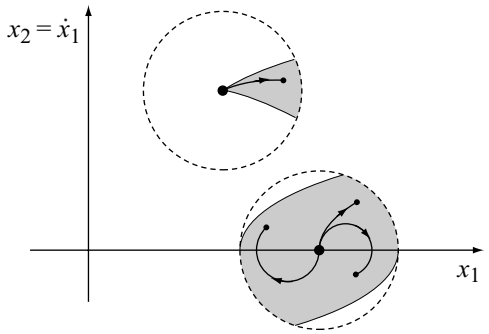

Figure 12.13 Two initial states and neighborhoods for the linear double-integrator with bounded control. The reachable sets from each initial state, by trajectories remaining in the neighborhood, are shaded, and example trajectories are shown. The system is STLC from the initial state where $\dot{x}_1 = x_2 = 0$, but not STLC from the initial state where $\dot{x}_1 = x_2 \neq 0$. Reaching a point left of this initial state (i.e., decreasing the x_1 value) requires $\dot{x}_1$ to become negative—the x_2 coordinate must leave the neighborhood.

For linear systems of the form $\dot{x} = Ax + Bu$, there is a single notion of controllability (see appendix J). For nonlinear systems, such as those we study, there are a number of notions of controllability, including the four we have defined here. A key point is that the linearizations of systems of interest to us are generally not controllable, meaning that their controllability is inherently a nonlinear phenomenon.

12.3.1 Local Accessibility and Controllability

Of the controllability properties, STLA can be checked by studying the Lie algebra of the vector fields $g_0, \ldots, g_m$.

THEOREM 12.3.1 *The system (12.6) is STLA from x if (and only if for analytic vector fields) it satisfies the* Lie algebra rank condition (LARC)*—the Lie algebra of the vector fields, evaluated at x, is the tangent space at x, or $\overline{\text{Lie}}(\{g_0, \ldots, g_m\})(x) = T_x\mathcal{M}$. This holds for any $\mathcal{U} \in \mathcal{U}_+$. If the system is symmetric (drift-free and $\mathcal{U} \in \mathcal{U}_\pm$), then the LARC also implies small-time local controllability.*

An early version of this result is due to W.-L. Chow [112], and it is sometimes called *Chow's theorem*.

EXAMPLE 12.3.2 Unicycle (cont.) *As shown previously, the rank of the unicycle Lie algebra is three at all states, so the LARC is satisfied. Therefore, for both $\mathcal{U} \in \mathcal{U}_+$*

and $\mathcal{U} \in \mathcal{U}_{\pm}$, the unicycle is STLA. For a control set $\mathcal{U} \in \mathcal{U}_{\pm}$, the system is also STLC everywhere, and therefore controllable because of the connectedness of its state manifold. It is also true that the unicycle is controllable (but not STLC) for any $\mathcal{U} \in \mathcal{U}_{+}$, though this cannot be shown by theorem 12.3.1. (The reader may wish to verify controllability by describing a constructive procedure to drive the unicycle to any goal location in an obstacle-free space.)

If we eliminate one vector field from the unicycle example, allowing it only to roll forward and backward (g_1^{uni}) or spin in place (g_2^{uni}), the unicycle is confined to an integral curve of the vector field, and none of the controllability properties is satisfied.

Second-order systems with nonzero drift, such as the PBWT, are not symmetric for any control set. The system may still be STLC at zero velocity states, however, since symmetry plus the LARC is sufficient but not necessary for STLC. Sussmann [401] provided a more general sufficient condition for STLC that includes the symmetric case ($g_0 = 0$ and $\mathcal{U} \in \mathcal{U}_{\pm}$) as a special case. To understand it, we first define a Lie product term to be a *bad bracket* if the drift term g_0 appears an odd number of times in the product and each control vector field g_i, $i = 1 \ldots m$, appears an even number of times (including zero). A *good bracket* is any Lie product that is not bad. For example, $[g_1, [g_0, g_1]]$ is a bad bracket and $[g_2, [g_1, [g_0, g_1]]]$ and $[g_1, [g_2, [g_1, g_2]]]$ are good brackets. With these definitions, we can state a version of Sussmann's theorem:

THEOREM 12.3.3 *The system (12.6) is STLC at x if*

1. $g_0(x) = 0$,
2. $\mathcal{U} \in \mathcal{U}_{\pm}$,
3. *the LARC is satisfied by good Lie bracket terms up to degree k, and*
4. *any bad bracket of degree $j \leq k$ can be expressed as a linear combination of good brackets of degree less than j.*

The intuition behind the theorem is the following. Bad brackets are called bad because, after generating the net motion obtained by following the Lie bracket motion prescription, we find that the controls u_i only appear in the net motion with even exponents, meaning that the vector field can only be followed in one direction. In this sense, a bad bracket is similar to a drift field, and we must be able to compensate for it. Since motions in Lie product directions of high degree are essentially "slower" than those in directions with a lower degree, we should only try to compensate for bad bracket motions by good bracket motions of lower degree. If a bad bracket of

degree j can be expressed as a linear combination of good brackets of degree less than j, the good brackets are said to *neutralize* the bad bracket. For the bad bracket of degree 1 (the drift vector field g_0) there are no lower degree brackets that can be used to neutralize it, so we require $g_0(x) = 0$. Therefore, this result only holds at states x where the drift vanishes, i.e., equilibrium states.

EXAMPLE 12.3.4 PBWT (cont.) *Assume that the PBWT moves in a horizontal plane, so* $a_g = 0$. *As before, we define* $g_3^{\text{pbwt}} = [g_0^{\text{pbwt}}, g_1^{\text{pbwt}}]$, $g_4^{\text{pbwt}} = [g_0^{\text{pbwt}}, g_2^{\text{pbwt}}]$, $g_5^{\text{pbwt}} = [g_1^{\text{pbwt}}, [g_0^{\text{pbwt}}, g_2^{\text{pbwt}}]]$, *and* $g_6^{\text{pbwt}} = [g_0^{\text{pbwt}}, [g_1^{\text{pbwt}}, [g_0^{\text{pbwt}}, g_2^{\text{pbwt}}]]]$. *Again as before, a computation shows that*

$$\det\left[g_1^{\text{pbwt}} \;\; g_2^{\text{pbwt}} \;\; g_3^{\text{pbwt}} \;\; g_4^{\text{pbwt}} \;\; g_5^{\text{pbwt}} \;\; g_6^{\text{pbwt}}\right] = d^4.$$

The LARC is satisfied, so the system is STLA at all states for either control set. If $\mathcal{U} \in \mathcal{U}_{\pm}$, *we would like to know if the system satisfies Sussmann's sufficient condition for STLC at equilibrium states* $x = [q_1, q_2, q_3, 0, 0, 0]^T$, *where* $g_0^{\text{pbwt}}(x) = 0$. *Because we use bracket terms up to degree 4 to demonstrate LARC, we must be able to neutralize all bad bracket terms of degree 4 or less. The only such bad bracket terms are the degree 3 terms*

$$\left[g_1^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_1^{\text{pbwt}}\right]\right] = [0, 0, 0, 0, 0, 0]^T$$

$$\left[g_2^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right] = [0, 0, 0, 2d\cos x_3, 2d\sin x_3, 0]^T = 2dg_1^{\text{pbwt}}.$$

The second term is neutralized by g_1^{pbwt}. *Therefore, by Sussmann's theorem, the system is STLC at equilibrium states.*

Note that in gravity, $a_g \neq 0$, *so* $g_0^{\text{pbwt}}(x) \neq 0$ *at any state and Sussmann's theorem does not allow us to prove or disprove STLC.*

Now consider the case where the PBWT is equipped with a single thruster. If the single thruster corresponds to the vector field g_1^{pbwt}, *the thrust always passes through the body center of mass, and the angular velocity of the body cannot be changed. The system is not accessible. If the single thruster corresponds to the vector field* g_2^{pbwt}, *however, we can define the vector fields*

$$\left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right], \quad \left[g_2^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right], \quad \left[g_2^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right]\right],$$

$$\left[g_2^{\text{pbwt}}, \left[g_2^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right]\right]\right],$$

$$\left[g_0^{\text{pbwt}}, \left[g_2^{\text{pbwt}}, \left[g_2^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, \left[g_0^{\text{pbwt}}, g_2^{\text{pbwt}}\right]\right]\right]\right]\right]$$

and see that the determinant of the matrix formed by these columns is $-16d^8$, *indicating that the system is STLA for either* $\mathcal{U} \in \mathcal{U}_+$ *or* $\mathcal{U} \in \mathcal{U}_{\pm}$. *Bad brackets cannot be neutralized, so theorem 12.3.3 cannot be used to show STLC. Note, however, that*

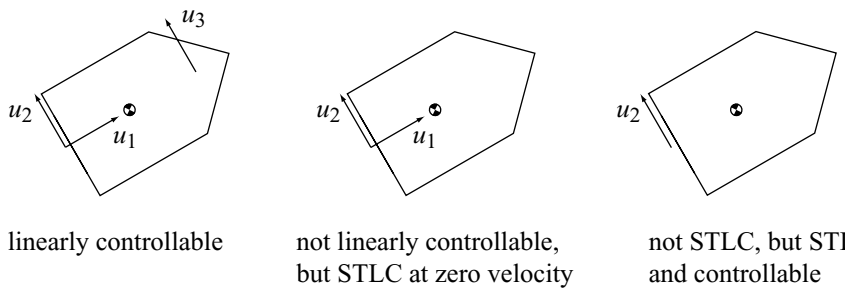

Figure 12.14 The PBWT in zero gravity with different numbers of thrusters. The PBWT on the left has three thrusters that can generate any force and torque combination, and it is controllable by linear control theory. Eliminating one thruster, we get the PBWT in the middle, which is no longer linearly controllable but is STLC at zero velocity. Finally, reducing the thruster count to one, we get the PBWT on the right, which is no longer STLC but remains STLA and controllable in a global sense. (Note that the PBWT with only u_1 thrust is not STLA.) All thrusters are bidirectional.

reducing to a single control vector field does not reduce the dimension of the reachable space, as it did for the kinematic unicycle case. This is because the second-order system provides a drift field with which Lie bracket terms can be generated.

Finally, the PBWT with the single control vector field g_2^{pbwt}*, a control set* $\mathcal{U} \in \mathcal{U}_{\pm}$*, and* $a_g = 0$ *turns out to be (globally) controllable — any state is reachable in finite time from any other state [303]. Thus the PBWT in zero gravity provides a simple example of different controllability properties (figure 12.14). If we equip it with three independent control vector fields, e.g., a control for each coordinate, the PBWT is a linear system of three double-integrators and it is controllable by linear control theory (see appendix J). If we equip it with the two control vector fields* g_1^{pbwt} *and* g_2^{pbwt}*, it is no longer linearly controllable, but remains STLC at zero velocity. If we equip it with just the single control vector field* g_2^{pbwt}*, it is no longer STLC at zero velocity, but remains STLA and globally controllable.*

12.3.2 Global Controllability

For kinematic systems that are STLC everywhere on a connected manifold, (global) controllability follows easily. In general, however, controllability is not easy to decide, as it may depend on nonlocal features of the control system. In the special case of a control system (12.6) with $\mathcal{U} \in \mathcal{U}_{\pm}$ and a drift vector field that repeatedly returns the system to a neighborhood of its initial state, however, demonstrating controllability is as easy as demonstrating the LARC.

First, some definitions. Consider the flow ϕ^{g_0} of the drift vector field. A point $x \in \mathcal{M}$ is called *positively Poisson stable* (PPS) for g_0 if for all $T > 0$ and any neighborhood V of x, there exists a time $t > T$ such that the flow of the vector field returns the system to V, i.e., $\phi_t^{g_0}(x) \in V$. The drift vector field g_0 is called positively Poisson stable if the set of PPS points for g_0 is dense in $\mathcal{M}$.

A point $x \in \mathcal{M}$ is called a *nonwandering point* of g_0 if for all time $T > 0$ and any neighborhood V of x there exists a time $t > T$ such that $\phi_t^{g_0}(V) \cap V \neq \emptyset$, where $\phi_t^{g_0}(V) = \{\phi_t^{g_0}(x) \,|\, x \in V\}$. (A positively Poisson stable point is necessarily a nonwandering point.) The *nonwandering set* of g_0 is the set of all nonwandering points of g_0. Finally, we say that the drift vector field g_0 is *weakly positively Poisson stable* (WPPS) if its nonwandering set is $\mathcal{M}$.

We now state the main theorem, taken from Lian, Wang, and Fu [289]. Related results can be found in (Jurdjevic and Sussmann [212]; Lobry [295]; Brockett [65]; Bonnard [58]; and Jurdjevic [211]).

THEOREM 12.3.5 *Assume that the drift vector field g_0 is WPPS. Then the system (12.6) with $\mathcal{U} \in \mathcal{U}_\pm$ is controllable on $\mathcal{M}$ if the LARC is satisfied.*

As an example, consider the system on $\mathbb{R}^2$ described by $\dot{x} = g_0(x) + g_1(x)u_1$, $u_1 \in [-1, 1] \in \mathcal{U}_\pm$, where $g_0(x) = \frac{1}{2}[-x_2, x_1]^T$ and $g_1(x) = [1, 0]^T$. The drift vector field (shown in figure 12.15) is WPPS, as its orbits are closed. We find that $[g_0, g_1] = [0, -\frac{1}{2}]^T$ and $\det[g_1\ [g_0, g_1]] = -\frac{1}{2}$, so the LARC is satisfied at all x. By theorem 12.3.5, every state is reachable from every other state. Intuitively, u_1 is used to control x_1 and (waiting) time is used to "control" x_2. (In fact, in this example, it is

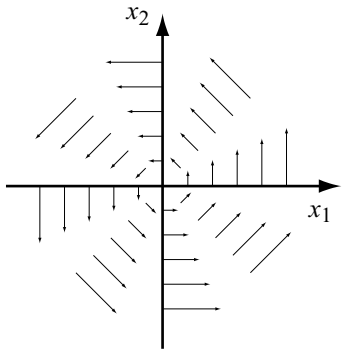

Figure 12.15 The integral curves of the WPPS drift vector field $\frac{1}{2}[-x_2, x_1]^T$ are closed (circles).

not hard to see that controllability also holds for $u_1 \in \mathcal{U} \in \mathcal{U}_+$.) This system is only STLC at the origin.

Theorem 12.3.5 is a powerful tool for establishing the global controllability of systems with drift. Systems with periodic natural unforced dynamics (such as an undamped planar pendulum or the example of figure 12.15) or energy-conserving drift on compact configuration spaces are examples of systems with WPPS drift vector fields. The latter follows from an application of Poincaré's recurrence theorem; see, e.g., the discussion by Arnold [26]. As an example, a rotating satellite moves on the compact configuration space $SO(3)$, and its natural unforced motion conserves energy. Therefore, the drift is WPPS. The LARC can be satisfied by a single body-fixed control torque, meaning that the satellite can be driven to any orientation and angular velocity with a single control vector field.

For systems with non-WPPS drift, it may be possible to construct feedback laws that always keep the system in a periodic orbit. If the system is always controllable about these periodic trajectories, i.e., if the system can reach neighborhoods of the controlled periodic trajectories, then similar reasoning can be used to demonstrate controllability of the system [87, 304].

12.4 Simple Mechanical Control Systems

This section discusses second-order mechanical control systems in greater depth, leading to simplified controllability tests and ideas that lead to reduced-complexity motion planning. We introduce the minimal set of ideas from Riemannian geometry that allows us to do this. We do not attempt to be rigorous or thorough in our treatment of mechanical systems from a geometric viewpoint. The motivated reader is instead referred to the books by Abraham and Marsden [10], do Carmo [131], Marsden and Ratiu [308], Bloch [50], Bullo and Lewis [77], and Boothby [60] for further study of differential geometry in mechanics and control. The results of this section are used in subsection 12.5.7, but otherwise this section can be skipped without affecting the reading of the rest of the chapter.

In chapter 10 we derived equations of motion of the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = T(q)f, \tag{12.7}$$

where f is a generalized force vector, $T(q)$ defines the action of f on the coordinates, $M(q)$ is the inertia matrix, $C(q,\dot{q})\dot{q}$ are Coriolis and centrifugal terms, and $g(q)$ are potential terms. Recall that $C(q,\dot{q})\dot{q} = \dot{q}^T\Gamma(q)\dot{q}$, where $\Gamma(q)$ is the set of n_Q^3 Christoffel symbols of the inertia matrix $M(q)$, and the computation $\dot{q}^T\Gamma(q)\dot{q}$ is

described in chapter 10.[2] We restrict our discussion in this section to *simple mechanical control systems* of this form with $g(q) = 0$.

Since we are considering underactuated systems, we can write $f = [u^T, 0^T]^T$, where $u \in \mathbb{R}^m$ is the control vector and 0 is an $(n_Q - m)$-vector of zeros. In this case, $T(q)$ can be written as an $n_Q \times m$ matrix, and the equations are

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = T(q)u. \tag{12.8}$$

Premultiplying both sides by $M^{-1}(q)$ (assuming full rank) and rearranging, we get

$$\ddot{q} = -M^{-1}(q)C(q, \dot{q})\dot{q} + M^{-1}(q)T(q)u. \tag{12.9}$$

Writing the m columns of $M^{-1}(q)T(q)$ as $Y_i(q)$, $i = 1 \ldots m$, we get

$$\ddot{q} = -M^{-1}(q)C(q, \dot{q})\dot{q} + \sum_{i=1}^{m} Y_i(q)u_i. \tag{12.10}$$

In other words, $\ddot{q}$ is the sum of a drift term due to Coriolis and centrifugal effects and a term due to the controls. According to the development so far, the control system can be expressed in the form of equation (12.6) by writing the state, drift vector field, and control vector fields as

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \quad g_0(x) = \begin{bmatrix} \dot{q} \\ -M^{-1}(q)C(q, \dot{q})\dot{q} \end{bmatrix}, \quad g_i(x) = \begin{bmatrix} 0 \\ Y_i(q) \end{bmatrix}$$

and expressing the control system as

$$\dot{x} = g_0(x) + \sum_{i=1}^{m} g_i(x)u_i. \tag{12.11}$$

We can then apply controllability tests as previously described.

12.4.1 Simplified Controllability Tests

The approach described above, while correct, ignores some structure of the equations of motion of a simple mechanical control system. Lewis and Murray [285, 286] have studied the Lie bracket structure of simple mechanical control systems to derive simplified controllability tests at equilibrium states. These tests take advantage of the Lie bracket structure to reduce the number of computations. In the rest of this section, we will assume the control set $\mathcal{U}$ belongs to the class $\mathcal{U}_{\pm}$.

2. Most works in the differential geometry literature define a slightly different set of Christoffel symbols Γ^* that have the inverse of the inertia matrix embedded in them, such that $\dot{q}^T \Gamma^* \dot{q} = M^{-1}\dot{q}^T \Gamma \dot{q}$. We instead use the same definition used in chapter 10.

The key simplification is that we will study vector fields *only on the configuration space* $\mathcal{Q}$, *not the full state space* $T\mathcal{Q} = \mathcal{M}$. This will be possible because $M(q)$, $\Gamma(q)$, and $T(q)$ all depend on q only, not $\dot{q}$. To study dynamics, however, we must be able to define derivatives (accelerations) of vector fields on $\mathcal{Q}$. In particular, we need a definition of how a vector field $Y_2(q)$ on $\mathcal{Q}$ is changing (accelerating) along the direction of another vector field $Y_1(q)$. This is called the *covariant derivative* of $Y_2(q)$ with respect to $Y_1(q)$, and it is also a vector field on $\mathcal{Q}$.

To define the covariant derivative, we need to define the acceleration of a curve $q(t)$. Often we think of $\ddot{q}(t)$ as the "coordinate" acceleration, but by this definition, whether the system is accelerating or not depends on the choice of coordinates, as we will see shortly. The vector $\ddot{q}(t)$ is not generally contained in the tangent space $T_{q(t)}\mathcal{Q}$, and the misalignment is caused by nonzero Christoffel symbols of $M(q)$, which define how the configuration space "curves" in the coordinates q. To express the acceleration as an element of the tangent space, so that an observer living on $\mathcal{Q}$ can "see" it in the tangent space, we use the Christoffel symbols (i.e., the Coriolis terms) to project the coordinate acceleration $\ddot{q}$ back to the tangent space:

$$\text{acceleration} = \ddot{q} + M^{-1}(q)C(q,\dot{q})\dot{q} = \ddot{q} + M^{-1}(q)\dot{q}^T\Gamma(q)\dot{q} \in T_{q(t)}\mathcal{Q} \tag{12.12}$$

This is an "intrinsic" definition of acceleration independent of the coordinates chosen.

As a first example, imagine a point mass moving in a one-dimensional configuration space $\mathcal{Q} = S^1$, visualized as a unit circle in $\mathbb{R}^2$. If the position is described by $q = [x, y]^T$ in Cartesian coordinates and θ in polar coordinates, we have

$$q = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix},$$

with second derivative

$$\ddot{q} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \ddot{\theta}\begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} + \dot{\theta}^2\begin{bmatrix} -\cos\theta \\ -\sin\theta \end{bmatrix}.$$

The $\dot{\theta}^2$ term in $\ddot{q}$ enforces the constraint that the point mass stays on the unit circle. The $\ddot{\theta}$ term expresses the change in speed tangent to the circle, and this is the only acceleration visible to an observer in $\mathcal{Q}$ who cannot see the $\mathbb{R}^2$ space it is embedded in. The $\ddot{\theta}$ term is the intrinsic definition of acceleration we are looking for. We subtract the $\dot{\theta}^2$ term [the centripetal acceleration, corresponding to the negative of the velocity product term in equation (12.12)] from $\ddot{q}$ to project $\ddot{q}$ to the tangent space (see figure 12.16).

For an example that does not use a manifold embedded in a higher-dimensional space, consider a point mass m moving in the plane with no forces applied to it. We can write the equations of motion in either Cartesian coordinates (x, y) or polar

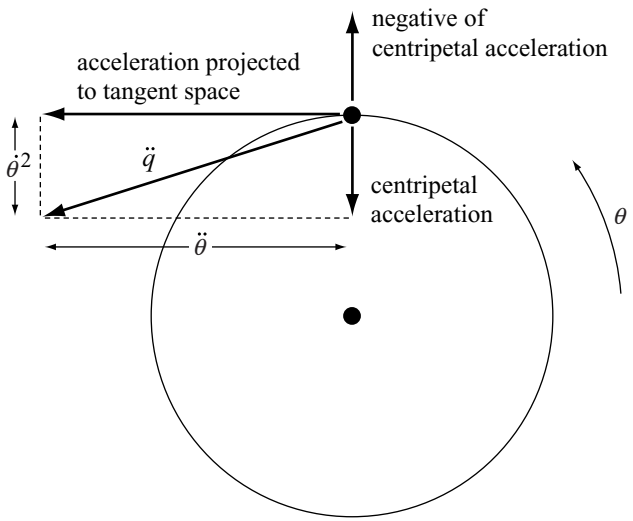

Figure 12.16 Subtracting the centripetal acceleration vector (pointing toward the center of the circle) from $\ddot{q} = [\ddot{x}, \ddot{y}]^T$ of a point mass moving around a circle gives an acceleration tangent to the circle.

coordinates (r, θ):

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} m & 0 \\ 0 & m\,r^2 \end{bmatrix} \begin{bmatrix} \ddot{r} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} -m\,r\dot{\theta}^2 \\ 2m\,r\dot{r}\dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

In Cartesian coordinates $q = [x, y]^T$, the mass moves such that $\ddot{x} = \ddot{y} = 0$. If we represent the configuration with polar coordinates $q = [r, \theta]^T$, however, the same motions of the mass will have $\ddot{r} \neq 0$, $\ddot{\theta} \neq 0$. So is the mass accelerating or not? The answer is that we should not think of $\ddot{q}$ as an acceleration; instead, we should think of the acceleration as being the inverse of the inertia matrix $M^{-1}(q)$ times the force. In both cases, then, since the force is zero, the acceleration is zero. The second time-derivatives $\ddot{r}$ and $\ddot{\theta}$ are not zero due to the nonzero Christoffel symbols of the inertia matrix in this choice of coordinates, as discussed in chapter 10.

Figure 12.17 shows paths of the point mass as it moves with zero force applied to it, shown in both Cartesian coordinates and polar coordinates. We say that each path in figure 12.17 is a *shortest path,* and the tangent vectors to these paths are *orthogonal* to each other at any intersection point. The notions of shortest paths (straight lines) and orthogonality are clear in our usual understanding of Euclidean geometry for the Cartesian coordinate case, but less clear in the polar coordinate case. In the polar coordinate case we need an inner product so we can define orthogonality. The inner

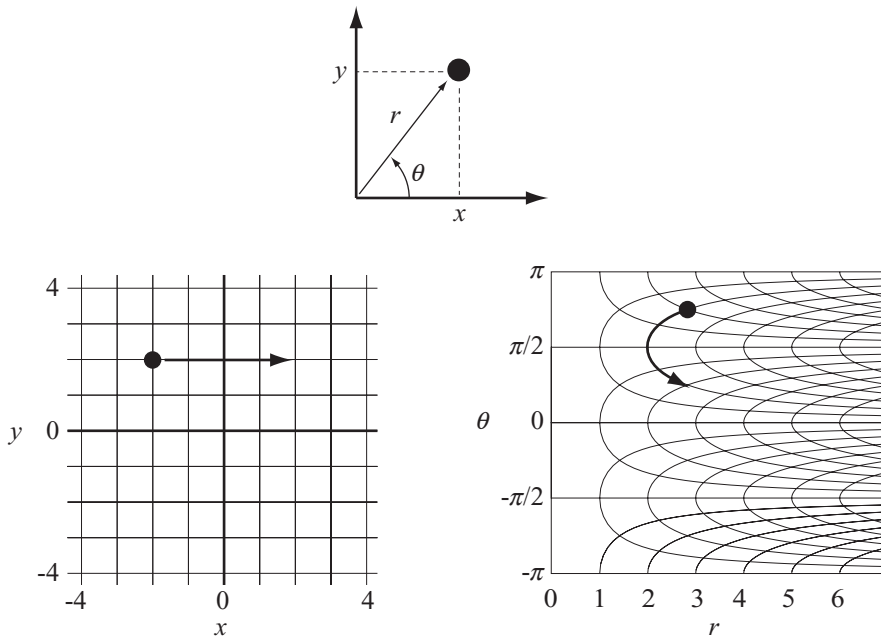

Figure 12.17 Example unforced motions of a point mass in the plane, represented in Cartesian coordinates (x, y) and polar coordinates (r, θ).

product also defines the distance metric, allowing us to talk about shortest paths between two points. For example, the length of each side of each grid "box" in the Cartesian plot of figure 12.17 is equivalent, just as the length of each side of a "box" in the polar plot is equivalent by its distance metric. Also, the "areas" of the "boxes" in the polar coordinate plot of figure 12.17 are equivalent, just as they are equivalent for each box in the Cartesian coordinate plot.

An inner product maps two tangent vectors $v_1, v_2 \in T_q\mathcal{Q}$ to $\mathbb{R}$. We write this inner product $\langle v_1, v_2 \rangle$, and v_1 and v_2 are orthogonal if $\langle v_1, v_2 \rangle = 0$. The inner product is associated with a metric $d : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ measuring the distance between two points. The metric and the inner product are related by the fact that the shortest path or *geodesic* between q_0 and q_f occurs for trajectories $q(t)$ minimizing the path length

$$\int_{t_0}^{t_f} \langle \dot{q}(t), \dot{q}(t) \rangle^{1/2} \, dt$$

out of all possible trajectories connecting $q(t_0) = q_0$ and $q(t_f) = q_f$. This relation allows us to call the inner product itself a metric.

As an example, the familiar Euclidean metric states that geodesics are straight lines, and the familiar inner product or "dot product" associated with the Euclidean metric is $\langle v_1, v_2 \rangle = v_1 \cdot v_2 = v_1^T v_2$. In the Euclidean metric, the rate of change of $Y_2(q)$ along the direction of $Y_1(q)$, i.e., the covariant derivative of $Y_2(q)$ with respect to $Y_1(q)$, is simply $(\partial Y_2/\partial q)Y_1$, i.e., the partial of Y_2 projected to the direction Y_1.

The Euclidean metric is natural for thinking about the motion of a point in space with standard Cartesian coordinates. Appropriate metrics for other systems are less obvious. For example, for a 2R robot arm, what does it mean for two velocities to be orthogonal? At a particular configuration of the arm, if v_1 and v_2 are Cartesian velocity vectors at the end-effector and w_1 and w_2 are the corresponding joint velocity vectors, then the condition $v_1^T v_2 = 0$ is not the same as $w_1^T w_2 = 0$. These conditions imply different and somewhat arbitrary metrics.

For the second-order mechanical systems we are interested in, the inertia matrix $M(q)$ defines a physically meaningful metric and inner product. This is sometimes called the kinetic energy metric, since it is associated with the kinetic energy $\frac{1}{2}\dot{q}^T M(q)\dot{q}$. The inner product defined by $M(q)$ is

$$\langle v_1, v_2 \rangle = v_1^T M(q) v_2, \quad v_1, v_2 \in T_q\mathcal{Q}.$$

The result has units of energy, which is physically meaningful and independent of the choice of coordinates. This metric is an example of a *Riemannian metric,* as it is bilinear (linear in each of v_1 and v_2) and symmetric and positive definite (because $M(q)$ is symmetric and positive definite). The kinetic energy metric defines a two-form, as it takes two elements of $T_q\mathcal{Q}$ and returns a real number linear in each of the two elements. When $M(q)$ is the identity matrix, the Euclidean metric is obtained.

The metric defines an *affine connection* ∇, which allows us to define the derivatives of vector fields in non-Euclidean spaces. For the affine connection ∇ associated with $M(q)$, the covariant derivative of $Y_2(q)$ with respect to $Y_1(q)$ is the vector field $\nabla_{Y_1(q)}Y_2(q)$,

$$\nabla_{Y_1(q)}Y_2(q) = \frac{\partial Y_2(q)}{\partial q}Y_1(q) + M^{-1}(q)Y_1^T(q)\Gamma(q)Y_2(q),$$

where the Christoffel symbols $\Gamma(q)$ describe how geodesics of the mechanical system "bend" in this choice of coordinates. For the Euclidean metric, the Christoffel symbols are zero.

The covariant derivative allows us to write the acceleration as an element of the tangent space to the configuration space, $\nabla_{\dot{q}(t)}\dot{q}(t) \in T_{q(t)}\mathcal{Q}$. We can write the equations

of motion (12.10) in the following equivalent form:

$$\begin{aligned}\nabla_{\dot{q}}\dot{q} &= \frac{\partial \dot{q}}{\partial q}\dot{q} + M^{-1}(q)\dot{q}^T\Gamma(q)\dot{q} \\ \nabla_{\dot{q}}\dot{q} &= \ddot{q} + M^{-1}(q)C(q,\dot{q})\dot{q} \\ \nabla_{\dot{q}}\dot{q} &= \sum_{i=1}^{m} Y_i(q)u_i, \quad u \in \mathcal{U} \in \mathcal{U}_{\pm}\end{aligned} \tag{12.13}$$

Equation (12.13) is the familiar $a = M^{-1}f$—acceleration is equal to the inverse of the mass times the force. Here, however, the acceleration a is not just the second derivative of the coordinates $\ddot{q}$, but $\ddot{q} + M^{-1}\dot{q}^T\Gamma\dot{q}$, accounting for the noninertial coordinates. We have $a = \ddot{q}$ only if the Christoffel symbols are zero, which is the case if the inertia matrix has no dependence on the configuration q.

Unforced motions $q(t)$, i.e., motions satisfying

$$\nabla_{\dot{q}(t)}\dot{q}(t) = 0,$$

are geodesics of ∇.

The covariant derivative also allows us to define the *symmetric product* of Y_1 and Y_2, the vector field

$$\langle Y_1 : Y_2 \rangle = \nabla_{Y_1} Y_2 + \nabla_{Y_2} Y_1,$$

which is useful in controllability calculations. But why should this be so? After all, we have already seen in equation (12.11) that a $Y_i(q)$ from equation (12.10) can be turned into a control vector field $g_i(x)$ on the full state space $\mathcal{M} = T\mathcal{Q}$, and that the drift can be expressed as $g_0(x)$, allowing us to take Lie brackets as before to test controllability. If we assume the system begins from rest, however, we notice that symmetric products of the $Y_i(q)$ appear again and again in the bracket terms. In particular, we can identify the following patterns in the calculations for $i, j = 1, \ldots, m$:

$$\begin{aligned}[g_i, g_j] &= \left[\begin{bmatrix} 0 \\ Y_i \end{bmatrix}, \begin{bmatrix} 0 \\ Y_j \end{bmatrix}\right] = 0, \\ [g_0, g_i] &= \begin{bmatrix} -Y_i \\ 0 \end{bmatrix}, \\ [g_i, [g_0, g_j]] &= \begin{bmatrix} 0 \\ \langle Y_i : Y_j \rangle \end{bmatrix}\end{aligned}$$

The first shows that no new motion directions will be created if the drift field is not included in the Lie bracket. The second shows that Lie bracketing the drift with a control vector field has the effect of taking the "acceleration" direction of the control vector field to a velocity direction (this only holds at zero velocity). The

last shows that iterated Lie brackets including the drift field can be evaluated by calculating lower-degree symmetric products of the $Y_i(q)$. An added benefit is that the symmetric product operates on vector fields with half as many elements.

The key point is, for controllability computations for simple mechanical control systems beginning from rest, the symmetric product allows us to think of $Y_i(q)$ *as a control vector field on the system configuration space* $\mathcal{Q}$*, without constructing a higher-dimensional vector field* $g_i(q)$ *on the full state manifold* $\mathcal{M} = T\mathcal{Q}$. The symmetric product captures the effect of drift in the controllability computations.

For a control system (12.13) consisting of a set of control vector fields $\mathcal{Y} = \{Y_1, \ldots, Y_m\}$, let the *symmetric closure* $\overline{\text{Sym}}(\mathcal{Y})$ be the distribution defined by $\mathcal{Y}$ and the iterated symmetric products of these vector fields. In this sense, the symmetric closure by the symmetric product is defined similarly to the involutive closure by the Lie bracket. Also, we can define the degree of a symmetric product to be the number of the original vector fields Y_i appearing in the expression. A symmetric product is *bad* if each of the vector fields appears an even number of times, and is *good* otherwise. With these definitions, Lewis and Murray [285, 286] proved the following theorem, building on theorem 12.3.3.

THEOREM 12.4.1 *Beginning from an equilibrium state* $x = [q^T, 0^T]^T$*, the system (12.13) is*

1. *STLA from* x *if and only if* $\overline{\text{Sym}}(\mathcal{Y})(q) = T_q\mathcal{Q}$*, and*

2. *STLC from* x *if* $\overline{\text{Sym}}(\mathcal{Y})(q) = T_q\mathcal{Q}$ *and every bad symmetric product can be expressed as a linear combination of good symmetric products of lower degree.*

Beginning from an equilibrium state $x = [q^T, 0^T]^T$, it is sometimes of interest to understand the locally reachable set of configurations irrespective of the velocities at those configurations. Lewis and Murray define a system to be *small-time locally configuration accessible* (*STLCA*) from q if the locally reachable set is full-dimensional on $\mathcal{Q}$, and *small-time locally configuration controllable* (*STLCC*) from q if the locally reachable set on $\mathcal{Q}$ contains q in the interior. A stronger condition than STLCC is *small-time local equilibrium controllability* (*STLEC*) from q if the locally reachable set contains zero velocity states forming a neighborhood of q on $\mathcal{Q}$. STLEC is stronger than STLCC, as STLEC demands that nearby configurations be reachable at zero velocity, while STLCC says nothing about the velocity. Finally, the system is *equilibrium controllable* if the system can reach any equilibrium state from any other equilibrium state.

Note that STLEC is a weaker property than STLC, as STLC requires that the locally reachable *states* contain a neighborhood of x on $T\mathcal{Q}$, while STLEC only requires that

the locally reachable *configurations* (at zero velocity) contain a neighborhood of q on $\mathcal{Q}$. STLEC is particularly relevant to systems subject to velocity constraints, which may be equilibrium controllable despite being confined to a reachable space of dimension less than n.

The following theorem, due to Lewis and Murray [285, 286], provides tests for STLCA, STLCC, STLEC, and equilibrium controllability.

THEOREM 12.4.2 *Beginning from an equilibrium state $x = [q^T, 0^T]^T$, the system (12.13) is*

1. *STLCA from q if and only if $\overline{\text{Lie}}(\overline{\text{Sym}}(\mathcal{Y}))(q) = T_q\mathcal{Q}$, and*

2. *both STLCC and STLEC from q if $\overline{\text{Lie}}(\overline{\text{Sym}}(\mathcal{Y}))(q) = T_q\mathcal{Q}$ and if every bad symmetric product can be expressed as a linear combination of good symmetric products of lower degree. If these conditions are satisfied at all $q \in \mathcal{Q}$, then the system is equilibrium controllable.*

Roughly speaking, $\overline{\text{Sym}}(\mathcal{Y})(q)$ corresponds to the reachable velocity directions from $[q^T, 0^T]^T$, and $\overline{\text{Lie}}(\overline{\text{Sym}}(\mathcal{Y}))(q)$ corresponds to the reachable configuration directions from $[q^T, 0^T]^T$.

EXAMPLE 12.4.3 PBWT (cont.) *The dynamics of the PBWT can be written*

$$\ddot{q} = -M^{-1}(q)C(q, \dot{q})\dot{q} + M^{-1}(q)T(q)u,$$

where $M^{-1}(q)$ is the 3×3 identity matrix, $C(q, \dot{q})\dot{q} = 0$ (the Christoffel symbols are all zero), $u = [u_1, u_2]^T$, and

$$T(q) = \begin{bmatrix} \cos q_3 & -\sin q_3 \\ \sin q_3 & \cos q_3 \\ 0 & -d \end{bmatrix}.$$

The control vector fields $\mathcal{Y} = \{Y_1(q), Y_2(q)\}$ are the two columns of $M^{-1}(q)T(q) = T(q)$. Calculating the symmetric product of $Y_1(q)$ and $Y_2(q)$, we get

$$\langle Y_1 : Y_2 \rangle = \nabla_{Y_1} Y_2 + \nabla_{Y_2} Y_1 = \begin{bmatrix} d \sin q_3 \\ -d \cos q_3 \\ 0 \end{bmatrix}.$$

We see that $\text{rank}(Y_1 \quad Y_2 \quad \langle Y_1 : Y_2 \rangle) = 3$ *for all q, so $\overline{\text{Sym}}(\mathcal{Y}) = T_q\mathcal{Q}$ for all q, and the system is STLA by theorem 12.4.1. The bad products are $\langle Y_1 : Y_1 \rangle = 0$ and $\langle Y_2 : Y_2 \rangle = [d \cos q_3, d \sin q_3, 0]^T$, which is neutralized by Y_1, so the system is also STLC by theorem 12.4.1. This confirms our previous result.*

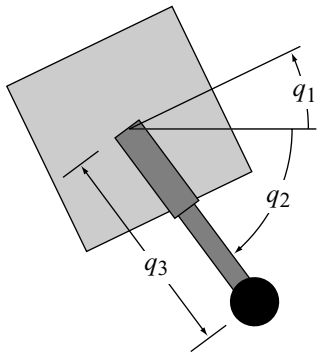

Figure 12.18 A simple single-leg hopping robot.

The reader may also wish to verify that $\text{rank}(Y_2 \;\; \langle Y_2 : Y_2 \rangle \;\; \langle Y_2 : \langle Y_2 : Y_2 \rangle \rangle) = 3$, *so the PBWT is STLA with the single thruster* u_2 *by theorem 12.4.1. This also confirms our previous result.*

EXAMPLE 12.4.4 Hopper example. *Consider a simple model of a planar single-leg hopping robot in flight, ignoring the translational motion of the body. We will model the body of the hopper as a rigid body pinned to a wall by a revolute joint at its center of mass. The inertia of the body is* I *about its center of mass. An extensible massless leg is attached to the center of mass of the body, and the foot of the leg is a point mass* m. *The configuration of the system is* $q = [q_1, q_2, q_3]^T$, *where* q_1 *is the angle of the body relative to an inertial frame,* q_2 *is the angle of the leg relative to the inertial frame, and* $q_3 > 0$ *is the extension of the leg (figure 12.18). The leg has two actuators, one providing a torque to control the orientation of the leg relative to the body, and the other providing a force to control the extension of the leg. We would like to know if it is possible to control the configuration* q *using only these two actuators.*

This system is subject to a velocity constraint: the total angular momentum of the system is zero throughout the motion (problem 27). This constraint implies that the dimension of the reachable state space can be no more than five, while $n = 2n_{\mathcal{Q}} = 6$. *This rules out the possibility of STLA and STLC, so instead we focus on the reachable configurations.*

The system mass matrix and its inverse are

$$M(q) = \begin{bmatrix} I & 0 & 0 \\ 0 & mq_3^2 & 0 \\ 0 & 0 & m \end{bmatrix}, \quad M^{-1}(q) = \begin{bmatrix} \frac{1}{I} & 0 & 0 \\ 0 & \frac{1}{mq_3^2} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix},$$

and the only nonzero Christoffel symbols are $\Gamma^2_{23} = \Gamma^2_{32} = mq_3$ *and* $\Gamma^3_{22} = -mq_3$. *The matrix* $T(q)$ *describing the generalized forces from the actuators is*

$$T(q) = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \end{bmatrix},$$

and $Y_1(q) = [1/I, -1/mq_3^2, 0]^T$, $Y_2(q) = [0, 0, 1/m]^T$. *Calculations show that*

$$\langle Y_1 : Y_1 \rangle = \begin{bmatrix} 0 \\ 0 \\ -\frac{2}{m^2 q_3^3} \end{bmatrix}, \quad \langle Y_1 : Y_2 \rangle = \langle Y_2 : Y_2 \rangle = 0, \quad [Y_1, Y_2] = \begin{bmatrix} 0 \\ -\frac{2}{m^2 q_3^3} \\ 0 \end{bmatrix}.$$

We see that Y_1, Y_2, *and* $[Y_1, Y_2]$ *span* $T_q\mathcal{Q}$ *for all* q *with* $q_3 > 0$. *Also, the bad symmetric product* $\langle Y_1 : Y_1 \rangle$ *is neutralized by* Y_2 *and the bad symmetric product* $\langle Y_2 : Y_2 \rangle$ *is zero. Therefore, by theorem 12.4.2, the system is STLEC at all* q *and equilibrium controllable. We also see that the distribution* $\overline{\mathrm{Sym}}(\mathcal{Y})$ *is only two-dimensional, so the system is not STLA by theorem 12.4.1.*

12.4.2 Kinematic Reductions for Motion Planning

In the controllability tests above, the symmetric product essentially allows us to treat the Y_i like velocity vector fields on a configuration space, thus halving the dimension of the vector fields in our controllability calculations. It is also sometimes possible to plan trajectories for underactuated mechanical systems as if they were kinematic systems. This reduction decreases the dimension of the search space by a factor of two. Since many search algorithms run in time exponential in the dimension of the search space, this reduction can greatly speed up motion planning.

Consider the original second-order mechanical system

$$\nabla_{\dot{q}}\dot{q} = \sum_{i=1}^{m} Y_i(q)u_i, \quad u \in \mathbb{R}^m, \tag{12.14}$$

and a first-order driftless kinematic system

$$\dot{q} = \sum_{i=1}^{\ell} V_i(q)w_i, \quad w \in \mathbb{R}^\ell, \quad w(t) \text{ continuous}, \tag{12.15}$$

where the set of control (velocity) vector fields is written $\mathcal{V} = \{V_1, \ldots, V_\ell\}$. We make the extra stipulation that $w(t)$ be continuous because discontinuous velocities would require infinite forces. Also, in this section on kinematic reductions, the control sets are taken to be unbounded for simplicity.

A kinematic system (12.15) is a *kinematic reduction* of a mechanical system (12.14) if all feasible trajectories for the kinematic system are also feasible for the second-order system. We further say that a mechanical system (12.14) is *maximally reducible to a kinematic system* if there exists a kinematic reduction such that all feasible trajectories of the mechanical system, starting with an initial velocity in span($\mathcal{Y}$), are also trajectories of the kinematic reduction. For example, all fully actuated mechanical systems are maximally reducible to kinematic systems—we can equivalently assume the controls are either (continuous) velocities or forces.

Some underactuated mechanical systems are also maximally reducible to kinematic systems. The test is given by the following theorem due to Lewis [283].

THEOREM 12.4.5 *A second-order mechanical system (12.14) is maximally reducible to a kinematic system if and only if* $\overline{\text{Sym}}(\mathcal{Y}) = \text{span}(\mathcal{Y})$.

EXAMPLE 12.4.6 Hopper (cont.) *Our previous calculations showed that* $\langle Y_1 : Y_2 \rangle = \langle Y_2 : Y_2 \rangle = 0$ *and* $\langle Y_1 : Y_1 \rangle \in \text{span}(Y_2)$. *Therefore* $\overline{\text{Sym}}(\mathcal{Y}) = \text{span}(\mathcal{Y})$ *and the hopper is maximally reducible to a kinematic system with* $\mathcal{V} = \mathcal{Y}$. *Intuitively, this is because the underactuation constraint of the hopper can be integrated to a velocity constraint: conservation of angular momentum. Therefore, any motion possible by controlling force on the leg extension and torque on the leg rotation is also possible by driving the kinematic reduction with* Y_1 *and* Y_2 *as velocity vector fields.*

A maximal kinematic reduction that generates all trajectories of the mechanical system [with initial velocity in span($\mathcal{Y}$)] could be called a *rank m kinematic reduction*, as the controlled velocities of the reduction form an m-dimensional distribution $\mathcal{V}$ with $\text{span}(\mathcal{V}) = \text{span}(\mathcal{Y})$. The class of underactuated systems admitting rank m kinematic reductions is relatively small, however, so we would like to explore kinematic reductions for more general underactuated systems. In particular, a system that is not maximally reducible to a kinematic system may nonetheless admit a *rank* 1 *kinematic reduction* $\dot{q} = V_1(q)w_1(t)$. A rank 1 kinematic reduction has a single control vector field V_1, also known as a *decoupling vector field*. (The word "decoupling" stems from the fact that trajectory planning for the second-order system along an integral curve of such a vector field can be decoupled into choosing the distance traveled along the integral curve, followed by time-scaling the path according to actuator limits. This brings to mind the decoupled trajectory planning approach for fully actuated systems in the previous chapter.)

An underactuated mechanical system can follow the integral curve of a decoupling vector field $V_1(q)$ at any speed and acceleration, i.e., for any continuous $w_1(t)$. A second-order mechanical system (12.14) can have no more than m linearly

independent decoupling vector fields at any q. For a maximally reducible mechanical system, every vector field in $\text{span}(\mathcal{Y})$ is a decoupling vector field.

How do we know if a proposed decoupling vector field $V \in \text{span}(\mathcal{Y})$ is actually decoupling? The following theorem provides the answer (Bullo and Lynch [79]).

THEOREM 12.4.7 *A vector field V is a decoupling vector field of the second-order mechanical system (12.14) if and only if $V \in \text{span}(\mathcal{Y})$ and $\nabla_V V \in \text{span}(\mathcal{Y})$.*

Stated another way, from the definition of the kinematic reduction (12.15) for the single control vector field $V(q)$, we have

$$\dot{q} = V(q)w \tag{12.16}$$

$$\ddot{q} = V(q)\dot{w} + \frac{\partial V}{\partial q} V(q)w^2, \tag{12.17}$$

and plugging these into the equations of motion (12.8), we see that $V(q)$ is decoupling if and only if there exists a $u \in \mathbb{R}^m$ satisfying the equations of motion (12.8) for all $w, \dot{w} \in \mathbb{R}$.

EXAMPLE 12.4.8 PBWT (cont.) *Decoupling vector fields for the PBWT are $V_1 = [\cos q_3, \sin q_3, 0]^T$ and $V_2 = [-\sin q_3, \cos q_3, -d]^T$. To verify this by theorem 12.4.7, we see that $V_1 = Y_1$, $V_2 = Y_2$, $\nabla_{V_1} V_1 = 0$, and $\nabla_{V_2} V_2 = [d \cos q_3, d \sin q_3, 0]^T \in \text{span}(Y_1)$. Note, however, that $\langle Y_1 : Y_2 \rangle \notin \text{span}(\mathcal{Y})$, so the PBWT is not maximally reducible to a kinematic system; i.e., linear combinations of V_1 and V_2 are* not *decoupling vector fields.*

Consider the physical meaning of these decoupling vector fields (figure 12.19). The vector field V_1 is pure translation of the PBWT along the line of action of the thruster u_1. It is clear that translation in this direction is possible at any speed and acceleration by proper choice of the thrust. The vector field V_2 corresponds to rotation of the PBWT about a point fixed relative to the body, the center of percussion *or* center of oscillation *of the PBWT relative to the line of action of thruster u_2. The center of percussion is the instantaneously unaccelerated point for nonzero thrust u_2. This point is a distance $1/d$ from the center of mass on a line through the center of mass and perpendicular to the line of action of u_2. (In the case that the PBWT has nonunit mass m and nonunit inertia I, the center of percussion is a distance I/md from the center of mass.) Rotation about this point is possible at any speed and acceleration by using u_2 to provide the torque to rotate the PBWT and using u_1 to keep the center of percussion stationary.*

In some cases the decoupling vector fields for a second-order system are apparent by inspection. In other cases it is possible to calculate the decoupling vector fields by

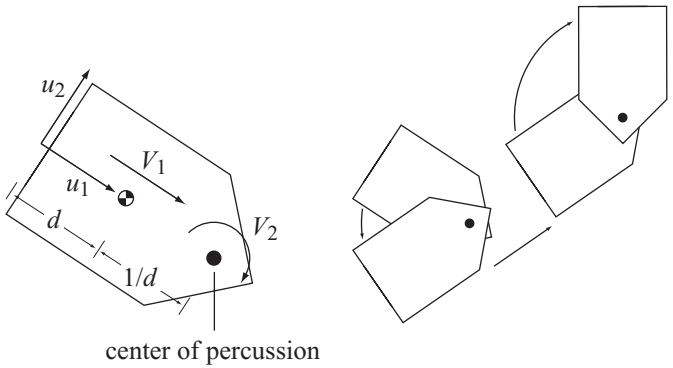

Figure 12.19 The decoupling vector fields V_1 and V_2 for the PBWT: translation along the line of action of u_1, and rotation about the center of percussion with respect to u_2. An example motion following these decoupling vector fields is shown.

solving a system of quadratic equations. To see this, recognize that any decoupling vector field $V(q)$ is contained in span($\mathcal{Y}$), so can be written in the form

$$V(q) = h_1(q)Y_1(q) + \cdots + h_m(q)Y_m(q), \quad h_i(q) \in \mathbb{R}.$$

The problem is to solve for functions $h_i(q)$ such that $\langle X_c, \nabla_V V \rangle = 0$ at all q, where the $X_c(q)$, $c = 1 \ldots n_{\mathcal{Q}} - m$, are linearly independent basis vectors of $\mathbb{R}^{n_{\mathcal{Q}}}$ that are orthogonal [according to $M(q)$] to $\mathcal{Y}(q)$ at all q.

Kinematic reductions allow us to plan a path using a kinematic system, then "lift" this path to a trajectory for the full second-order system. One important issue is to understand the locally reachable configurations of the kinematic reductions. Beginning from an equilibrium state $[q^T, 0^T]^T$, we say that a second-order mechanical system (12.14) is *small-time locally kinematically controllable* (*STLKC*) from q if there exists a set of decoupling vector fields $V_1, \ldots, V_p$ such that $\overline{\text{Lie}}(\{V_1, \ldots, V_p\})(q) = T_q\mathcal{Q}$. This means that a kinematic system that can only move along these decoupling vector fields is STLC at q. If this holds for all $q \in \mathcal{Q}$, the second-order system is *kinematically controllable*, meaning that it is equilibrium controllable by using only decoupling vector fields. These decoupling vector fields can therefore be used as primitives in a *kinematic* motion planner, reducing the complexity of the search for a feasible trajectory. A planned path would then consist of the concatenation of integral curves of the decoupling vector fields. Switches between decoupling vector fields must occur at zero velocity. Note that both STLC and STLKC imply STLEC, but no other implications hold generally.

The reader may easily verify that both the PBWT and the hopper are STLKC by two decoupling vector fields. For the PBWT, e.g., any configuration is reachable by concatenating integral curves of V_2, V_1, and V_2 (rotation, translation, and rotation).

12.4.3 Simple Mechanical Systems with Nonholonomic Constraints

The ideas discussed here also apply to second-order mechanical systems subject to nonholonomic constraints, such as rolling without slipping [78]. Such systems can be expressed as

$$M(q)\ddot{q} + \dot{q}^T\Gamma(q)\dot{q} = T(q)u + A^T(q)\lambda \tag{12.18}$$

$$A(q)\dot{q} = 0, \tag{12.19}$$

where $A(q)$ is a $k \times n_{\mathcal{Q}}$ matrix describing the k Pfaffian constraints $A(q)\dot{q} = 0$ and therefore the distribution of feasible velocities $\mathcal{D}$, and $A^T(q)\lambda \in \mathbb{R}^{n_{\mathcal{Q}}}$ is a set of constraint forces, where $\lambda \in \mathbb{R}^k$ is a vector of Lagrange multipliers. As described in chapter 10, we can eliminate λ from equation (12.18) using the matrix

$$P = \mathcal{I} - M^{-1}A^T(AM^{-1}A^T)^{-1}A$$

to project general motions to motions satisfying the nonholonomic constraints. We replace equation (12.18) with the $n_{\mathcal{Q}} - k$ independent equations

$$P(\ddot{q} + M^{-1}\dot{q}^T\Gamma\dot{q}) = PM^{-1}Tu. \tag{12.20}$$

In the notation of equation (12.13), we equivalently write

$$P(q)\nabla_{\dot{q}}\dot{q} = \sum_{i=1}^{m} P(q)Y_i(q)u_i, \quad u \in \mathbb{R}^m. \tag{12.21}$$

For the constrained system of equations (12.19) and (12.20), we can define the *constrained affine connection* $\widetilde{\nabla}$, and rewrite (12.21) as

$$\widetilde{\nabla}_{\dot{q}}\dot{q} = \sum_{i=1}^{m} \widetilde{Y}_i(q)u_i, \quad u \in \mathbb{R}^m, \tag{12.22}$$

where $\widetilde{Y}_i = PY_i$. For vector fields $\widetilde{Y}_1, \widetilde{Y}_2 \in \mathcal{D}$, the constrained affine connection $\widetilde{\nabla}$ is defined

$$\widetilde{\nabla}_{\widetilde{Y}_1}\widetilde{Y}_2 = P\nabla_{\widetilde{Y}_1}\widetilde{Y}_2.$$

Using the constrained affine connection $\widetilde{\nabla}$ and the constrained vector fields $\widetilde{\mathcal{Y}} = \{\widetilde{Y}_1, \ldots, \widetilde{Y}_m\}$ instead of ∇ and $\mathcal{Y}$, we can use the same simplified controllability tests and conditions for kinematic reductions that we used for unconstrained systems. Keep in mind that reachable velocities are confined to an $(n_{\mathcal{Q}} - k)$-dimensional distribution $\mathcal{D}$ due to the k velocity constraints. Therefore, the most we can hope for is STLEC and $\overline{\mathrm{Sym}}(\widetilde{\mathcal{Y}}) = \mathcal{D}$.

Although we will not use these here, a formula for calculating the Christoffel symbols $\widetilde{\Gamma}$ of $\widetilde{\nabla}$ is given by Lewis [284], which also holds for vector fields not

restricted to $\mathcal{D}$. Bullo and Zefran [80] give a computationally simpler formulation for vector fields restricted to $\mathcal{D}$. Their formulation uses an orthogonal basis of vector fields of the distribution $\mathcal{D}$ to construct modified Christoffel symbols. Choosing a basis of vector fields for the free motions of a nonholonomically constrained system is analogous to choosing a coordinate basis for the reachable configuration space of a holonomically constrained system, as discussed in chapter 10.

EXAMPLE 12.4.9 *In chapter 10 we considered the example of a knife-edge sliding on a plane. For this system, the inertia matrix $M(q)$ and the constraint matrix $A(q)$ are given by*

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, \quad A(q) = [\sin q_3, -\cos q_3, 0].$$

Assume that the available controls are u_1, a force in the q_1 direction, and u_3, a torque along q_3. Then

$$T(q) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad M^{-1}(q)T(q) = \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{I} \end{bmatrix},$$

and $Y_1 = [1/m, 0, 0]^T$, $Y_2 = [0, 0, 1/I]^T$. Using the projection matrix P defined by A and M,

$$P = \begin{bmatrix} \cos^2 q_3 & \sin q_3 \cos q_3 & 0 \\ \sin q_3 \cos q_3 & \sin^2 q_3 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

we get

$$\widetilde{Y}_1 = PY_1 = \left[\frac{\cos^2 q_3}{m}, \frac{\sin q_3 \cos q_3}{m}, 0\right]^T, \quad \widetilde{Y}_2 = PY_2 = \left[0, 0, \frac{1}{I}\right]^T,$$

$$\langle \widetilde{Y}_1 : \widetilde{Y}_1 \rangle = \langle \widetilde{Y}_2 : \widetilde{Y}_2 \rangle = 0,$$

$$\langle \widetilde{Y}_1 : \widetilde{Y}_2 \rangle = \left[-\frac{\sin q_3 \cos q_3}{mI}, -\frac{\sin^2 q_3}{mI}, 0\right]^T = -\frac{\tan q_3}{I}\widetilde{Y}_1,$$

$$[\widetilde{Y}_1, \widetilde{Y}_2] = \left[\frac{\sin 2q_3}{mI}, -\frac{\cos 2q_3}{mI}, 0\right]^T,$$

$$[\widetilde{Y}_2, [\widetilde{Y}_1, \widetilde{Y}_2]] = \left[2\frac{\cos 2q_3}{mI^2}, 2\frac{\sin 2q_3}{mI^2}, 0\right]^T.$$

We see that $\overline{\text{Sym}}(\{\widetilde{Y}_1, \widetilde{Y}_2\}) = \text{span}(\{\widetilde{Y}_1, \widetilde{Y}_2\})$*, so the system is maximally reducible to a kinematic system by theorem 12.4.5. The control (velocity) vector fields of the kinematic reduction are* $\widetilde{Y}_1$*, motion along straight lines (when* $q_3 \neq \pm\pi/2$*), and* $\widetilde{Y}_2$*, spinning in place.*

The distribution and filtration defined by $\{\widetilde{Y}_1, \widetilde{Y}_2\}$ *are not regular, as* $\widetilde{Y}_1$ *vanishes at* $q_3 = \pm\pi/2$*. We also see that*

$$\det[\widetilde{Y}_1 \;\; \widetilde{Y}_2 \;\; [\widetilde{Y}_1, \widetilde{Y}_2]] = \frac{\cos^2 q_3}{m^2 I^2},$$

so we cannot conclude STLCA at all q *until we construct* $[\widetilde{Y}_2, [\widetilde{Y}_1, \widetilde{Y}_2]]$ *and see*

$$\det[\widetilde{Y}_2 \;\; [\widetilde{Y}_1, \widetilde{Y}_2] \;\; [\widetilde{Y}_2, [\widetilde{Y}_1, \widetilde{Y}_2]]] = \frac{2}{m^2 I^4},$$

so the system is STLCA and STLEC at all q *by theorem 12.4.2.*

12.5 Motion Planning

Motion planning for nonholonomic and underactuated systems has been the subject of a great deal of recent research, and the results could easily fill several books (see, e.g., the books edited by Li and Canny [288] and Laumond [266]). In this section we summarize a few useful approaches. The approaches can be classified by the type of robot to which they apply (e.g., the structure of the equations of motion, and with or without control constraints or drift) or the nature of the problem (with or without obstacles or cost function to be minimized). Motion-planning approaches with roots in control theory tend to apply to systems with particular structure and no obstacles, while approaches based on search algorithms are computationally intensive and are suited to finding collision-free trajectories among obstacles. Some approaches attempt to combine the benefits of control-theoretic and search-based methods.

The problem is to find a motion $(x(t), u(t)), t \in [0, t_f]$ satisfying the equations of motion (12.6) such that $x(0) = x_{\text{start}}, x(t_f) = x_{\text{goal}}$. In the presence of obstacles, where $\mathcal{Q}_{\text{free}}$ represents the free configuration space, we also require $q(t) \in \mathcal{Q}_{\text{free}}, t \in [0, t_f]$.

12.5.1 Optimal Control

For some simple underactuated systems, it is possible to solve analytically for optimal controls transferring the system from one state to another using the ideas developed in the previous chapter. Consider, e.g., a driftless system with $m = 2$ controls and

$n = 3$ states and the control vector fields $g_1 = [1, 0, x_2]^T$ and $g_2 = [0, 1, 0]^T$. Optimal control for such a system was first studied by Brockett [66]. The system can be written

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_2 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{12.23}$$

Assume the time of motion is fixed, $t_f = 1$, and the objective is to minimize a measure of the control input energy:

$$J = \frac{1}{2} \int_0^{t_f} u^T u \, dt. \tag{12.24}$$

Then the Hamiltonian is written

$$\mathcal{H} = \frac{1}{2}\left(u_1^2 + u_2^2\right) + \lambda_1 u_1 + \lambda_2 u_2 + \lambda_3 x_2 u_1.$$

Solving the necessary condition $\partial \mathcal{H}/\partial u = 0$, we get

$$u_1 = -\lambda_1 - \lambda_3 x_2, \quad u_2 = -\lambda_2. \tag{12.25}$$

The adjoint equation $\dot{\lambda} = -\partial \mathcal{H}/\partial x$ indicates that λ_1 and λ_3 are constant, and $\dot{\lambda}_2 = -\lambda_3 u_1$. Differentiating equation (12.25) with respect to time, we get

$$\dot{u}_1 = -\dot{\lambda}_1 - \dot{\lambda}_3 x_2 - \lambda_3 \dot{x}_2 = -\lambda_3 u_2$$
$$\dot{u}_2 = -\dot{\lambda}_2 = \lambda_3 u_1.$$

These differential equations imply that optimal controls $u_1(t)$ and $u_2(t)$ are 90-degree out-of-phase sinusoids of the same amplitude and frequency, i.e.,

$$u_1(t) = u_1(0)\cos(\lambda_3 t) - u_2(0)\sin(\lambda_3 t) \tag{12.26}$$

$$u_2(t) = u_1(0)\sin(\lambda_3 t) + u_2(0)\cos(\lambda_3 t). \tag{12.27}$$

Given $x(0)$ and $x(1)$, the integrals of the equations of motion (12.23) define three equations to solve for λ_3, the frequency of the sinusoids, and the constants $u_1(0)$ and $u_2(0)$, defining the amplitude and phase.

These equations may be difficult to solve generally, but one simple case is of particular interest. We will choose controls so that at the end of the motion, x_1 and x_2 have returned to their initial values, while x_3 has changed from $x_3(0) = 0$ to $x_3(1) = x_{3,\text{goal}}$. In this case, $\lambda_3 = 2k\pi$, where k is any nonzero integer. This assures that

$$\int_0^1 u_1(t)dt = \int_0^1 u_2(t)dt = 0,$$

so x_1 and x_2 return to their initial values. Plugging $\lambda_3 = 2k\pi$ into the the controls [equations (12.26) and (12.27)], and putting the controls into the objective function (12.24), we find that

$$J = \frac{1}{2}(u_1(0)^2 + u_2(0)^2).$$

The cost of the motion is independent of the choice of k, so we choose $k = \pm 1$.

Integrating the equation of motion for x_3, we find that

$$x_3(1) = -\frac{u_1(0)^2 + u_2(0)^2}{4\pi}$$

for $k = 1$, and

$$x_3(1) = \frac{u_1(0)^2 + u_2(0)^2}{4\pi}$$

for $k = -1$. Therefore, if $x_{3,\text{goal}} > 0$, we choose $k = -1$ and any choice of $u_1(0)$ and $u_2(0)$ satisfying the condition $x_3(1) = x_{3,\text{goal}}$ (all choices have the same cost). If $x_{3,\text{goal}} < 0$, we choose $k = 1$ and proceed similarly. Notice that there is a one-dimensional set of solutions in the two-dimensional $(u_1(0), u_2(0))$ space, as we only need to satisfy the single equation $x_3(1) = x_{3,\text{goal}}$.

The motion described above suggests a strategy for motion planning for a more general class of systems. First, use the controls to drive m state variables directly to their goal values. Then perform motions that return these state variables to their goal values, but cause a desired net motion in the other state variables.

As an example, beginning from rest, a free-floating astronaut in space can control the orientation of his body by moving his arms in a cyclic pattern. At the end of a cycle, the *shape* (arm joint angles) of the astronaut is restored, but the orientation of his body has changed. (Keep in mind that the astronaut's total angular momentum is zero throughout the motion since there are no external forces.) We can decompose the astronaut's configuration into *shape* variables (also called *base* variables), describing the variables over which he has direct control (the arm joint angles), and *fiber* variables (the orientation of his body), which are coupled to the controls in a state-dependent manner.[3] In the example system (12.23), the shape variables are x_1 and x_2 and the fiber variable is x_3. In the single-leg hopper system of example 12.4.4, the shape variables are the leg angle q_2 and leg extension q_3, and the fiber variable is the body angle q_1 (see figure 12.20).

3. *Fiber controllability,* a weaker concept than complete controllability, concerns the controllability of the fiber variables without concern for the evolution of the shape variables. See [120, 233].

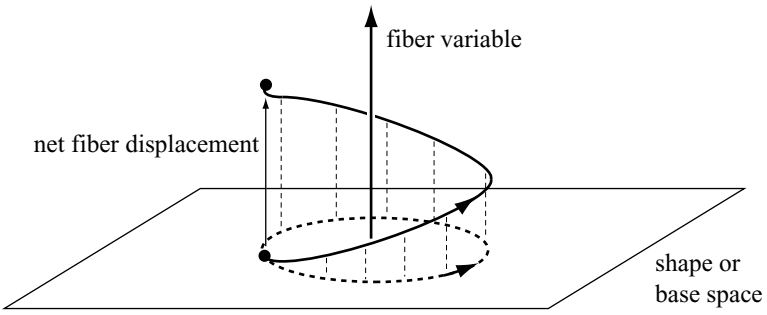

Figure 12.20 A closed loop in shape variables producing a net displacement in the fiber variable.

Many driftless systems can be transformed to a form similar to equation (12.23), allowing the control strategy of directly steering the shape variables to their goal configurations, and then performing closed loops (e.g., sinusoids) in the shape variables to achieve desired motions in the fiber variables. For example, the unicycle can be converted to this form by performing a coordinate transformation and an input feedback transformation. Define new coordinates $z = A(x)x$ and transformed controls $v = B(x)u$, where

$$A(x) = \begin{bmatrix} 0 & 0 & 1 \\ \cos x_3 & \sin x_3 & 0 \\ \sin x_3 & -\cos x_3 & 0 \end{bmatrix} \quad B(x) = \begin{bmatrix} 0 & 1 \\ 1 & x_2 \cos x_3 - x_1 \sin x_3 \end{bmatrix}.$$

Then the original unicycle system is transformed to an equivalent system of the form (12.23):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \cos x_3 & 0 \\ \sin x_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \iff \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ z_2 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

A system like this is an example of a broader class of *chained-form* systems, which are the topic of the following subsection.

The motion strategy of driving the shape variables to their goal values and then performing closed loops generally results in suboptimal motions, but it is rarely possible to solve the optimality conditions analytically. In any case, the quadratic "energy-like" objective function (12.24) may not have much physical meaning for an input-transformed system such as the unicycle above. Section 12.5.3 discusses numerical methods for finding approximately optimal motion plans.

12.5.2 Steering Chained-Form Systems Using Sinusoids

Consider the following system with $m = 2$ controls and $n \geq 3$ states, generalizing the system with $n = 3$ described above:

$$\dot{x} = u_1 g_1(x) + u_2 g_2(x), \quad g_1(x) = \begin{bmatrix} 1 \\ 0 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \end{bmatrix}, \quad g_2(x) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{12.28}$$

Such a system is said to be in *chained form*. Considering a Lie product of the form $[g_1, [g_1, \ldots [g_1, [g_1, g_2]]]]$, where g_1 appears k times, we find that the Lie product has a value $(-1)^k$ in the $k+2$ component of the vector field, and zeros in all other components. Therefore, the Lie algebra $\overline{\text{Lie}}(\{g_1, g_2\})$ is full rank, and the system is STLC at all x for $\mathcal{U} \in \mathcal{U}_{\pm}$.

To steer such a system, we can generalize the approach presented previously. First, drive x_1 and x_2 to their final values. Then choose controls $u_1(t)$ and $u_2(t)$ to be sinusoids at integrally related frequencies. For example, let

$$u_1(t) = a \sin 2\pi t \tag{12.29}$$

$$u_2(t) = b \cos 2k\,\pi t, \tag{12.30}$$

where k is a positive integer. Then $\dot{x}_3$ has components at frequency $2\pi(k-1)$, $\dot{x}_4$ has components at $2\pi(k-2)$, etc. Applying the controls for $t_f = 1$ will return the $x_1, \ldots, x_{k+1}$ variables to their initial values—the nonzero frequency means that $\dot{x}_1, \ldots, \dot{x}_{k+1}$ integrate to zero net change over the cycle. For the variables $x_{k+2}, \ldots, x_n$, however, the periodic controls will result in a nonzero DC (zero frequency) component in their time-derivatives, meaning that they will be changed over the cycle. The net change to x_{k+2} can be computed to be

$$x_{k+2}(1) - x_{k+2}(0) = \left(\frac{a}{4\pi}\right)^k \frac{b}{k!}. \tag{12.31}$$

Therefore, an algorithm to drive the system to the goal state is the following:

1. Drive the variables x_1 and x_2 directly to their goal values.

2. For each $k = 1 \ldots n-2$, in ascending order, apply the controls (12.29) and (12.30) for time $t_f = 1$ with a and b selected according to equation (12.31) to drive x_{k+2} to its desired value, leaving all $x_1, \ldots, x_{k+1}$ unchanged.

A number of n-state 2-input systems can be transformed to the chained form (12.28), including tractor-trailer systems with multiple trailers [393]. There are also other forms similar to the chained form presented here, including forms with more than two inputs; see, e.g., [85, 330, 332, 419].

Many motion-planning algorithms based on series expansions and averaging of the equations of motion use sinusoidal inputs at appropriately chosen frequencies and phases to "tickle" the system in desired Lie bracket directions (see, e.g., [34, 50, 74–76, 175, 279, 280, 327, 328, 333, 342, 422, 423]). This idea also applies to mechanical systems with drift. For a locomotion system where motions of the shape variables induce motion of the fiber variables, these sinusoidal inputs generate "gaits" for the system.

12.5.3 Nonlinear Optimization

The previous chapter outlined a method for using gradient-based nonlinear optimization to find locally optimal trajectories for fully actuated dynamic systems. A similar approach can be applied to underactuated systems [147, 148, 343]. We typically choose a finite parameterization of the control history $u(t)$, since any a priori parameterization of the trajectory $q(t)$ will likely describe trajectories that are infeasible for the system due to the underactuation constraints. Alternatively, we could solve for $q(t)$ and enforce equality constraints at time instants throughout the trajectory to ensure that there exists a feasible $u(t)$ for the given $q(t)$.

Two features of this approach are (1) it is very general—motion planning problems for many underactuated systems, including those with drift, can be encoded as nonlinear programs; and (2) the ability to minimize an objective function may result in motions that are "efficient" in some way. Significant drawbacks, however, are as outlined in the previous chapter: (1) a good initial guess must be provided to the solver, as the solver will find only a locally optimal solution; and (2) numerical difficulties, singularities, and nonconvexity may prevent the solver from converging to a solution. The generality of the approach means that it uses little information about the particular structure of the system to ensure convergence.

For convex systems, systems with particular structure [314], or particular choices of the control parameterization, it may be possible to demonstrate favorable convergence properties for nonlinear optimization. In general, however, there are no guarantees that nonlinear optimization will be able to find any solution, let alone a good solution, to a particular problem.

12.5.4 Gradient Methods for Driftless Systems

To improve the convergence properties of gradient-based motion planning, we focus on the class of driftless systems and give up on finding optimal motions. Let $u^p = [u_1^p, \ldots, u_r^p]^T \in \mathbb{R}^r$ be a finite parameterization of the control history $u(t)$, e.g., the coefficients of truncated Fourier series for the control inputs. Let the time of motion be $t_f = 1$, and define an end-state map f that maps the initial state x_{start} and the control u^p to a final state of the system x_f:

$$x_f = f(x_{\text{start}}, u^p)$$

The end-state map f is typically obtained numerically.

Now the problem is to find a u^p so that the desired goal state x_{goal} is reached. Define the end-state error vector $e = [e_1, \ldots, e_n]^T$ to be

$$e = f(x_{\text{start}}, u^p) - x_{\text{goal}}.$$

We would like to know the direction to change u^p to move x_f in the direction $-e$, to reduce the error. This direction in the u^p space is $v \in \mathbb{R}^r$, where

$$v^T = -e^T \left[\frac{\partial f(x_{\text{start}}, u^p)}{\partial u^p} \right] = -[e_1, \ldots, e_n] \begin{bmatrix} \frac{\partial f_1}{\partial u_1^p} & \cdots & \frac{\partial f_1}{\partial u_r^p} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1^p} & \cdots & \frac{\partial f_n}{\partial u_r^p} \end{bmatrix},$$

where the partial derivatives are evaluated at the current guess for u^p. Given a current guess $u^p(i)$, we can update it as follows:

$$u^p(i+1) = u^p(i) + \alpha v(i),$$

where α is a small positive constant, perhaps chosen by a line search to maximally decrease the error. We then calculate the new vector $v(i+1)$ for $u^p(i+1)$ and iterate until we reach an iteration k such that $\| f(x_{\text{start}}, u^p(k)) - x_{\text{goal}} \| < \epsilon$ for a small constant ϵ.

This algorithm is guaranteed to converge to a solution for a sufficiently small α if there are no state or control constraints and if $\partial f / \partial u^p$ is rank n everywhere. The rank condition means that any point in a sufficiently small neighborhood of $x_f = f(x_{\text{start}}, u^p)$ is reachable by a small change to u^p, indicating that it is possible to move the error vector e in any direction. Generically, if the system is STLC everywhere and we have a rich enough control parameterization u^p, the matrix $\partial f / \partial u^p$ will have rank n (figure 12.21).

If $\partial f / \partial u^p$ loses rank at $u^p(i)$, then there are one or more directions in which we cannot move the error vector e. Such a $u^p(i)$ is a singular control and could cause the algorithm to get stuck. In this case, we add a control to the end of the motion, where

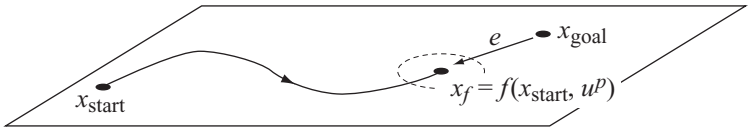

Figure 12.21 The initial state x_{start}, the state x_f reached after applying the control u^p, and the end-state error vector e. The full-rank condition on $\partial f/\partial u^p$ ensures that a neighborhood of x_f can be reached by a small variation in the control vector u^p.

the control is chosen randomly from the set of controls that result in no net motion of the system. Such a control is easy to generate because the system is driftless. Because such a control is generically nonsingular, it is said to generate a *generic loop* for the system. If we append the random control to the control u^p and treat the entire thing as the new control, the control vector is no longer singular, and the algorithm can continue.

There are many possible variants of this basic approach [129, 130, 391, 392, 400, 425]. For example, we could define a path on the error space from $e(0)$ to the origin, then choose the iterates $u^p(i)$ to force the error to track this path. We could use a metric on the error coordinates other than the standard metric implicit in the approach above. We could also perform a more sophisticated search, perhaps using the Hessian $\partial^2 f/(\partial u^p)^2$, to achieve faster convergence. Certain obstacle constraints can also be incorporated. This motion-planning method has been applied to find paths for a truck pulling trailers [129, 130, 425].

12.5.5 Differentially Flat Systems

Differentially flat systems [152, 153, 309] have a structure that makes motion planning (in the absence of control and configuration constraints such as obstacles) particularly simple. For a differentially flat system with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, there exists a set of m functions y_i, $i = 1 \ldots m$, of the state, the control, and its derivatives,

$$y_i\left(x, u, \dot{u}, \ldots, u^{(r)}\right), \quad i = 1 \ldots m,$$

such that the states and control inputs can be expressed as functions of y and its time-derivatives:

$$x = \phi\left(y, \dot{y}, \ldots, y^{(p)}\right)$$
$$u = \psi\left(y, \dot{y}, \ldots, y^{(p)}\right)$$

The functions y_i are known as the *flat outputs*. Armed with a set of flat outputs, the problem of finding a feasible trajectory $(x(t), u(t))$, $x(0) = x_{start}$, $x(t_f) = x_{goal}$, $t \in [0, t_f]$ for the underactuated system is transformed to the problem of finding a curve

$y(t)$ satisfying constraints on $y(0), \dot{y}(0), \ldots, y^{(p)}(0)$ and $y(t_f), \dot{y}(t_f), \ldots, y^{(p)}(t_f)$ specified by x_{start} and x_{goal}. In other words, the problem of finding a trajectory satisfying the underactuation constraints becomes the relatively simple algebraic problem of finding a curve to fit the start and end constraints on y. Any curve $y(t)$ maps directly to a consistent pair of state and control histories $x(t)$ and $u(t)$.

The flat outputs for mechanical systems are often a function of configuration variables only, and sometimes are just the location of particular points on the system. Unfortunately, there is no systematic way to determine if a system is differentially flat, or what the flat outputs for a system are. Many important systems have been shown to be differentially flat, however. These systems include the unicycle, the PBWT, and chained-form systems.

EXAMPLE 12.5.1 Unicycle (cont.) *The flat outputs for the unicycle are $y_1 = x_1$ and $y_2 = x_2$. The state x and controls u can be derived from the flat outputs and their derivatives as follows:*

$$[x_1, x_2, x_3]^T = \left[y_1, y_2, \tan^{-1} \frac{\dot{y}_2}{\dot{y}_1} \right]^T \tag{12.32}$$

$$[u_1, u_2]^T = \left[\pm\sqrt{\dot{y}_1^2 + \dot{y}_2^2}, \frac{\dot{y}_1 \ddot{y}_2 - \ddot{y}_1 \dot{y}_2}{\dot{y}_1^2 + \dot{y}_2^2} \right]^T . \tag{12.33}$$

The orientation x_3 and the turning control u_2 are not well defined as a function of the flat outputs when the unicycle is not translating. Also, because x_3 will be 45 degrees for both $\dot{y}_1 = \dot{y}_2 = 1$ and $\dot{y}_1 = \dot{y}_2 = -1$, the sign of the forward-backward speed u_1 should be consistent with the angle x_3.

Now we would like to find a feasible trajectory from $x_{\text{start}} = [0, 0, 0]^T$ to $x_{\text{goal}} = [1, 1, 0]^T$. Since there are six state variables in the specification of the start and goal points, there are six constraints on the flat outputs y and their derivatives at the beginning and end of motion. These constraints can be written

$$\begin{array}{ll} y_1(0) = 0 & y_2(0) = 0 \\ y_1(t_f) = 1 & y_2(t_f) = 1 \\ & \dot{y}_2(0) = 0 \\ & \dot{y}_2(t_f) = 0, \end{array}$$

where the last two constraints indicate that the initial and final motion of the unicycle must be along the x-axis, indicating that the wheel is oriented with the x-axis. The simplest polynomial functions of time that have enough free coefficients to satisfy these constraints are

$$y_1(t) = a_0 + a_1 t$$
$$y_2(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3.$$

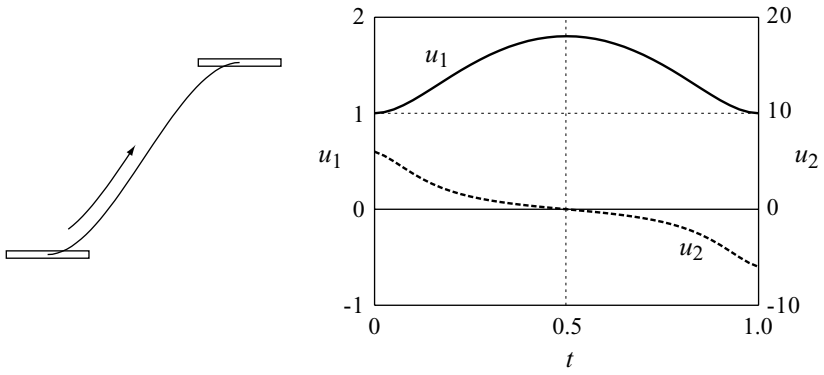

Figure 12.22 A feasible path for the unicycle from $[0, 0, 0]^T$ to $[1, 1, 0]^T$ and the controls.

Setting the time of motion $t_f = 1$ and using the constraints to solve for the polynomial coefficients, we get

$$y_1(t) = t$$
$$y_2(t) = 3t^2 - 2t^3.$$

The state $x(t)$ and control $u(t)$ can be obtained from equations (12.32) and (12.33). The unicycle motion is shown in figure 12.22.

In fitting a curve $y(t)$, we must choose a family of curves with enough degrees of freedom to satisfy the initial and terminal constraints. We may choose a family of curves with more degrees of freedom, however, and use the extra degrees of freedom to, individually or severally, (1) satisfy bounds on the control $u(t)$, (2) avoid obstacles in the configuration space, or (3) minimize a cost function. Incorporating these conditions in the calculation of $y(t)$ typically requires resorting to numerical optimization methods, and is a topic of current research. A good way to generate an initial guess for the optimization is to solve exactly for a minimal number of coefficients to satisfy the initial and terminal constraints, setting the other coefficients to zero.

For the PBWT, the flat outputs are

$$y_1 = x_1 + \frac{1}{d}\cos x_3$$
$$y_2 = x_2 + \frac{1}{d}\sin x_3.$$

The flat outputs (y_1, y_2) define a point fixed to the PBWT, at the PBWT's center of percussion with respect to the location of the thrusters.

For a car with steerable front wheels and parallel, fixed-orientation rear wheels, the flat outputs are the Cartesian coordinates of the point halfway between the rear wheels. If the car is towing a two-wheel trailer hitched midway between the rear wheels of the car, the flat outputs are the coordinates midway between the wheels of the trailer. If there are more trailers, all hitched midway between the wheels of the trailer in front, the coordinates of the midpoint of the wheels of the last trailer are flat outputs for the entire system. The orientation of the car and each trailer can be determined from sufficiently high time-derivatives of the evolution of these two coordinates.

The paper by Martin et al. [309] provides a catalog of systems known to be flat. Some notable results include the following. A system of the form (12.6) with n states and $m = n - 1$ inputs is flat if it is STLC. A driftless system (12.6) with n states and $m = n - 2$ inputs is flat if it is STLC. All chained-form systems are flat with the first and last states x_1 and x_n as the flat outputs. Other example flat systems are given in [331].

12.5.6 Cars and Cars Pulling Trailers

From a motion-planning perspective, easily the most heavily studied examples of nonholonomic systems are the kinematic car and the car pulling one or more trailers. Because of the obvious applications of automatic motion planning to systems like these, a great deal of effort has been spent in deriving efficient and complete motion planners for these systems moving in cluttered environments. The excellent book edited by Laumond [266] is focused entirely on these systems. In this subsection, we provide a brief review of important concepts and approaches to motion planning for cars and cars pulling trailers.

Cars

We focus on driftless kinematic models of cars, where the inputs are velocities and the state x is simply the configuration q. Alternatively, we could consider dynamic extensions of these models, where the inputs are accelerations.

A kinematic model of a standard car is shown in figure 12.23. The location of the point midway between the rear wheels is (x_1, x_2), the steering angle is x_3, and the orientation of the car is x_4. To ensure that the wheels do not slip, each of the front wheels must be perpendicular to the line through the wheel and the rotation center of the car. Therefore, x_3 is measured at a "virtual" wheel midway between the two front wheels. The wheelbase is L.

If the control u_2 is the rate of change of the steering angle x_3 and u_1 is the driving speed of the car, measured at the point midway between the rear wheels, the control

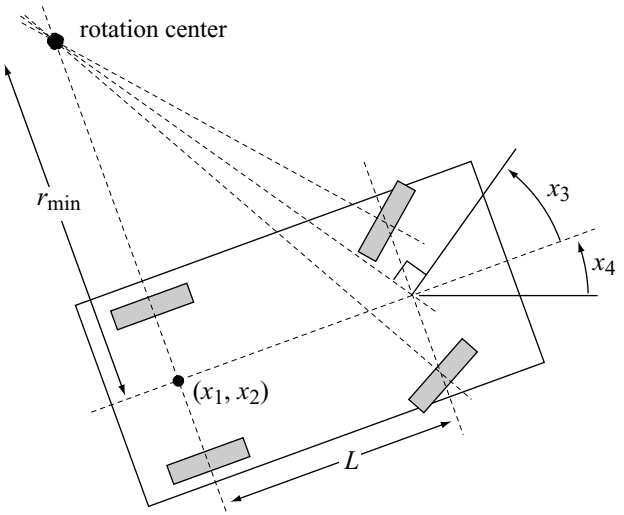

Figure 12.23 A model of a car turning at its minimum turning radius.

system can be written

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2, \quad g_1(x) = \begin{bmatrix} \cos x_4 \\ \sin x_4 \\ 0 \\ \frac{1}{L}\tan x_3 \end{bmatrix}, \quad g_2(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{12.34}$$

Typically the steering angle is limited to $-\gamma < x_3 < \gamma, \gamma \in (0, \pi/2)$, giving the car a minimum turning radius $r_{\min} = L/\tan^{-1}\gamma$. The control set for this system is $\mathcal{U} \in \mathcal{U}_{\pm}$, and this symmetric system can be shown to be STLC by the LARC. Since the car is STLC at all configurations, the steering angle limits do not affect the reachable positions and orientations of the car. Interestingly, this means that a car that cannot turn right, for instance, can reach any position and orientation among obstacles that a fully functional car can.

Since we are primarily concerned with the position and orientation of the body of the car, we could decide to eliminate the steering angle x_3 from the representation of the configuration and treat the steering angle as part of the control. Consider the modified control inputs (v, ω), where v is the linear speed of the car and ω is the angular velocity of the body of the car. In this case, the control system becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \cos x_4 \\ \sin x_4 \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{12.35}$$

which is identical to the unicycle, except that the control set satisfies the turning radius constraint $|\omega| < |v/r_{\min}|$. This does not affect the symmetry of the system, however, so it is still STLC.

As with the unicycle, if the car is limited to driving forward only, then it is globally controllable (in the absence of obstacles) but not STLC. In this section we will focus only on STLC models of cars.

Another car model of interest is the *differential-drive* car. In this case, the front wheels are replaced by casters that roll freely in any direction. The rear wheels are parallel and their speeds are controlled independently. If the speeds of the two rear wheels are u_1 and u_2, and the configuration of the car is $[x_1, x_2, x_4]^T$ as in figure 12.23, the control system can be written

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\cos x_4 \\ \frac{1}{2}\sin x_4 \\ \frac{1}{L} \end{bmatrix} u_1 + \begin{bmatrix} \frac{1}{2}\cos x_4 \\ \frac{1}{2}\sin x_4 \\ \frac{-1}{L} \end{bmatrix} u_2.$$

With the input transformation $v = (u_1 + u_2)/2$ and $\omega = (u_1 - u_2)/L$, the system again becomes the unicycle. The major difference from the standard car model is the lack of a turning radius constraint.

Small-time local controllability for these car models implies the following important consequence: if there is a free path from q_{start} to q_{goal} for the car body moving *without any nonholonomic motion constraint,* i.e., moving as a free-flying planar body, and if there is an open set of free space about each configuration in the path, then there is a free path for the car with the motion constraint. Stated equivalently, if $\mathcal{Q}_{\text{free}}$ is connected and open so that every $q \in \mathcal{Q}_{\text{free}}$ has a neighborhood of free space, then there exists a path for the car from any $q_{\text{start}} \in \mathcal{Q}_{\text{free}}$ to any $q_{\text{goal}} \in \mathcal{Q}_{\text{free}}$. This implies that it is possible to parallel-park your car into any parking space $\epsilon > 0$ longer than your car. However, the number of direction reversals in your parking maneuver grows proportionally to $1/\epsilon^2$ [266], so you could be there a while if ϵ is small!

Let's turn our focus to the simplified standard car model [equation (12.35)] with a limited turning radius. One question that has received considerable attention is the following: Given a start and goal configuration for the car moving in an obstacle-free space, find the path that minimizes the arclength traveled by the point midway between the rear wheels. If we assume a bound on the linear velocity $|v| < v_{max}$, and no acceleration limits, then this path also corresponds to the minimum-time motion. This problem has been solved owing in large part to contributions by Reeds and Shepp [360], Sussmann and Tang [402], and Souères and Laumond [395]. See also the chapter by Souères and Boissonnat [394].

Reeds and Shepp [360] showed that the optimal path must be one of a discrete and computable set of paths. It turns out that each member of this set consists of a

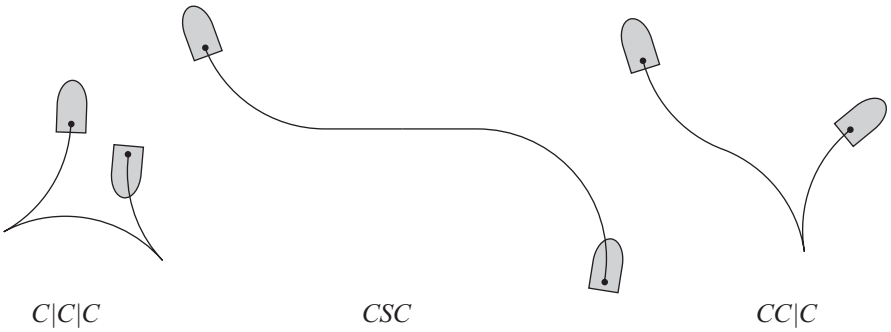

Figure 12.24 Three Reeds-Shepp curves.

concatenation of straight-line segments and circular arcs at the car's minimum turning radius. If C indicates a circular arc segment, S indicates a straight-line segment, and | indicates a cusp in the motion where the linear velocity v changes sign, the optimal path is guaranteed to be contained in the following set of path types:

$$\{C \mid C \mid C,\ \ CC \mid C,\ \ C \mid CC,\ \ CC_a|C_aC,\ \ C \mid C_aC_a \mid C,$$
$$C \mid C_{\pi/2}SC,\ \ CSC_{\pi/2} \mid C,\ \ C \mid C_{\pi/2}SC_{\pi/2} \mid C,\ \ CSC\}$$

The subscript a indicates an arc of angle a. One or more of the segments may be zero length. Figure 12.24 illustrates three Reeds-Shepp curves.

In the absence of obstacles, we can simply look up the optimal path from the set above using a map indexed by the goal configuration relative to the initial configuration [394, 395]. Shortest paths may not be unique. Analogous results for time-optimal motions of a differential-drive car with wheel velocity limits were derived by Balkcom and Mason [35].

The following motion-planning methods apply to the case of carlike robots in the presence of obstacles.

Grid Search

Barraquand and Latombe [41] developed a simple planner for cars moving in a bounded (typically rectangular) subset of the plane. Define six actions for the car to be $L_\pm$, $R_\pm$, and $S_\pm$, for the steering wheels turned all the way to the left, turned all the way to the right, or pointed straight ahead, with the subscripts "+" and "−" indicating forward and reverse velocity, respectively. Pseudocode is given in algorithm 22, CAR_GRID_SEARCH.

Algorithm 22 CAR_GRID_SEARCH

Input: Start configuration q_{start}, goal region $\mathcal{G}(q_{\text{goal}})$
Output: A path from q_{start} to $\mathcal{G}(q_{\text{goal}})$ or FAILURE

```
Initialize search tree T and list OPEN with start configuration q_start
while OPEN not empty and size(T) < MAXTREESIZE do
   q ← first in OPEN, remove from OPEN
   if q ∈ G(q_goal) then
      Report SUCCESS, return path
   end if
   if q is not near a previously occupied configuration then
      Mark q occupied
      for all actions in {L±, R±, S±} do
         Integrate forward a fixed time to q_new
         if path to q_new is collision-free then
            Make q_new a successor to q in T
            Compute cost of path to new configuration q_new
            Place q_new in OPEN, sorted by cost
         end if
      end for
   end if
end while
Report FAILURE
```

Conceptually, the planner keeps a tree T of configurations reached in the search and a list *OPEN* of pointers to configurations in T whose successors have not yet been generated. The pointers in the list *OPEN* are sorted by the costs of the paths to the associated configurations. The planner begins by making the car's initial configuration q_{start} the root of the tree T and initializing the list *OPEN* with a pointer to this configuration. The main loop of the planner is a simple best-first search. The planner sets the current configuration to that indicated by the minimum-cost pointer in *OPEN,* and it removes this pointer from *OPEN*. Subsequent configurations are generated by integrating each motion forward a short time, and each new collision-free configuration is added to the tree T with a record of the motion taking it there as well as a pointer to the previous configuration. Pointers to these new configurations are then inserted into the sorted list *OPEN*. This continues until one of the termination conditions is satisfied: (1) the list *OPEN* becomes empty, or the number of nodes in the tree T exceeds some user-specified value, indicating failure of the search; or (2) the planner reaches a configuration in a user-specified neighborhood $\mathcal{G}(q_{\text{goal}})$ of the

goal configuration q_{goal}. Note that the planner is not exact, as it only finds a path to a goal neighborhood.

The cost of a path is a function of the number of motion steps, the number of changes in the steering direction, and the number of cusps (switches between forward and reverse linear velocity). For example, for positive weighting factors a, b, and c, the cost could be a times the number of motion steps plus b times the number of steering changes plus c times the number of reversals. For $b = c = 0$, the planner will find short paths, and for $a = b = 0$, the planner will minimize the number of cusps (see figure 12.25).

By choosing the weighting factors a, b, c to be non-negative integers, inserting pointers into the sorted list *OPEN* can be done in constant time. This is accomplished by defining a one-dimensional array with cost as the index. The configurations of cost C, then, are stored in a linked list pointed at by element C of the array. To insert a configuration of cost C into *OPEN,* we simply append it to the end of the linked list pointed at by element C of the array.

The planner discards paths that are not collision-free. For a polygonal car and obstacles, collision detection can be done exactly by recognizing that all points on

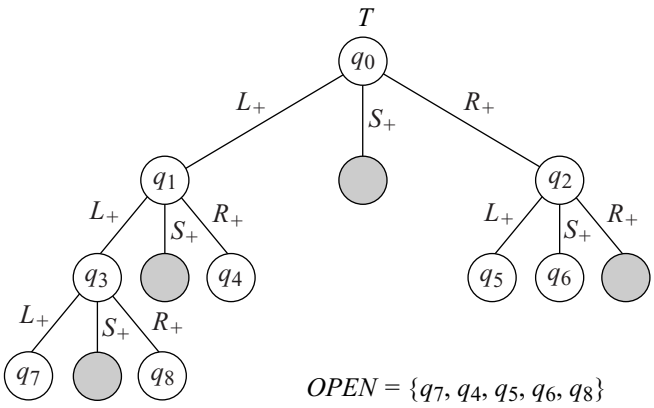

Figure 12.25 An example search tree T and list *OPEN*. (To keep the tree size reasonable, this search uses only the actions L_+, S_+, R_+.) If the path to a configuration is not free, the node is marked gray. In this example, the cost of each motion is 1, and the cost of changing steering direction is 5, to penalize excessive steering. This explains the ordering of the nodes in the sorted list *OPEN*. The cost of the path to q_7 is 3, the cost of paths to q_4, q_5, and q_6 is 7, and the cost of the path to q_8 is 8. Therefore, the planner will next remove q_7 from *OPEN* and generate its children in T. The configuration q_8 lies in the goal region, but the planner will not check this until q_8 pops off the front of the list *OPEN*. This is to ensure finding the minimum-cost path.

the car move in circular arcs or straight lines. Obstacle intersection then becomes a problem of intersecting arcs or line segments with line segments. A simpler approach is to surround the car by a disk and to only check for collision of the disk at the end of a motion step. The disk should be chosen large enough to guarantee that collision detection is *conservative*—only feasible plans will be found, but feasible paths through tight spaces could be rejected.

The planner also discards any configuration that is sufficiently close to a configuration from which the children have already been generated. Two configurations are considered sufficiently close if they occupy the same cell of a predefined grid on the configuration space. The car is assumed to be confined to a rectangular region of the plane, so $q \in (0, x_{max}) \times (0, y_{max}) \times [0, 2\pi)$, and the configuration space grid contains d^3 boxes, where d is the number of partitions of each dimension of the configuration space.

The user must specify the parameters defining the size of the goal neighborhood $\mathcal{G}(q_{\text{goal}})$, the integration time of the control steps, and the resolution of the configuration space grid used to check for prior occupancy. These parameters are interdependent. The resolution of the grid should be sufficiently fine that the application of any control step will always move the configuration to a new grid cell, and the goal neighborhood should be large enough that the car will not easily jump over it. In practice, the user should decide how much configuration error is acceptable along each dimension at the goal configuration, choose d so that each grid box is no larger (and usually somewhat smaller) than the goal region, and then choose the control step to be just long enough to guarantee that the car will exit its current grid box.

This planner is resolution complete, meaning that if the step size is sufficiently small, the planner will find a path if one exists. Because the planner uses a best-first search, the path found will be optimal for the user's cost function and the given step size. This planner actually runs faster in cluttered spaces because the obstacles prune the search tree.

One important property of the approach is that any path with p cusps (linear speed reversals) using the full motion capabilities of the car can be followed arbitrarily closely by a path with only p cusps using only motions with the steering wheel turned all the way to the left or right. This means that the discretization of the car's possible motion directions does not preclude the possibility of finding minimum-cusp motions.

A drawback to this planner is that it is not exact; paths found by the planner go to a neighborhood of the goal, not exactly to the goal. It can also be slow in open spaces due to the exponential growth of the search tree T. The maximum number of configurations that will be marked "occupied" is upper-bounded by the d^3 boxes of the occupancy grid, however, which may not be too large considering the system has only three degrees of freedom.

Examples of planner output are shown in figures 12.26 and 12.27.

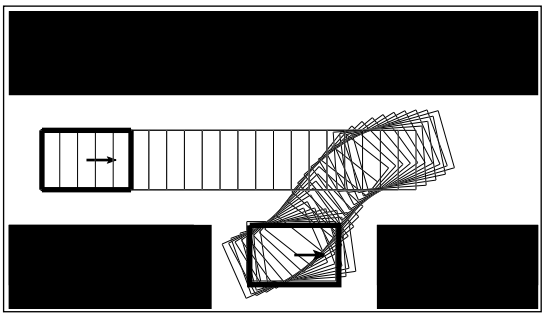

Figure 12.26 The classic parallel-parking problem.

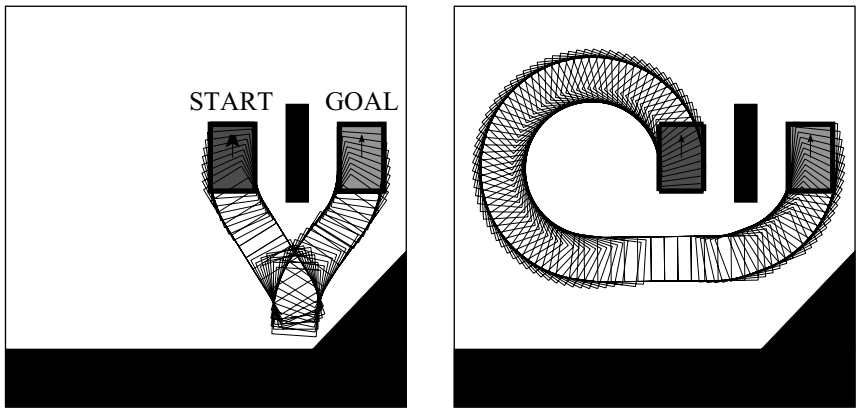

Figure 12.27 The same planning problem solved with two different cost functions. On the left, $a = 1$ and $b = c = 0$, meaning that the planner looks for the shortest path. On the right, $a = 1$, $b = 0$, and $c = 300$, meaning that the planner heavily penalizes cusps.

Omnidirectional to Nonholonomic Path Transformation

The following approach to motion planning for a car was proposed in [267]:

1. Use a path planner to find a path for a car with no motion constraints (i.e., a free-flying body).

2. Transform the path into a path satisfying the nonholonomic constraint.

3. Optimize the path.

Because the car is STLC, the path transformation in the second step is always possible if the path found in the first step does not touch any obstacles.

Step 2 proceeds as follows. Parameterize the original path returned in the first step by $s \in [0, 1]$, where $q(0)$ is the initial configuration and $q(1)$ is the final configuration. Using the lookup table of Reeds-Shepp curves, find the shortest path connecting $q(0)$ and $q(1)$. If this path is collision-free, then we have found the shortest path, and we are done. If it is not collision-free, divide the s interval $[0, 1]$ into two equal pieces, and calculate the shortest paths between $q(0)$ and $q(\frac{1}{2})$, and between $q(\frac{1}{2})$ and $q(1)$. If either of these paths is in collision, subdivide that interval again, and continue recursively until a free path is found. This procedure is guaranteed to terminate if the path found in the first step touches no obstacles. This guarantee relies on a topological property that says for any open ball $B_\delta(q)$ of radius $\delta > 0$ about a configuration q, there exists another ball $B_\epsilon(q)$ such that for any $q' \in B_\epsilon(q)$, the local path planner (Reeds-Shepp curves in this case) finds a path between q and q' that is contained in $B_\delta(q)$ (figure 12.28).

We now have a feasible path for the car, but it may be unnecessarily long. In the final step, we choose two randomly selected points along the path and replace the path in between by the shortest Reeds-Shepp path, if it is collision-free. We iterate this procedure, stopping when it fails to shorten the path a prespecified number of times in a row.

Randomized Methods

Probabilistic roadmap methods, as discussed in chapter 7, can also be applied to carlike robots. All that is required is the specification of a local planner that quickly finds a path connecting two configurations in the absence of obstacles. For example, the local planner could use Reeds-Shepp curves as described in [394, 395] to quickly find a shortest path. Two configurations are connected in the "roadmap" representation of

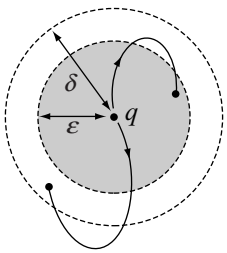

Figure 12.28 For a given open ball $B_\delta(q)$, there exists an open ball $B_\epsilon(q)$ such that paths found by the local planner from q to any point in $B_\epsilon(q)$ are completely contained in $B_\delta(q)$.

the free space if the path returned by the local planner is collision-free. This approach is probabilistically complete—the probability of finding a solution, if one exists, approaches 100% as the running time goes to infinity.

Smooth Paths

A drawback to using Reeds-Shepp curves, and in using the CAR_GRID_SEARCH planner (algorithm 22), is that the curvature of the paths is discontinuous at the transitions between straight and curved segments, even where there is no cusp. This means that either the steering angle must change instantaneously, the car must come to a stop at the transitions, or we must be willing to accept the error in execution that comes from ignoring this problem. To overcome this problem, several approaches have been proposed to planning paths with continuous steering angles. For example, a postprocessing step could be used to smooth the transitions (e.g., see [151]). A problem with this approach is that the new, smoothed path might collide with obstacles. Instead, smooth primitives can be used directly in the local planner, perhaps based on the car's differential flatness properties [159, 258, 351, 370].

Cars Pulling Trailers

Figure 12.29 shows a car pulling two trailers, with each trailer hitched between the rear wheels of the body in front. For a car pulling p trailers, the configuration of the system is written $x = [x_1, \ldots, x_{p+4}]^T$, where $x_{i+4} = \theta_i$ gives the orientation of the ith trailer relative to the body in front. The controls for this system are still u_1, the forward speed of the car, and u_2, the rate of change of the steering angle.

To derive the equations of motion for this system, let's start by looking at a single trailer (see figure 12.30). The trailer is being pulled with a linear velocity w at an

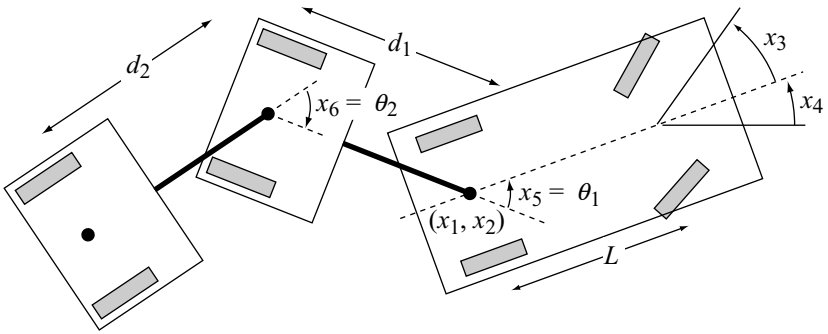

Figure 12.29 A car pulling two trailers.

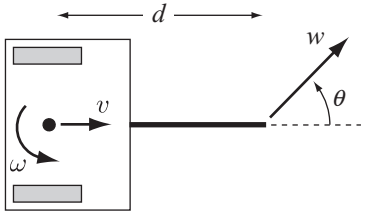

Figure 12.30 A single trailer.

angle θ at the hitch a distance d from the trailer wheels. The trailer's linear velocity is v and the angular velocity is ω. The trailer velocity (v, ω) is related to the pulling velocity w at the hitch by

$$v = w \cos\theta \tag{12.36}$$

$$\omega = \frac{w}{d} \sin\theta. \tag{12.37}$$

The linear velocity v becomes the hitch velocity w for the trailer behind this trailer, and equation (12.36) shows that the linear velocity is nonincreasing as we move back in the trailer chain.

Extending the reasoning above, the car and trailer system can be written

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2,$$

where

$$g_1(x) = \begin{bmatrix} \cos x_4 \\ \sin x_4 \\ 0 \\ \frac{1}{L}\tan x_3 \\ \frac{1}{d_1}\sin\theta_1 \\ \frac{1}{d_2}\sin\theta_2\cos\theta_1 \\ \frac{1}{d_3}\sin\theta_3\cos\theta_1\cos\theta_2 \\ \vdots \\ \frac{1}{d_p}\sin\theta_p\prod_{i=1}^{p-1}\cos\theta_i \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{12.38}$$

Constructing Lie brackets of g_1 and g_2 shows that the tractor-trailer system is STLC at any x for a symmetric control set [265].

To plan motions for a tractor-trailer system among obstacles, we could extend the grid search approach of Barraquand and Latombe [41]. For example, if the car is

pulling a single trailer, we can add a dimension to the configuration space grid and proceed as before with the six motion primitives $L_{\pm}$, $S_{\pm}$, and $R_{\pm}$. The only difference is that the equations of motion must be numerically integrated to determine the net change of the trailer orientation at the end of the motion step. Alternatively, this computation can be done offline and stored in a lookup table.

The path transformation and randomized approaches can also be extended to tractor-trailer systems by using local planners based on exact closed-form motion plans in the absence of obstacles [378, 406]. Such local planners may take advantage of the fact that all trailer systems of the form described above can be converted to chained form, allowing the use of sinusoidal controls [377, 393].

The path transformation method first finds a free path ignoring the nonholonomic constraints, and then transforms the path into one respecting the constraints. For the path transformation method to work, the local planner must satisfy the topological property discussed earlier for cars [259, 378]. A generalization of this approach turns this single transformation step into a sequence of steps, each one introducing one more nonholonomic constraint to be satisfied by the transformation [379]. This multilevel approach can lead to increased computational efficiency and shorter paths. Finally, a path can be turned into a trajectory by a time scaling $s(t)$ respecting actuator limits, as discussed in chapter 11 [259].

Figure 12.31, taken from [378], shows two paths found for a two-trailer system using a path transformation planner. The local planner used to transform the original path to a feasible path uses sinusoidal inputs and the two-trailer system's chained-form equations.

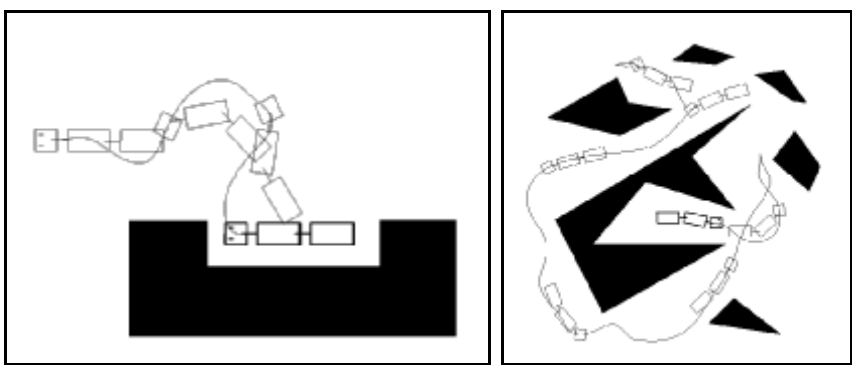

Figure 12.31 Two paths for a two-trailer system found using an omnidirectional-to-nonholonomic path transformation planner. (From Sekhavat and Laumond [378], ©1998 IEEE.)

12.5.7 Kinematic Reductions of Mechanical Systems

Subsection 12.4.2 described a class of underactuated second-order mechanical systems, called kinematically controllable systems, for which trajectory planning can be decoupled into path planning followed by time scaling according to actuator limits. The big advantage of this is that the search for a feasible motion plan can occur in the system's $n_{\mathcal{Q}}$-dimensional configuration space instead of the $2n_{\mathcal{Q}}$-dimensional state space. Since many search algorithms run in time exponential in the dimension of the search space, this reduction can greatly speed up motion planning.

We focus on systems that are not maximally reducible to kinematic systems but possess p decoupling vector fields $V_1, \ldots, V_p$ satisfying $\ldots \overline{\text{Lie}}(\{V_1, \ldots, V_p\})(q) = T_q\mathcal{Q}$ at all $q \ldots$, meaning that the system is STLKC. The path-planning problem is to find a concatenation of integral curve segments of these vector fields to take the system from the initial configuration q_{start} to the goal configuration q_{goal}. Because the velocity must be brought to zero at switches between vector fields, in the interest of minimizing execution time, it is reasonable to design the planner to minimize the number of switches. This implicitly minimizes the use of slow Lie bracket motions. Because the second step of the procedure time-optimally time-scales the motion along the planned path, the approach produces fast trajectories in a computationally efficient manner. (Global time-optimality is precluded because of our decoupling of the problem.)

An example of a kinematically controllable system is a 3R robot arm moving in a horizontal plane with the third joint unactuated and frictionless (figure 12.32). It

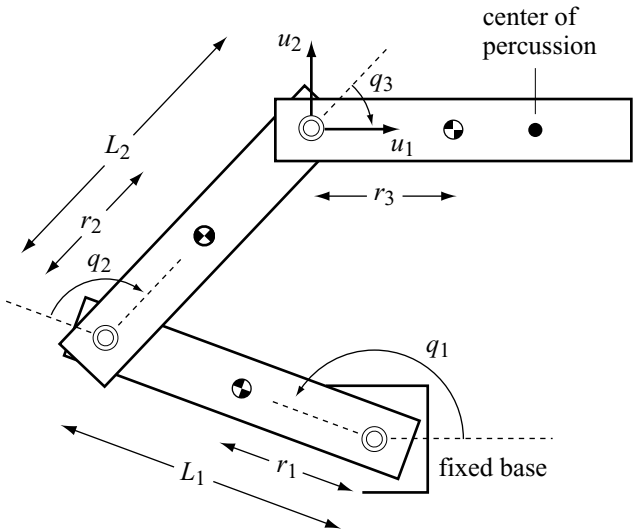

Figure 12.32 A 3R robot arm in a horizontal plane with an unactuated third joint.

is not hard to see that the third link of the arm is equivalent to the PBWT in zero gravity, except the "thruster" forces at the third joint are generated by the actuators at the first two joints. As long as the second joint is not completely extended, the arm can apply a force in any direction at the third joint. Because there is no actuator there, however, the arm cannot generate torque about this joint. The two decoupling vector fields for this system are translation along the length of the third link and rotation about the center of percussion of the third link with respect to the joint. Small-time local kinematic controllability of the PBWT implies STLKC of the 3R arm away from $q_2 = 0$ and $q_2 = \pi$.

To find motion plans minimizing the number of switches between these vector fields, we can adapt the grid search motion planner for cars, "driving" the third link around. In this problem, the four motion primitives are forward and backward translation and clockwise and counterclockwise rotation (about the center of percussion). We choose the path cost function to be the number of switches between the primitives. Inverse kinematics is used to calculate the robot's entire configuration as the third link moves, and collisions must be checked along the entire robot arm, not just at the third link. Apart from these modifications, the algorithm is the same as for carlike robots.

Once a path is found using the decoupling vector fields, the path can be time-scaled arbitrarily while respecting the underactuation constraint (zero torque at the third joint). To perform the time-optimal time scaling, we use the manipulator dynamics

$$M(q)\ddot{q} + \dot{q}^T \Gamma(q)\dot{q} = u = \begin{bmatrix} u_1 \\ u_2 \\ 0 \end{bmatrix}$$

in the time-scaling algorithm described in the previous chapter. For the 3R arm, the inertia matrix $M(q)$ is given by

$$\begin{aligned}
M_{11} &= I_1 + I_2 + I_3 + m_1 r_1^2 + m_2\left(L_1^2 + r_2^2 + 2L_1 r_2 \cos q_2\right) \\
&\quad + m_3\left(L_1^2 + L_2^2 + r_3^2 + 2L_1 L_2 \cos q_2 + 2L_2 r_3 \cos q_3 + 2L_1 r_3 \cos(q_2 + q_3)\right) \\
M_{12} &= I_2 + I_3 + m_2\left(r_2^2 + L_1 r_2 \cos q_2\right) \\
&\quad + m_3\left(L_2^2 + r_3^2 + L_1 L_2 \cos q_2 + L_2 r_3 \cos q_3 + L_1 r_3 \cos(q_2 + q_3)\right) \\
M_{13} &= I_3 + m_3\left(r_3^2 + L_2 r_3 \cos q_3 + L_1 r_3 \cos(q_2 + q_3)\right) \\
M_{22} &= I_2 + I_3 + m_2 r_2^2 + m_3\left(L_2^2 + r_3^2 + 2L_2 r_3 \cos q_3\right) \\
M_{23} &= I_3 + m_3\left(r_3^2 + L_2 r_3 \cos q_3\right) \\
M_{33} &= I_3 + m_3 r_3^2,
\end{aligned}$$

where m_i is the mass of link i, I_i is the inertia of link i about its center of mass, and r_i and L_i are defined in figure 12.32. Recall that $M_{21} = M_{12}$, $M_{31} = M_{13}$, $M_{32} = M_{23}$. The Christoffel symbols $\Gamma^i_{jk}(q)$ are derived from $M(q)$.

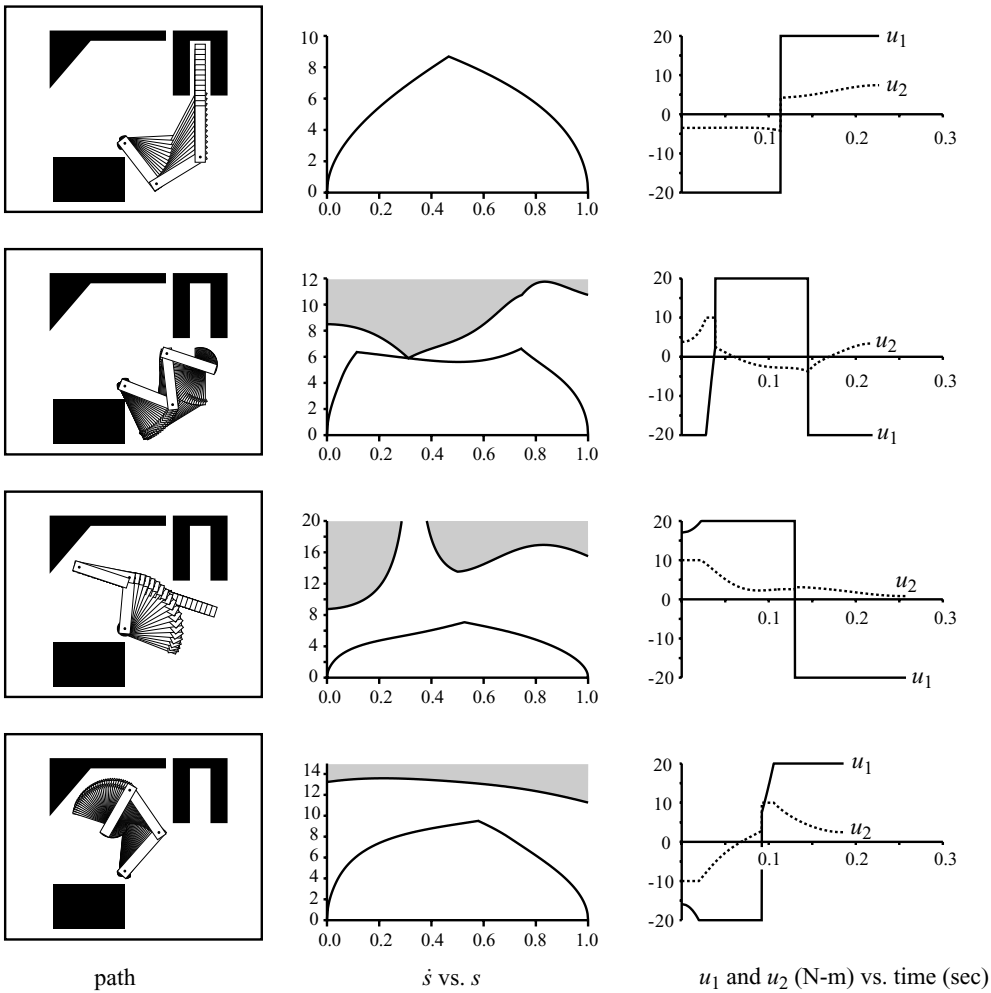

Figure 12.33 A collision-free trajectory for the 3R robot with an unactuated third joint. The left column shows the path and the right column shows the actuator profiles. One of the actuators is always saturated. The middle column illustrates the speed along the path segment vs. a path parameter (normalized rotation angle or translation distance), as described in chapter 11.

Joint i	L_i (m)	r_i (m)	m_i (kg)	I_i (kg-m^2)	$\tau_i^{\max}$ (N-m)	$\tau_i^{\min}$ (N-m)
1	0.3	0.15	2.0	0.02	20	-20
2	0.3	0.15	1.0	0.01	10	-10
3		0.15	0.5	0.004125	0	0

Table 12.1 Kinematic parameters, inertial parameters, and actuator limits for the simulated robot.

Figure 12.33 shows a trajectory planned for the robot arm with the kinematic parameters, inertial parameters, and actuator limits given in table 12.1. The path consists of four separate motions along decoupling vector fields, and the motion must come to a halt at the switches. Therefore, the time-scaling problem becomes four separate problems. The complete time-optimal trajectory along the path takes 0.890 s.

This planner has been successfully implemented on an experimental underactuated 3R arm, though not using the time-optimal motions along the paths [305].

A brute-force best-first search along decoupling vector fields can be applied to any STLKC mechanical system. For systems with more degrees of freedom, however, the computational expense may be prohibitive. In this case we might give up on finding motion plans that minimize switches between decoupling vector fields. Possible approaches are multiresolution grid-search methods, probabilistic roadmap methods, or transformation of omnidirectional paths (as described for cars in subsection 12.5.6) using exact local planners based on the decoupling vector fields [200,310], and rapidly-exploring random trees (RRT's) modified to reduce the number of vector field switches [111].

12.5.8 Other Approaches

Fictitious Inputs for Drift-Free Systems

Lafferriere and Sussmann proposed a general method for steering drift-free STLC systems [253, 254]. If the original system is

$$\dot{x} = \sum_{i=1}^{m} g_i(x) u_i,$$

then an *extended* system is defined to be

$$\dot{x} = \sum_{i=1}^{r} g_i(x) v_i, \ r \geq n,$$

where the vector fields $g_{m+1}, \ldots, g_r$ correspond to Lie product motions of the system. These vector fields are chosen so that $\text{span}(\{g_1, \ldots, g_r\})\ (x) = T_x\mathcal{M}$ at all x, and $v_{m+1}, \ldots, v_r$ are called *fictitious* inputs.

There are no nonholonomic constraints for the extended system, so motion planning for this system is identical to motion planning for an unconstrained system. Once we have found a path for the unconstrained system using the controls v, we transform it to a path for the original system with controls u. This transformation uses the Campbell-Baker-Hausdorff formula [330, 369] describing the motion generated by composing motions along two different vector fields, and is beyond the scope of this chapter. If the vector fields $g_1, \ldots, g_m$ are nilpotent, or *nilpotentizable* by a feedback transformation, the transformation provides an exact expression for the motion of the system with a finite number of Lie products of the two vector fields, and the transformation produces an exact motion plan. Otherwise, small errors are introduced due to higher-order terms in the Lie bracket motion prescription. These errors can be arbitrarily reduced by iterating the procedure.

This approach applies to any STLC drift-free system. The quality of the solution depends on the initial path chosen for the system, and it is in no sense optimal. For more details, see the original papers by Lafferriere and Sussmann [253, 254] or the summaries in the textbooks [330, 368].

Motion Libraries

A motion library consists of a set of canonical motions or primitives that are feasible for the underactuated system, along with a set of conditions (or transition maneuvers) for concatenating these primitives. A search for a feasible trajectory is then restricted to compositions of these primitives. The decoupling vector fields of kinematically controllable systems are examples of motion primitives that are concatenable at any configuration q. As another example, a set of primitives for an airplane might include flying level, a steady dive, and a constant climbing turn. Symmetries in the system dynamics can be exploited to minimize the number of primitives; e.g., the dynamics of flying level (in the absence of wind) are invariant to the airplane's position and orientation in a horizontal plane. The library should consist of a sufficient number of primitives to ensure controllability of the system. One formalization of these ideas is given by the Maneuver Automaton of Frazzoli, Dahleh, and Feron [160].

Rapidly Exploring Random Trees and Expansive Space Trees

RRTs and ESTs, as described in chapter 7, apply to a broad class of systems, including nonholonomic and underactuated systems. All that is required is a state equation

$\dot{x} = f(x, u)$ and a distance metric appropriate to the problem. Because no particular structure of the system is utilized, motion plans may be inefficient. The planning time may be sensitive to the chosen distance metric. (See also chapter 7, section 7.5.1.)

Problems

1. Choose a grid of points on $\mathbb{R}^2$ and sketch the tangent vectors of the vector field $[x_2, x_1^2]^T$ at those points. You may draw these with a computer if you wish. Sketch by hand or use a computer to draw a few integral curves of this vector field. Does this vector field define a regular one-dimensional distribution?

2. For the vector fields $g_1 = [x_1 + x_2, 0]^T$ and $g_2 = [0, 1 + x_2]^T$ on $\mathbb{R}^2$, sketch the integral manifolds, or foliation, defined by the distribution span($\{g_1, g_2\}$). (See figure 12.9 for a drawing of a foliation.) Is the distribution regular?

3. For the vector fields $g_1 = [1, 0, 0]^T$ and $g_2 = [0, 1, 0]^T$ on $\mathbb{R}^3$, sketch the foliation defined by the distribution span($\{g_1, g_2\}$). Is the distribution regular? Is it involutive?

4. For the vector fields $g_1 = [x_3, 0, 0]^T$ and $g_2 = [0, 1, 0]^T$ on $\mathbb{R}^3$, sketch the foliation defined by the distribution span($\{g_1, g_2\}$). Is the distribution regular? Is it involutive?

5. For the vector fields $g_1 = [1+x_3, 1-x_2, 0]^T$ and $g_2 = [0, 0, 1]^T$ on $\mathbb{R}^3$, sketch the foliation defined by the distribution span($\{g_1, g_2\}$). Is the distribution regular? Is it involutive?

6. For the vector fields $g_1 = [1+x_3, 1-x_3, 0]^T$ and $g_2 = [0, 0, 1]^T$ on $\mathbb{R}^3$, sketch the foliation defined by the distribution span($\{g_1, g_2\}$). Is the distribution regular? Is it involutive?

7. Describe physical systems with the following properties, not using the examples discussed in the chapter: accessible but not STLA; accessible but not controllable; controllable but not STLA; STLA but not STLC; STLC but not controllable.

8. For vector fields that are *linear* in the state, e.g., $g_1(x) = Ax$ and $g_2(x) = Bx$, the Lie bracket has the particularly simple form

(12.39) $$[g_1, g_2](x) = (AB - BA)x,$$

called the *matrix commutator* of A and B. For such vector fields, the Lie bracket can be calculated without differentiation. As an example, let $x = [x_1, x_2, x_3]^T$ and

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix},$$

and let $g_1 = Ax$ and $g_2 = Bx$. Use equation (12.5) to show $[g_1, g_2](x) = (BA - AB)x$, the negative of the expression in equation (12.39). (Note that the sign of the Lie bracket is immaterial in generating the distribution.)

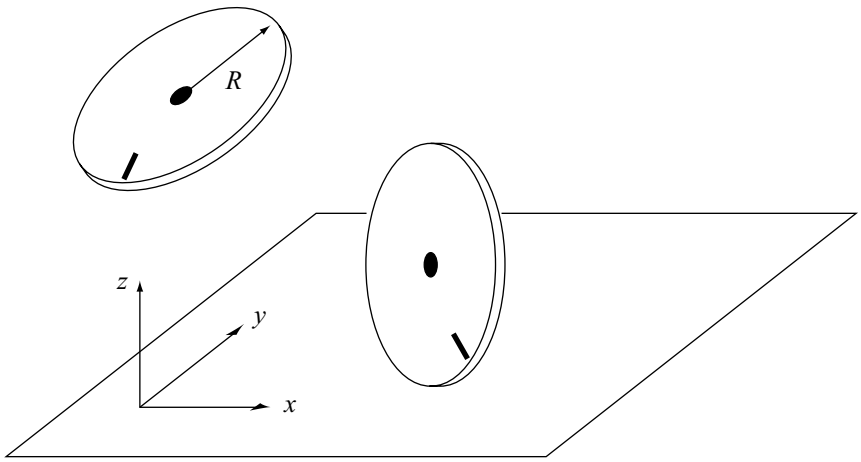

Figure 12.34 A wheel in space, then constrained to stand upright on the $z = 0$ plane. The notch on the wheel is used to keep track of the wheel's rolling angle.

9. Write Mathematica (or other symbolic software) code to take two vector fields and calculate their Lie bracket.

10. The configuration of a wheel of radius R has six degrees of freedom in three-dimensional space, described globally as $SE(3)$ or locally using x-y-z and roll-pitch-yaw coordinates. Choose six coordinates to describe the wheel's configuration in space, where the x-y-z coordinates describe the position of the center of the wheel. With these six coordinates, write two holonomic constraints that constrain the wheel to stand upright on a plane at $z = 0$ (figure 12.34). If you choose your coordinates properly, this will leave you with four coordinates to describe the configuration of the wheel on the plane. Using the time-derivatives of these coordinates, write the two nonholonomic constraints that prevent slipping at the contact between the plane and the wheel as it moves.

    This system is identical to the unicycle example in this chapter, except the configuration space is four-dimensional (the rolling angle of the wheel is included). If the two controls are the rolling angular velocity of the wheel and the turning-in-place angular velocity of the wheel, write the two control vector fields, and write the system as a control-affine nonlinear control system. Using Lie brackets of the vector fields, show that the system is STLC at any configuration if the control set belongs to $\mathcal{U}_{\pm}$.

11. For the wheel of problem 10, describe a four-step motion-planning algorithm to take the unicycle to an arbitrary configuration in its obstacle-free four-dimensional configuration space. The final step of the algorithm should drive the wheel around a circle to achieve the desired rolling angle. Your algorithm should take the final configuration as input (assuming the wheel starts from the origin configuration $[0, 0, 0, 0]^T$) and return a sequence of control values u_1 and u_2 and the times they are applied.

12. Transform the control system of problem 10 to chained form. This may be challenging!

13. Prove that the standard car model of equation (12.34) is STLC.

14. Derive the drift and control vector fields for the PBWT, assuming that it has mass m and inertia I about the center of mass. Then set $m = I = 1$ and verify that your vector fields match those given in the text.

15. Because the PBWT has three degrees of freedom but only two controls, there is a constraint on its possible accelerations. Derive this constraint, and show that it can be written in the form $\omega(x)\dot{x} = 0$ in the absence of gravity.

16. Imagine a PBWT where the control u_2 is a pure torque about the center of mass. Write the two control vector fields, put the system in the control-affine form (12.6), and use theorem 12.3.3 to show that it is STLC at zero velocity in the absence of gravity. Then put the system in the covariant derivative form of equation (12.13) and use theorem 12.4.1 to prove the same thing.

17. Flat outputs for the PBWT are $y_1 = x_1 + \frac{1}{d}\cos x_3$ and $y_2 = x_2 + \frac{1}{d}\sin x_3$. Find the maps ϕ and ψ to recover the states $x(t)$ and control inputs $u(t)$ as a function of the trajectory of the flat outputs $y(t)$.

18. Flat outputs for the car pulling trailers, described in subsection 12.5.6, are the two coordinates describing the planar location of the point midway between the two wheels of the last trailer. Find the maps ϕ and ψ to recover the states $x(t)$ and control inputs $u(t)$ as a function of the trajectory of the flat outputs.

19. Let u_1 and u_2 be the torques at the two joints of a 2R robot arm in a horizontal plane (figure 12.35). Write the dynamics of the 2R arm in the form of equation (12.6), where the masses of the first and second links are m_1 and m_2, and the inertias of the links about their centers of mass are I_1 and I_2. Because the drift vector field g_0 is energy-conserving and the arm configuration space $S^1 \times S^1 = T^2$ is compact, the drift vector field is WPPS. If possible, use theorem 12.3.5 to show that the robot arm is (globally) controllable with $u_2 = 0$; in other words, any state is reachable from any other state by using only torques at the first joint. If you cannot, explain whether you believe the arm is controllable or not, and how you might demonstrate your belief. Why is the arm not controllable if $u_2 \in \mathbb{R}$ but $u_1 = 0$?

20. Let $[u_1, u_2, u_3]^T$ be the torques at the three joints of the 3R robot arm of figure 12.32 in a horizontal plane. Write the dynamics of the 3R arm in the form of equation (12.6).

    In subsection 12.5.7, a motion planner is described for the underactuated 3R arm with $u_3 = 0$. In this case, there are two decoupling vector fields, or two rank 1 kinematic reductions. If instead only the first actuator is missing, so $u_1 = 0$, there is a rank 2 kinematic reduction—the system is maximally reducible to a kinematic system. This means that the acceleration constraint due to $u_1 = 0$ can actually be integrated to a velocity constraint: the total angular momentum about the first joint is conserved. Assuming that the 3R robot arm begins at rest, write this velocity constraint and give the rank 2 kinematic reduction in the form of equation (12.15).

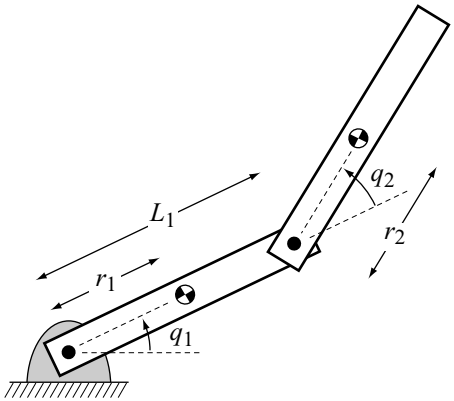

Figure 12.35 A 2R robot arm.

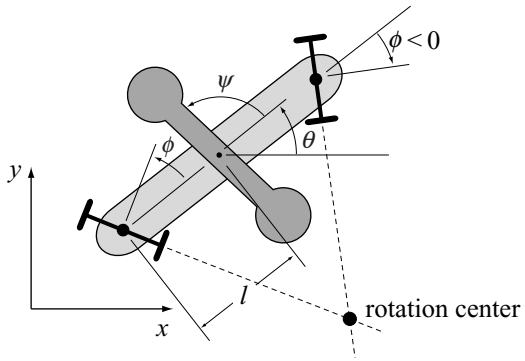

Figure 12.36 The snakeboard model.

21. The dynamics of an RP manipulator are derived at the beginning of chapter 10. Set gravity a_g to zero, and assume that the robot is missing the actuator at the prismatic joint, so $u_2 = 0$. Find the input vector field $Y_1(q)$ and use theorem 12.4.1 to show that the system is STLA when $q_2 > 0$ using only the actuator at the revolute joint. Also, provide an argument either supporting or rejecting the hypothesis that the arm is globally controllable on its state space.

22. The *snakeboard* is a commercial toy whose concept is derived from the well known skateboard. It is composed of two steerable wheeled platforms joined by a coupling bar, and the rider propels herself forward without touching the ground by steering the wheels and twisting her body back and forth. A simple model of the snakeboard is shown in figure 12.36. Here a momentum rotor simulates the rider by spinning back and forth, and by conservation of angular momentum about the rotation center chosen by the wheels, the snakeboard body

moves. The snakeboard model is an underactuated mechanical system with nonholonomic constraints, which we will write in the form of equations (12.18) and (12.19).

Let the configuration of the snakeboard be represented by $q = [x, y, \theta, \psi, \phi]^T$, where (x, y) represents the Cartesian position of the center of the snakeboard coupler, θ is its angle, and ψ and ϕ are the angle of the rotor and the steering angle of the wheels, respectively, expressed in the body frame. The inertia matrix for the snakeboard is given by

$$M = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & I + I_r + I_w & I_r & 0 \\ 0 & 0 & I_r & I_r & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix},$$

where m is the total mass of the snakeboard, I is the inertia of the coupler about its center of mass, I_r is the rotor inertia, and $\frac{1}{2}I_w$ is the inertia of each set of wheels about its pivot point. (Note that because the inertia matrix is invariant to the configuration, the Christoffel symbols are zero.) The system is subject to two control inputs: a torque u_ψ that controls the rotor angle ψ, and a torque u_ϕ controlling the steering angle ϕ. Therefore $u = [u_\psi, u_\phi]^T$ and $T(q)$ can be written

$$T(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The wheels are assumed to roll without lateral slipping, and the wheel angle chooses a rotation center along a line perpendicular to the body of the snakeboard and through its center. The no-slip constraints can be manipulated into the form

$$A(q) = \begin{bmatrix} \sin\phi & 0 & -l\cos\theta\cos\phi & 0 & 0 \\ 0 & \sin\phi & -l\sin\theta\cos\phi & 0 & 0 \end{bmatrix}.$$

Write the equations of motion in the form of equations (12.18) and (12.19). Find the projection matrix $P(q)$ and the two input vector fields $\widetilde{Y}_1(q)$ and $\widetilde{Y}_2(q)$. Show that these two vector fields are decoupling vector fields and that the system is STLKC by these two decoupling vector fields. Explain what this means in terms of motion planning for this system.

23. Implement the grid search path planner CAR_GRID_SEARCH (algorithm 22).

24. For a differential-drive car, include the drive wheel angles in the description of the configuration, giving the car a five-dimensional configuration space (position and orientation of the body and two wheel orientations). Write out the control system and prove that this system is or is not STLC on this five-dimensional space.

25. Prove that all chained-form systems are differentially flat with the first and last states x_1 and x_n as flat outputs, and describe a method for finding the mappings ϕ and ψ from the flat outputs and their derivatives to the state x and the control u, respectively.

26. In example 12.4.4, perform the calculations to verify $\langle Y_1 : Y_1 \rangle$, $\langle Y_2 : Y_2 \rangle$, $\langle Y_1 : Y_2 \rangle$, and $[Y_1, Y_2]$. You may write symbolic manipulation code (e.g., in Mathematica) to do these computations for you.

27. Derive the equations of motion of the hopper in example 12.4.4 in the form of equation 12.7, and show that the underactuation implies an acceleration constraint that can be integrated to give a conservation of angular momentum constraint.

28. Write the Pfaffian constraint for the hopper in example 12.4.4 and give a driftless kinematic model of the system (the rank 2 kinematic reduction).

29. Derive the formula for the Lie bracket

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2.$$

In other words, show that the net motion obtained by following g_1 for time ϵ, g_2 for time ϵ, $-g_1$ for time ϵ, and $-g_2$ for time ϵ is

$$\epsilon^2 \left(\frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2 \right) + O(\epsilon^3),$$

where ϵ is small. To do this, perform a Taylor expansion to express the net motion, throwing away terms of higher order than ϵ^2. For example, after following g_1 for time ϵ, we have

$$x(\epsilon) = x(0) + \epsilon \dot{x}(0) + \frac{1}{2}\epsilon^2 \ddot{x}(0) + O(\epsilon^3).$$

Subsituting $g_1(x(0)) = \dot{x}(0)$ and $\frac{\partial g_1}{\partial x} g_1(x(0)) = \ddot{x}(0)$, where $\frac{\partial g_1}{\partial x}$ is evaluated at $x(0)$, we get

$$x(\epsilon) = x(0) + \epsilon g_1(x(0)) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial x} g_1(x(0)) + O(\epsilon^3).$$

After following g_2 for time ϵ, we have

$$x(2\epsilon) = x(\epsilon) + \epsilon g_2(x(\epsilon)) + \frac{1}{2}\epsilon^2 \frac{\partial g_2}{\partial x} g_2(x(\epsilon)) + \ldots$$

Leaving out terms of higher order than ϵ^2, this becomes

$$\begin{aligned} x(2\epsilon) \approxeq\ & x(0) + \epsilon g_1(x(0)) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial x} g_1(x(0)) \\ & + \epsilon g_2(x(0) + \epsilon g_1(x(0))) + \frac{1}{2}\epsilon^2 \frac{\partial g_2}{\partial x} g_2(x(0)). \end{aligned}$$

This expands to

$$\begin{aligned} x(2\epsilon) \approxeq\ & x(0) + \epsilon g_1(x(0)) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial x} g_1(x(0)) \\ & + \epsilon g_2(x(0)) + \epsilon^2 \frac{\partial g_2}{\partial x} g_1(x(0)) + \frac{1}{2}\epsilon^2 \frac{\partial g_2}{\partial x} g_2(x(0)). \end{aligned}$$

Now continue by finding $x(3\epsilon)$ and $x(4\epsilon)$ to arrive at the result.