Chapter 15

System Theory and Analytical Techniques

This chapter is complementary to Chapter 14 in that it provides tools and concepts that can be used to develop better local planning methods (LPMs). Most of the material was developed in the field of control theory, which focuses mainly on characterizing the behavior of particular classes of systems, and controlling them in the absence of obstacles. The two-point boundary value problem (BVP), which was a frequent nuisance in Chapter 14, can be better understood and solved for many systems by using the ideas of this chapter. Keep in mind that throughout this chapter there are no obstacles. Although planning for this case was trivial in Part II, the presence of differential constraints brings many challenges.

The style in this chapter is to provide a brief survey of concepts and techniques, with the hope of inspiring further study in other textbooks and research literature. Modern control theory is a vast and fascinating subject, of which only the surface can be scratched in one chapter. Section 15.1 introduces stability and controllability concepts, both of which characterize possible arrivals in a goal state. Stability characterizes how the integral curves of a vector field behave around a goal point, and controllability indicates whether an action trajectory exists that arrives at a specified goal.

Section 15.2 revisits dynamic programming one last time. Here it becomes a partial differential equation expressed in terms of the optimal cost-to-go function. In some cases, it actually has a closed-form *solution*, as opposed to its main use in computer science, which is to obtain algorithm constraints. The powerful *Pontryagin's minimum principle*, which can be derived from dynamic programming, is also covered.

The remainder of the chapter is devoted to nonholonomic systems, which often arise from underactuated mechanical systems. Section 15.3 expresses the shortest paths between any pair of points for the Dubins car, the Reeds-Shepp car, and a differential drive, all of which were introduced in Section 13.1.2. The paths are a beautiful solution to the BVP and are particularly valuable as an LPM; for example, some have been used in the plan-and-transform method of Section 14.6.2.

Section 15.4 addresses some basic properties of nonholonomic systems. The most important issues are determining whether nonholonomic constraints are actually integrable (which removes all $\dot{x}_i$ variables) and characterizing reachable sets that arise due to nonholonomic constraints. Section 15.5 attempts to do the same as Section 15.3, but for more challenging nonholonomic systems. In these cases, the BVP problem may not be solved optimally, and some methods may not even reach the goal point precisely. Nevertheless, when applicable, they can be used to build powerful LPMs in a sampling-based motion planning algorithm.

15.1 Basic System Properties

This section provides a brief overview of two fundamental concepts in control theory: stability and controllability. Either can be considered as characterizing how a goal state is reached. Stability usually involves feedback and may only converge to the goal as time approaches infinity. Controllability assesses whether an action trajectory exists that leads exactly to a specified goal state. In both cases, there is no obstacle region in X.

15.1.1 Stability

The subject of stability addresses properties of a vector field with respect to a given point. Let X denote a smooth manifold on which the vector field is defined; X may be a C-space or a phase space. The given point is denoted as x_G and can be interpreted in motion planning applications as the goal state. Stability characterizes how x_G is approached from other states in X by integrating the vector field.

The given vector field f is considered as a velocity field, which is represented as

$$\dot{x} = f(x). \tag{15.1}$$

This looks like a state transition equation that is missing actions. If a system of the form $\dot{x} = f(x, u)$ is given, then u can be fixed by designing a feedback plan $\pi : X \rightarrow U$. This yields $\dot{x} = f(x, \pi(x))$, which is a vector field on X without any further dependency on actions. The dynamic programming approach in Section 14.5 computed such a solution. The process of designing a stable feedback plan is referred to in control literature as *feedback stabilization*.

Equilibrium points and Lyapunov stability At the very least, it seems that the state should remain fixed at x_G, if it is reached. A point $x_G \in X$ is called an *equilibrium point* (or *fixed point*) of the vector field f if and only if $f(x_G) = 0$. This does not, however, characterize how trajectories behave in the vicinity of x_G. Let $x_I \in X$ denote some initial state, and let $x(t)$ refer to the state obtained at time t after integrating the vector field f from $x_I = x(0)$.

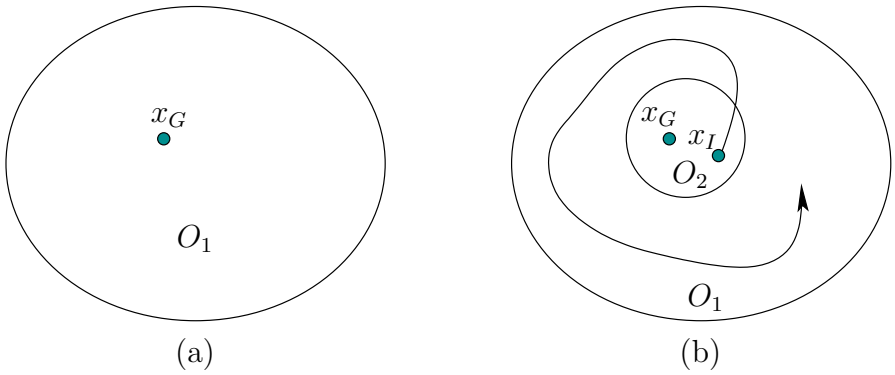

Figure 15.1: Lyapunov stability: (a) Choose any open set O_1 that contains x_G, and (b) there exists some open set O_2 from which trajectories will not be able to escape O_1. Note that convergence to x_G is not required.

See Figure 15.1. An equilibrium point $x_G \in X$ is called *Lyapunov stable* if for any open neighborhood[1] O_1 of x_G there exists another open neighborhood O_2 of x_G such that $x_I \in O_2$ implies that $x(t) \in O_1$ for all $t > 0$. If $X = \mathbb{R}^n$, then some intuition can be obtained by using an equivalent definition that is expressed in terms of the Euclidean metric. An equilibrium point $x_G \in \mathbb{R}^n$ is called *Lyapunov stable* if, for any $t > 0$, there exists some $\delta > 0$ such that $\|x_I - x_G\| < \delta$ implies that $\|x(t) - x_G\| < \epsilon$. This means that we can choose a ball around x_G with a radius as small as desired, and all future states will be trapped within this ball, as long as they start within a potentially smaller ball of radius δ. If a single δ can be chosen independently of every ϵ and x, then the equilibrium point is called *uniform* Lyapunov stable.

Asymptotic stability Lyapunov stability is weak in that it does not even imply that $x(t)$ converges to x_G as t approaches infinity. The states are only required to hover around x_G. Convergence requires a stronger notion called *asymptotic stability*. A point x_G is an *asymptotically stable* equilibrium point of f if:

1. It is a Lyapunov stable equilibrium point of f.
2. There exists some open neighborhood O of x_G such that, for any $x_I \in O$, $x(t)$ converges[2] to x_G as t approaches infinity.

For $X = \mathbb{R}^n$, the second condition can be expressed as follows: There exists some $\delta > 0$ such that, for any $x_I \in X$ with $\|x_I - x_G\| < \delta$, the state $x(t)$ converges to x_G as t approaches infinity. It may seem strange that two requirements are needed for asymptotic stability. The first one bounds the amount of wiggling room for the integral curve, which is not captured by the second condition.

[1] An *open neighborhood* of a point x means an open set that contains x.

[2] This convergence can be evaluated using the metric ρ on X.

Asymptotic stability appears to be a reasonable requirement, but it does not imply anything about how long it takes to converge. If x_G is asymptotically stable and there exist some $m > 0$ and $\alpha > 0$ such that

$$\|x(t) - x_G\| \leq me^{-\alpha t}\|x_I - x_G\|, \tag{15.2}$$

then x_G is also called *exponentially stable*. This provides a convenient way to express the rate of convergence.

For use in motion planning applications, even exponential convergence may not seem strong enough. This issue was discussed in Section 8.4.1. For example, in practice, one usually prefers to reach x_G in finite time, as opposed to only being "reached" in the limit. There are two common fixes. One is to allow asymptotic stability and declare the goal to be reached if the state arrives in some small, predetermined ball around x_G. In this case, the enlarged goal will always be reached in finite time if x_G is asymptotically stable. The other fix is to require a stronger form of stability in which x_G must be exactly reached in finite time. To enable this, however, discontinuous vector fields such as the inward flow of Figure 8.5b must be used. Most control theorists are appalled by this because infinite energy is usually required to execute such trajectories. On the other hand, discontinuous vector fields may be a suitable representation in some applications, as mentioned in Chapter 8. Note that without feedback this issue does not seem as important. The state trajectories designed in much of Chapter 14 were expected to reach the goal in finite time. Without feedback there was no surrounding vector field that was expected to maintain continuity or smoothness properties. Section 15.1.3 introduces controllability, which is based on actually arriving at the goal in finite time, but it is also based on the existence of one trajectory for a given system $\dot{x} = f(x, u)$, as opposed to a family of trajectories for a given vector field $x = f(x)$.

Time-varying vector fields The stability notions expressed here are usually introduced in the time-varying setting $\dot{x} = f(x, t)$. Since the vast majority of planning problems in this book are time-invariant, the presentation was confined to time-invariant vector fields. There is, however, one fascinating peculiarity in the topic of finding a feedback plan that stabilizes a system. *Brockett's condition* implies that for some time-invariant systems for which continuous, time-varying feedback plans exist, there does not exist a continuous time-invariant feedback plan [143, 156, 996]. This includes the class of driftless control systems, such as the simple car and the unicycle. This implies that to maintain continuity of the vector field, a time dependency must be introduced to allow the vector field to vary as x_G is approached! If continuity of the vector field is not important, then this concern vanishes.

Domains of attraction The stability definitions given so far are often called *local* because they are expressed in terms of a neighborhood of x_G. *Global* versions can also be defined by extending the neighborhood to all of X. An equilibrium

point is *globally asymptotically stable* if it is Lyapunov stable, and the integral curve from any $x_0 \in X$ converges to x_G as time approaches infinity. It may be the case that only points in some proper subset of X converge to x_G. The set of all points in X that converge to x_G is often called the *domain of attraction* of x_G. The funnels of Section 8.5.1 are based on domains of attraction. Also related is the backward reachable set from Section 14.2.1. In that setting, action trajectories were considered that lead to x_G in finite time. For the domain of attraction only asymptotic convergence to x_G is assumed, and the vector field is given (there are no actions to choose).

Limit cycles For some vector fields, states may be attracted into a *limit cycle*. Rather than stabilizing to a point, the state trajectories converge to a loop path in X. For example, they may converge to following a circle. This occurs in a wide variety of mechanical systems in which oscillations are possible. Some of the basic issues, along with several interesting examples for $X = \mathbb{R}^2$, are covered in [44].

15.1.2 Lyapunov Functions

Suppose a velocity field $\dot{x} = f(x)$ is given along with an equilibrium point, x_G. Can the various forms of stability be easily determined? One of the most powerful methods to prove stability is to construct a Lyapunov function. This will be introduced shortly, but first some alternatives are briefly mentioned.

If $f(x)$ is linear, which means that $f(x) = Ax$ for some constant $n \times n$ matrix A and $X = \mathbb{R}^n$, then stability questions with respect to the origin, $x_G = 0$, are answered by finding the eigenvalues of A [192]. The state $x = 0$ is asymptotically stable if and only if all eigenvalues of A have negative real parts. Consider the scalar case, $\dot{x} = ax$, for which $X = \mathbb{R}$ and a is a constant. The solution to this differential equation is $x(t) = x(0)\, e^{at}$, which converges to 0 only if $a < 0$. This can be easily extended to the case in which $X = \mathbb{R}^n$ and A is an $n \times n$ diagonal matrix for which each diagonal entry (or eigenvalue) is negative. For a general matrix, real or complex eigenvalues determine the stability (complex eigenvalues cause oscillations). Conditions also exist for Lyapunov stability. Every equilibrium state of $\dot{x} = Ax$ is Lyapunov stable if the eigenvalues of A all have nonpositive real parts, and the eigenvalues with zero real parts are distinct roots of the characteristic polynomial of A.

If $f(x)$ is nonlinear, then stability can sometimes be inferred by linearizing $f(x)$ about x_G and performing linear stability analysis. In many cases, however, this procedure is inconclusive (see Chapter 6 of [156]). Proving the stability of a vector field is a challenging task for most nonlinear systems. One approach is based on LaSalle's invariance principle [39, 156, 585] and is particularly useful for showing convergence to any of multiple goal states (see Section 5.4 of [846]). The other major approach is to construct a *Lyapunov function*, which is used as an intermediate tool to indirectly establish stability. If this method fails, then it still may be possible to show stability using other means. Therefore, it is a sufficient

condition for stability, but not a necessary one.

Determining stability Suppose a velocity field $\dot{x} = f(x)$ is given along with an equilibrium point x_G. Let ϕ denote a *candidate Lyapunov function*, which will be used as an auxiliary device for establishing the stability of f. An appropriate ϕ must be determined for the particular vector field f. This may be quite challenging in itself, and the details are not covered here. In a sense, the procedure can be characterized as "guess and verify," which is the way that many solution techniques for differential equations are described. If ϕ succeeds in establishing stability, then it is promoted to being called a *Lyapunov function* for f.

It will be important to characterize how ϕ varies in the direction of flow induced by f. This is measured by the *Lie derivative*,

$$\dot{\phi}(x) = \sum_{i=1}^{n} \frac{\partial \phi}{\partial x_i} f_i(x). \tag{15.3}$$

This results in a new function $\dot{\phi}(x)$, which indicates for each x the change in ϕ along the direction of $\dot{x} = f(x)$.

Several concepts are needed to determine stability. Let a function $h : [0, \infty) \rightarrow [0, \infty)$ be called a *hill* if it is continuous, strictly increasing, and $h(0) = 0$. This can be considered as a one-dimensional navigation function, which has a single local minimum at the goal, 0. A function $\phi : X \rightarrow [0, \infty)$ is called *locally positive definite* if there exists some open set $O \subseteq X$ and a hill function h such that $\phi(x_G) = 0$ and $\phi(x) \geq h(\|x\|)$ for all $x \in O$. If O can be chosen as $O = X$, and if X is bounded, then ϕ is called *globally positive definite* or just *positive definite*. In some spaces this may not be possible due to the topology of X; such issues arose when constructing navigation functions in Section 8.4.4. If X is unbounded, then h must additionally approach infinity as $\|x\|$ approaches infinity to yield a positive definite ϕ [846]. For $X = \mathbb{R}^n$, a quadratic form $x^T M x$, for which M is a positive definite matrix, is a globally positive definite function. This motivates the use of quadratic forms in Lyapunov stability analysis.

The Lyapunov theorems can now be stated [156, 846]. Suppose that ϕ is locally positive definite at x_G. If there exists an open set O for which $x_G \in O$, and $\dot{\phi}(x) \leq 0$ on all $x \in O$, then f is Lyapunov stable. If $-\dot{\phi}(x)$ is also locally positive definite on O, then f is asymptotically stable. If ϕ and $-\dot{\phi}$ are both globally positive definite, then f is globally asymptotically stable.

Example 15.1 (Establishing Stability via Lyapunov Functions) Let $X = \mathbb{R}$. Let $\dot{x} = f(x) = -x^5$, and we will attempt to show that $x = 0$ is stable. Let the candidate Lyapunov function be $\phi(x) = \frac{1}{2}x^2$. The Lie derivative (15.3) produces $\dot{\phi}(x) = -x^6$. It is clear that ϕ and $-\dot{\phi}$ are both globally positive definite; hence, 0 is a global, asymptotically stable equilibrium point of f. ■

Lyapunov functions in planning Lyapunov functions are closely related to navigation functions and optimal cost-to-go functions in planning. In the optimal discrete planning problem of Sections 2.3 and 8.2, the cost-to-go values can be considered as a discrete Lyapunov function. By applying the computed actions, a kind of discrete vector field can be imagined over the search graph. Each applied optimal action yields a reduction in the optimal cost-to-go value, until 0 is reached at the goal. Both the optimal cost-to-go and Lyapunov functions ensure that the trajectories do not become trapped in a local minimum. Lyapunov functions are more general than cost-to-go functions because they do not require optimality. They are more like navigation functions, as considered in Chapter 8. The requirements for a discrete navigation function, as given in Section 8.2.2, are very similar to the positive definite condition given in this section. Imagine that the navigation function shown in Figure 8.3 is a discrete approximation to a Lyapunov function over $\mathbb{R}^2$. In general, a Lyapunov function indicates some form of distance to x_G, although it may not be optimal. Nevertheless, it is based on making monotonic progress toward x_G. Therefore, it may serve as a distance function in many sampling-based planning algorithms of Chapter 14. Since it respects the differential constraints imposed by the system, it may provide a better indication of how to make progress during planning in comparison to a Euclidean metric that ignores these considerations. Lyapunov functions should be particularly valuable in the RDT method of Section 14.4.3, which relies heavily on the distance function over X.

15.1.3 Controllability

Now suppose that a system $\dot{x} = f(x, u)$ is given on a smooth manifold X as defined throughout Chapter 13 and used extensively in Chapter 14. The system can be considered as a parameterized family of vector fields in which u is the parameter. For stability, it was assumed that this parameter was fixed by a feedback plan to obtain some $\dot{x} = f(x)$. This section addresses *controllability*, which indicates whether one state is reachable from another via the existence of an action trajectory $\tilde{u}$. It may be helpful to review the reachable set definitions from Section 14.2.1.

Classical controllability Let $\mathcal{U}$ denote the set of permissible action trajectories for the system, as considered in Section 14.1.1. By default, this is taken as any $\tilde{u}$ for which (14.1) can be integrated. A system $\dot{x} = f(x, u)$ is called *controllable* if for all $x_I, x_G \in X$, there exists a time $t > 0$ and action trajectory $\tilde{u} \in \mathcal{U}$ such that upon integration from $x(0) = x_I$, the result is $x(t) = x_G$. Controllability can alternatively be expressed in terms of the reachable sets of Section 14.2.1. The system is controllable if $x_G \in R(x_I, \mathcal{U})$ for all $x_I, x_G \in X$.

A system is therefore controllable if a solution exists to any motion planning problem in the absence of obstacles. In other words, a solution always exists to the two-point boundary value problem (BVP).

Example 15.2 (Classical Controllability) All of the vehicle models in Section 13.1.2 are controllable. For example, in an infinitely large plane, the Dubins car can be driven between any two configurations. Note, however, that if the plane is restricted by obstacles, then this is not necessarily possible with the Dubins car. As an example of a system that is not controllable, let $X = \mathbb{R}$, $\dot{x} = u$, and $U = [0, 1]$. In this case, the state cannot decrease. For example, there exists no action trajectory that brings the state from $x_I = 1$ to $x_G = 0$. ■

Many methods for determining controllability of a system are covered in standard textbooks on control theory. If the system is linear, as given by (13.37) with dimensions m and n, then it is controllable if and only if the $n \times nm$ *controllability matrix*

$$M = [B \vdots AB \vdots A^2B \vdots \cdots \vdots A^{n-1}B] \tag{15.4}$$

has full rank [192]. This is called the *Kalman rank condition* [501]. If the system is nonlinear, then the controllability matrix can be evaluated on a linearized version of the system. Having full rank is sufficient to establish controllability from a single point (see Proposition 11.2 in [846]). If the rank is not full, however, the system may still be controllable. A fascinating property of some nonlinear systems is that they may be able to produce motions in directions that do not seem to be allowed at first. For example, the simple car given in Section 13.1.2 cannot slide sideways; however, it is possible to wiggle the car sideways by performing parallel-parking maneuvers. A method for determining the controllability of such systems is covered in Section 15.4.

For fully actuated systems of the form $\ddot{q} = h(q, \dot{q}, u)$, controllability can be determined by converting the system into double-integrator form, as considered in Section 14.4.1. Let the system be expressed as $\ddot{q} = u'$, in which $u' \in U'(q, \dot{q})$. If $U'(q, \dot{q})$ contains an open neighborhood of the origin of $\mathbb{R}^n$, and the same neighborhood can be used for any $x \in X$, then the system is controllable. If a nonlinear system is underactuated, as in the simple car, then controllability issues become considerably more complicated. The next concept is suitable for such systems.

STLC: Controllability that handles obstacles The controllability concept discussed so far has no concern for how far the trajectory travels in X before x_G is reached. This issue becomes particularly important for underactuated systems and planning among obstacles. These concerns motivate a natural question: Is there a form of controllability that is naturally suited for obstacles? It should declare that if a state is reachable from another in the absence of differential constraints, then it is also reachable with the given system $\dot{x} = f(x, u)$. This can be expressed using time-limited reachable sets. Let $R(x, \mathcal{U}, t)$ denote the set of all states reachable in time less than or equal to t, starting from x. A system $\dot{x} = f(x, u)$ is called *small-time locally controllable* (STLC) from x_I if there exists some $t > 0$ such that $x_I \in \text{int}(R(x_I, \mathcal{U}, t'))$ for all $t' \in (0, t]$ (here, int denotes the interior of a set, as

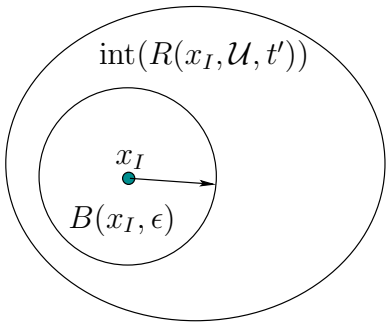

Figure 15.2: If the system is STLC, then motions can be made in any direction, in an arbitrarily small amount of time.

defined in Section 4.1.1). If the system $\dot{x} = f(x, u)$ is STLC from every $x_I \in X$, then the whole system is said to be STLC.

Consider using this definition to answer the question above. Since $\text{int}(R(x_I, \mathcal{U}, t'))$ is an open set, there must exist some small $\epsilon > 0$ for which the open ball $B(x_I, \epsilon)$ is a strict subset of $\text{int}(R(x_I, \mathcal{U}, t'))$. See Figure 15.2. Any point on the boundary of $B(x_I, \epsilon)$ can be reached, which means that a step of size ϵ can be taken in any direction, even though differential constraints exist. With obstacles, however, we have to be careful that the trajectory from x_I to the surface of $B(x_I, \epsilon)$ does not wander too far away.

Suppose that there is an obstacle region X_{obs}, and a violation-free state trajectory $\tilde{x}$ is given that terminates in x_G at time t_F and does not necessarily satisfy a given system. If the system is STLC, then it is always possible to find another trajectory, based on $\tilde{x}$, that satisfies the differential constraints. Apply the plan-and-transform method of Section 14.6.2. Suppose that intervals for potential replacement are chosen using binary recursive subdivision. Also suppose that an LPM exists that computes that shortest trajectory between any pair of states; this trajectory ignores obstacles but respects the differential constraints. Initially, $[0, t_F]$ is replaced by a trajectory from the LPM, and if it is not violation-free, then $[0, t_F]$ is subdivided into $[0, t_F/2]$ and $[t_F/2, t_F]$, and replacement is attempted on the smaller intervals. This idea can be applied recursively until eventually the segments are small enough that they must be violation-free.

This final claim is implied by the STLC property. No matter how small the intervals become, there must exist a replacement trajectory. If an interval is large, then there may be sufficient time to wander far from the original trajectory. However, as the time interval decreases, there is not enough time to deviate far from the original trajectory. (This discussion assumes mild conditions on f, such as being Lipschitz.) Suppose that the trajectory is protected by a collision-free tube of radius ϵ. Thus, all points along the trajectory are at least ϵ from the boundary of X_{free}. The time intervals can be chosen small enough to ensure that the trajectory deviations are less than ϵ from the original trajectory. Therefore, STLC is a very important property for a system to possess for planning in the presence of

obstacles. Section 15.4 covers some mathematical tools for determining whether a nonlinear system is STLC.

A concept closely related to controllability is *accessibility*, which is only concerned with the dimension of the reachable set. Let n be the dimension of X. If there exists some $t > 0$ for which the dimension of $R(x_I, \mathcal{U}, t)$ is n, then the system is called *accessible* from x_I. Alternatively, this may be expressed as requiring that $\text{int}(R(x_I, \mathcal{U}, t)) \neq \emptyset$.

Example 15.3 (Accessibility) Recall the system from Section 13.1.3 in which the state is trapped on a circle. In this case $X = \mathbb{R}^2$, and the state transition equation was specified by $\dot{x} = yu$ and $\dot{y} = -xu$. This system is not accessible because the reachable sets have dimension one. ■

A small-time version of accessibility can also be defined by requiring that there exists some t such that $\text{int}(R(x_I, \mathcal{U}, t')) \neq \emptyset$ for all $t' \in (0, t]$. Accessibility is particularly important for systems with drift.

15.2 Continuous-Time Dynamic Programming

Dynamic programming has been a recurring theme throughout most of this book. So far, it has always taken the form of computing optimal cost-to-go (or cost-to-come) functions over some sequence of stages. Both value iteration and Dijkstra-like algorithms have emerged. In computer science, dynamic programming is a fundamental insight in the development of algorithms that compute optimal solutions to problems. In its original form, however, dynamic programming was developed to solve the optimal control problem [84]. In this setting, a discrete set of stages is replaced by a continuum of stages, known as *time*. The dynamic programming recurrence is instead a partial differential equation, called the Hamilton-Jacobi-Bellman (HJB) equation. The HJB equation can be solved using numerical algorithms; however, in some cases, it can be *solved analytically*.[3] Section 15.2.2 briefly describes an analytical solution in the case of linear systems. Section 15.2.3 covers Pontryagin's minimum principle, which can be derived from the dynamic programming principle, and generalizes the optimization performed in Hamiltonian mechanics (recall Section 13.4.4).

15.2.1 Hamilton-Jacobi-Bellman Equation

The HJB equation is a central result in optimal control theory. Many other principles and design techniques follow from the HJB equation, which itself is just a statement of the dynamic programming principle in continuous time. A proper derivation of all forms of the HJB equation would be beyond the scope of this

[3]It is often surprising to computer scientists that dynamic programming in this case does not yield an algorithm. It instead yields a closed-form solution to the problem.

book. Instead, a time-invariant formulation that is most relevant to planning will be given here. Also, an informal derivation will follow, based in part on [95].

15.2.1.1 The discrete case

Before entering the continuous realm, the concepts will first be described for discrete planning, which is often easier to understand. Recall from Section 2.3 that if X, U, and the stages are discrete, then optimal planning can be performed by using value iteration or Dijkstra's algorithm on the search graph. The stationary, optimal cost-to-go function G^* can be used as a navigation function that encodes the optimal feedback plan. This was suggested in Section 8.2.2, and an example was shown in Figure 8.3.

Suppose that G^* has been computed under Formulation 8.1 (or Formulation 2.3). Let the state transition equation be denoted as

$$x' = f_d(x, u). \tag{15.5}$$

The dynamic programming recurrence for G^* is

$$G^*(x) = \min_{u \in U(x)} \{ l(x, u) + G^*(x') \}, \tag{15.6}$$

which may already be considered as a discrete form of the Hamilton-Jacobi-Bellman equation. To gain some insights into the coming concepts, however, some further manipulations will be performed.

Let u^* denote the optimal action that is applied in the min of (15.6). Imagine that u^* is hypothesized as the optimal action but needs to be tested in (15.6) to make sure. If it is truly optimal, then

$$G^*(x) = l(x, u^*) + G^*(f_d(x, u^*)). \tag{15.7}$$

This can already be considered as a discrete form of the Pontryagin minimum principle, which will appear in Section 15.2.3. By rearranging terms, a nice interpretation is obtained:

$$G^*(f_d(x, u^*)) - G^*(x) = -l(x, u^*). \tag{15.8}$$

In a single stage, the optimal cost-to-go drops by $l(x, u^*)$ when G^* is used as a navigation function (multiply (15.8) by -1). The optimal single-stage cost is revealed precisely when taking one step toward the goal along the optimal path. This incremental change in the cost-to-go function while moving in the best direction forms the basis of both the HJB equation and the minimum principle.

15.2.1.2 The continuous case

Now consider adapting to the continuous case. Suppose X and U are both continuous, but discrete stages remain, and verify that (15.5) to (15.8) still hold true.

Their present form can be used for any system that is approximated by discrete stages. Suppose that the discrete-time model of Section 14.2.2 is used to approximate a system $\dot{x} = f(x, u)$ on a state space X that is a smooth manifold. In that model, U was discretized to U_d, but here it will be left in its original form. Let Δt represent the time discretization.

The HJB equation will be obtained by approximating (15.6) with the discrete-time model and letting Δt approach zero. The arguments here are very informal; see [95, 570, 912] for more details. Using discrete-time approximation, the dynamic programming recurrence is

$$G^*(x) = \min_{u \in U(x)} \{ l_d(x, u) + G^*(x') \}, \tag{15.9}$$

in which l_d is a discrete-time approximation to the cost that accumulates over stage k and is given as

$$l_d(x, u) \approx l(x, u) \Delta t. \tag{15.10}$$

It is assumed that as Δt approaches zero, the total discretized cost converges to the integrated cost of the continuous-time formulation.

Using the linear part of a Taylor series expansion about x, the term $G^*(x')$ can be approximated as

$$G^*(x') \approx G^*(x) + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x, u) \Delta t. \tag{15.11}$$

This approximates $G^*(x')$ by its tangent plane at x. Substitution of (15.11) and (15.10) into (15.9) yields

$$G^*(x) \approx \min_{u \in U(x)} \left\{ l(x, u)\, \Delta t + G^*(x) + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x, u)\, \Delta t \right\}. \tag{15.12}$$

Subtracting $G^*(x)$ from both sides of (15.12) yields

$$\min_{u \in U(x)} \left\{ l(x, u)\, \Delta t + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x, u)\, \Delta t \right\} \approx 0. \tag{15.13}$$

Taking the limit as Δt approaches zero and then dividing by Δt yields the *HJB equation*:

$$\min_{u \in U(x)} \left\{ l(x, u) + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x, u) \right\} = 0. \tag{15.14}$$

Compare the HJB equation to (15.6) for the discrete-time case. Both indicate how the cost changes when moving in the best direction. Substitution of u^* for the optimal action into (15.14) yields

$$\sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x, u^*) = -l(x, u^*). \tag{15.15}$$

This is just the continuous-time version of (15.8). In the current setting, the left side indicates the derivative of the cost-to-go function along the direction obtained by applying the optimal action from x.

The HJB equation, together with a boundary condition that specifies the final-stage cost, sufficiently characterizes the optimal solution to the planning problem. Since it is expressed over the whole state space, solutions to the HJB equation yield optimal feedback plans. Unfortunately, the HJB equation cannot be solved analytically in most settings. Therefore, numerical techniques, such as the value iteration method of Section 14.5, must be employed. There is, however, an important class of problems that can be directly solved using the HJB equation; see Section 15.2.2.

15.2.1.3 Variants of the HJB equation

Several versions of the HJB equation exist. The one presented in (15.14) is suitable for planning problems such as those expressed in Chapter 14. If the cost-to-go functions are time-dependent, then the HJB equation is

$$\min_{u \in U(x)} \left\{ l(x,u,t) + \frac{\partial G^*}{\partial t} + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x,u,t) \right\} = 0, \tag{15.16}$$

and G^* is a function of both x and t. This can be derived again using a Taylor expansion, but with x and t treated as the variables. Most textbooks on optimal control theory present the HJB equation in this form or in a slightly different form by pulling $\partial G^*/\partial t$ outside of the min and moving it to the right of the equation:

$$\min_{u \in U(x)} \left\{ l(x,u,t) + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x,u,t) \right\} = -\frac{\partial G^*}{\partial t}. \tag{15.17}$$

In differential game theory, the HJB equation generalizes to the *Hamilton-Jacobi-Isaacs* (HJI) equations [59, 477]. Suppose that the system is given as (13.203) and a zero-sum game is defined using a cost term of the form $l(x,u,v,t)$. The HJI equations characterize saddle equilibria and are given as

$$\min_{u \in U(x)} \max_{v \in V(x)} \left\{ l(x,u,v,t) + \frac{\partial G^*}{\partial t} + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x,u,v,t) \right\} = 0 \tag{15.18}$$

and

$$\max_{v \in V(x)} \min_{u \in U(x)} \left\{ l(x,u,v,t) + \frac{\partial G^*}{\partial t} + \sum_{i=1}^{n} \frac{\partial G^*}{\partial x_i} f_i(x,u,v,t) \right\} = 0. \tag{15.19}$$

There are clear similarities between these equations and (15.16). Also, the swapping of the min and max operators resembles the definition of saddle points in Section 9.3.

15.2.2 Linear-Quadratic Problems

This section briefly describes a problem for which the HJB equation can be directly solved to yield a closed-form expression, as opposed to an algorithm that computes numerical approximations. Suppose that a linear system is given by (13.37), which requires specifying the matrices A and B. The task is to design a feedback plan that asymptotically stabilizes the system from any initial state. This is an infinite-horizon problem, and no termination action is applied.

An optimal solution is requested with respect to a cost functional based on matrix quadratic forms. Let Q be a nonnegative definite[4] $n \times n$ matrix, and let R be a positive definite $n \times n$ matrix. The *quadratic cost functional* is defined as

$$L(\tilde{x}, \tilde{u}) = \frac{1}{2} \int_0^\infty \big(x(t)^T Q x(t) + u(t)^T R u(t) \big) \, dt. \tag{15.20}$$

To guarantee that a solution exists that yields finite cost, several assumptions must be made on the matrices. The pair (A, B) must be *stabilizable*, and (A, Q) must be *detectable*; see [28] for specific conditions and a full derivation of the solution presented here.

Although it is not done here, the HJB equation can be used to derive the *algebraic Riccati equation*,

$$SA + A^T S - SBR^{-1}B^T S + Q = 0, \tag{15.21}$$

in which all matrices except S were already given. Methods exist that solve for S, which is a unique solution in the space of nonnegative definite $n \times n$ matrices.

The linear vector field

$$\dot{x} = \big(A - BR^{-1}B^T S \big) x \tag{15.22}$$

is asymptotically stable (the real parts of all eigenvalues of the matrix are negative). This vector field is obtained if u is selected using a feedback plan π defined as

$$\pi(x) = -R^{-1}B^T S x. \tag{15.23}$$

The feedback plan π is in fact optimal, and the optimal cost-to-go is simply

$$G^*(x) = \tfrac{1}{2} x^T S x. \tag{15.24}$$

Thus, for linear systems with quadratic cost, an elegant solution exists without resorting to numerical approximations. Unfortunately, the solution techniques do not generalize to nonlinear systems or linear systems among obstacles. Hence, the planning methods of Chapter 14 are justified.

However, many variations and extensions of the solutions given here do exist, but only for other problems that are expressed as linear systems with quadratic

[4]Nonnegative definite means $x^T Q x \geq 0$ for all $x \in \mathbb{R}$, and positive definite means $x^T R x > 0$ for all $x \in \mathbb{R}^n$.

cost. In every case, some variant of Riccati equations is obtained by application of the HJB equation. Solutions to time-varying systems are derived in [28]. If there is Gaussian uncertainty in predictability, then the linear-quadratic Gaussian (LQG) problem is obtained [564]. Linear-quadratic problems and solutions even exist for differential games of the form (13.204) [59].

15.2.3 Pontryagin's Minimum Principle

Pontryagin's minimum principle[5] is closely related to the HJB equation and provides conditions that an optimal trajectory must satisfy. Keep in mind, however, that the minimum principle provides *necessary* conditions, but not *sufficient conditions*, for optimality. In contrast, the HJB equation offered sufficient conditions. Using the minimum principle alone, one is often not able to conclude that a trajectory is optimal. In some cases, however, it is quite useful for finding candidate optimal trajectories. Any trajectory that fails to satisfy the minimum principle cannot be optimal.

To understand the minimum principle, we first return to the case of discrete planning. As mentioned previously, the minimum principle is essentially given by (15.7). This can be considered as a specialization of the HJB equation to the special case of applying the optimal action u^*. This causes the min to disappear, but along with it the global properties of the HJB equation also vanish. The minimum principle expresses conditions along the optimal trajectory, as opposed to the cost-to-go function over the whole state space. Therefore, it can at best assure local optimality in the space of possible trajectories.

The minimum principle for the continuous case is essentially given by (15.15), which is the continuous-time counterpart to (15.7). However, it is usually expressed in terms of adjoint variables and a Hamiltonian function, in the spirit of Hamiltonian mechanics from Section 13.4.4.

Let λ denote an n-dimensional vector of *adjoint variables*, which are defined as

$$\lambda_i = \frac{\partial G^*}{\partial x_i}. \tag{15.25}$$

The *Hamiltonian function* is defined as

$$H(x, u, \lambda) = l(x, u) + \sum_{i=1}^{n} \lambda_i f_i(x, u), \tag{15.26}$$

which is exactly the expression inside of the min of the HJB equation (15.14) after using the adjoint variable definition from (15.25). This can be compared to the Hamiltonian given by (13.192) in Section 13.4.4 (p from that context becomes λ

[5]This is often called Pontryagin's maximum principle, because Pontryagin originally defined it as a maximization [801]. The Hamiltonian used in most control literature is negated with respect to Pontryagin's Hamiltonian; therefore, it becomes minimized. Both names are in common use.

here). The two are not exactly the same, but they both are motivated by the same basic principles.

Under the execution of the optimal action trajectory $\tilde{u}^*$, the HJB equation implies that

$$H(x(t), u^*(t), \lambda(t)) = 0 \tag{15.27}$$

for all $t \geq 0$. This is just an alternative way to express (15.15). The fact that H remains constant appears very much like a conservation law, which was the basis of Hamiltonian mechanics in Section 13.4.4. The use of the Hamiltonian in the minimum principle is more general.

Using the HJB equation (15.14), the optimal action is given by

$$u^*(t) = \operatorname*{argmin}_{u \in U(x)} \{H(x(t), u(t), \lambda(t))\}. \tag{15.28}$$

In other words, the Hamiltonian is minimized precisely at $u(t) = u^*(t)$.

The missing piece of information so far is how λ evolves over time. It turns out that a system of the form

$$\dot{\lambda} = g(x, \lambda, u^*) \tag{15.29}$$

can be derived by differentiating the Hamiltonian (or, equivalently, the HJB equation) with respect to x. This yields two coupled systems, $\dot{x} = f(x, u^*)$ and (15.29). These can in fact be interpreted as a single system in a $2n$-dimensional phase space, in which each phase vector is (x, λ). This is analogous to the phase interpretation in Section 13.4.4 for Hamiltonian mechanics, which results in (13.198).

Remember that λ is defined in (15.25) just to keep track of the change in G^*. It would be helpful to have an explicit form for (15.29). Suppose that u^* is selected by a feedback plan to yield $u^* = \pi^*(x)$. In this case, the Hamiltonian can be interpreted as a function of only x and λ. Under this assumption, differentiating the Hamiltonian (15.26) with respect to x_i yields

$$\frac{\partial l(x, \pi^*(x))}{\partial x_i} + \sum_{j=1}^{n} \frac{\partial \lambda_j}{\partial x_i} f_j(x, \pi^*(x)) + \sum_{j=1}^{n} \lambda_j \frac{\partial f_j(x, \pi^*(x))}{\partial x_i}. \tag{15.30}$$

This validity of this differentiation requires a technical lemma that asserts that the derivatives of $\pi(x)$ can be disregarded (see Lemma 3.3.1 of [95]). Also, it will be assumed that U is convex in the arguments that follow, even though there exist proofs of the minimum principle that do not require this.

The second term in (15.30) is actually $\dot{\lambda}_i$, although it is hard to see at first. The total differential of λ_i with respect to the state is

$$d\lambda_i = \sum_{j=1}^{n} \frac{\partial \lambda_i}{\partial x_j} dx_j. \tag{15.31}$$

Dividing both sides by dt yields

$$\frac{d\lambda_i}{dt} = \sum_{j=1}^{n} \frac{\partial \lambda_i}{\partial x_j} \frac{dx_j}{dt} = \sum_{j=1}^{n} \frac{\partial \lambda_i}{\partial x_j} \dot{x}_j. \tag{15.32}$$

Each $\dot{x}_j$ is given by the state transition equation: $\dot{x}_j = f_j(x, \pi^*(x))$. Therefore,

$$\dot{\lambda}_i = \frac{d\lambda_i}{dt} = \frac{d}{dt}\frac{\partial G^*}{\partial x_i} = \sum_{j=1}^{n} \frac{\partial \lambda_i}{\partial x_j} f_j(x, \pi^*(x)). \tag{15.33}$$

Substituting (15.33) into (15.30) and setting the equation to zero (because the Hamiltonian is zero along the optimal trajectory) yields

$$\frac{\partial l(x, \pi^*(x))}{\partial x_i} + \dot{\lambda}_i + \sum_{j=1}^{n} \lambda_j \frac{\partial f_j(x, \pi^*(x))}{\partial x_i} = 0. \tag{15.34}$$

Solving for $\dot{\lambda}_i$ yields

$$\dot{\lambda}_i = -\frac{\partial l(x, \pi^*(x))}{\partial x_i} - \sum_{j=1}^{n} \lambda_j \frac{\partial f_j(x, \pi^*(x))}{\partial x_i}. \tag{15.35}$$

Conveniently, this is the same as

$$\dot{\lambda}_i = -\frac{\partial H}{\partial x_i}, \tag{15.36}$$

which yields the *adjoint transition equation*, as desired.

The transition equations given by $\dot{x} = f(x, u)$ and (15.36) specify the evolution of the system given by the minimum principle. These are analogous to Hamilton's equations (13.198), which were given in Section 13.4.4. The generalized momentum in that context becomes the adjoint variables here.

When applying the minimum principle, it is usually required to use the fact that the optimal action at all times must satisfy (15.28). Often, this is equivalently expressed as

$$H(x(t), u^*(t), \lambda(t)) \leq H(x(t), u(t), \lambda(t)), \tag{15.37}$$

which indicates that the Hamiltonian increases or remains the same whenever deviation from the optimal action occurs (the Hamiltonian cannot decrease).

Example 15.4 (Optimal Planning for the Double Integrator) Recall the double integrator system from Example 13.3. Let $\ddot{q} = u$, $\mathcal{C} = \mathbb{R}$, and $U = [-1, 1] \cup \{u_T\}$. Imagine a particle that moves in $\mathbb{R}$. The action is a force in either direction and has at most unit magnitude. The state transition equation is $\dot{x}_1 = x_2$ and $\dot{x}_2 = u$, and $X = \mathbb{R}^2$. The task is to perform optimal motion planning between any two states $x_I, x_G \in X$. From a given initial state x_I, a goal state x_G must be reached in minimum time. The cost functional is defined in this case as $l(x, u) = 1$ for all $x \in X$ and and $u \in U$ such that $u \neq u_T$.

Using (15.26), the Hamiltonian is defined as

$$H(x, u, \lambda) = 1 + \lambda_1 x_2 + \lambda_2 u. \tag{15.38}$$

The optimal action trajectory is obtained from (15.28) as

$$u^*(t) = \operatorname*{argmin}_{u \in [-1,1]} \{1 + \lambda_1(t)x_2(t) + \lambda_2(t)u(t)\} . \tag{15.39}$$

If $\lambda_2(t) < 0$, then $u^*(t) = 1$, and if $\lambda_2(t) > 0$, then $u^*(t) = -1$. Thus, the action may be assigned as $u^*(t) = -\text{sgn}(\lambda_2(t))$, if $\lambda_2(t) \neq 0$. Note that these two cases are the "bangs" of the bang-bang control from Section 14.6.3, and they are also the extremal actions used for the planning algorithm in Section 14.4.1. At the boundary case in which $\lambda_2(t) = 0$, any action in $[-1, 1]$ may be chosen.

The only remaining task is to determine the values of the adjoint variables over time. The adjoint transition equation is obtained from (15.36) as $\dot{\lambda}_1 = 0$ and $\dot{\lambda}_2 = -\lambda_1$. The solutions are $\lambda_1(t) = c_1$ and $\lambda_2(t) = c_2 - c_1 t$, in which c_1 and c_2 are constants that can be determined at $t = 0$ from (15.38) and (15.39). The optimal action depends only on the sign of $\lambda_2(t)$. Since its solution is the equation of a line, it can change signs at most once. Therefore, there are four possible kinds of solutions, depending on the particular x_I and x_G:

1. Pure acceleration, $u^*(t) = 1$, is applied for all time.

2. Pure deceleration, $u^*(t) = -1$, is applied for all time.

3. Pure acceleration is applied up to some time t' and is followed immediately by pure deceleration until the final time.

4. Pure deceleration is applied up to some time t' followed immediately by pure acceleration until the final time.

For the last two cases, t' is often called the *switching time*, at which point a discontinuity in $\tilde{u}^*$ occurs. These two are bang-bang solutions, which were described in Section 14.6.3. ■

This was one of the simplest possible examples, and the optimal solution was easily found because the adjoint variables are linear functions of time. Section 15.3 covers optimal solutions for the Dubins car, the Reeds-Shepp car, and the differential drive, all of which can be established using the minimum principle combined with some geometric arguments. As systems become more complicated, such analysis is unfortunately too difficult. In these cases, sampling-based methods, such as those of Chapter 14, must be used to determine optimal trajectories.

One common complication is the existence of *singular arcs* along the solution trajectory. These correspond to a degeneracy in H with respect to u over some duration of time. This could be caused, for example, by having H independent of u. In Example 15.4, H became independent of u when $\lambda_2(t) = 0$; however, there was no singular arc because this could only occur for an instant of time. If the duration had been longer, then there would be an interval of time over which the optimal action could not be determined. In general, if the Hessian (recall definition

from (8.48)) of H with respect to u is a positive definite matrix, then there are no singular arcs (this is often called the Legendre-Clebsch condition). The minimum principle in this case provides a sufficient condition for local optimality in the space of possible state trajectories. If the Hessian is not positive definite for some interval $[t_1, t_2]$ with $t_1 < t_2$, then additional information is needed to determine the optimal trajectory over the singular arc from $x^*(t_1)$ to $x^*(t_2)$.

Note that all of this analysis ignores the existence of obstacles. There is nothing to prevent the solutions from attempting to enter an obstacle region. The action set $U(x)$ and cost $l(x, u)$ can be adjusted to account for obstacles; however, determining an optimal solution from the minimum principle becomes virtually impossible, except in some special cases.

There are other ways to derive the minimum principle. Recall from Section 13.4.4 that Hamilton's equations can be derived from the Euler-Lagrange equation. It should not be surprising that the minimum principle can also be derived using variational principles [95, 789]. The minimum principle can also be interpreted as a form of constrained optimization. This yields the interpretation of λ as Lagrange multipliers. A very illuminating reference for further study of the minimum principle is Pontryagin's original works [801].

Time optimality Interesting interpretations of the minimum principle exist for the case of optimizing the time to reach the goal [424, 903]. In this case, $l(x, u) = 1$ in (15.26), and the cost term can be ignored. For the remaining portion, let λ be defined as

$$\lambda_i = -\frac{\partial G^*}{\partial x_i}, \tag{15.40}$$

instead of using (15.25). In this case, the Hamiltonian can be expressed as

$$H(x, u, \lambda) = \sum_{i=1}^{n} \lambda_i f_i(x, u) = \left\langle -\frac{\partial G^*}{\partial x}, f(x, u) \right\rangle, \tag{15.41}$$

which is an inner product between $f(x, u)$ and the negative gradient of G^*. Using (15.40), the Hamiltonian should be maximized instead of minimized (this is equivalent to Pontryagin's original formulation [801]). An inner product of two vectors increases as their directions become closer to parallel. Optimizing (15.41) amounts to selecting u so that $\dot{x}$ is as close as possible to the direction of steepest descent of G^*. This is nicely interpreted by considering how the boundary of the reachable set $R(x_0, \mathcal{U}, t)$ propagates through X. By definition, the points on $\partial R(x_0, \mathcal{U}, t)$ must correspond to time-optimal trajectories. Furthermore, $\partial R(x_0, \mathcal{U}, t)$ can be interpreted as a propagating wavefront that is perpendicular to $-\partial G^*/\partial x$. The minimum principle simply indicates that u should be chosen so that $\dot{x}$ points into the propagating boundary, as close to being orthogonal as possible [424].

15.3 Optimal Paths for Some Wheeled Vehicles

For some of the wheeled vehicle models of Section 13.1.2, the shortest path between any pair of configurations was completely characterized. In this section, $X = \mathcal{C} = \mathbb{R}^2 \times \mathbb{S}^1$, which corresponds to the C-space for a rigid body in the plane. For each model, the path length in $\mathcal{C}$ must be carefully defined to retain some physical significance in the world $\mathcal{W} = \mathbb{R}^2$ in which the vehicle travels. For example, in the case of the simple car, the distance in $\mathcal{W}$ traveled by the center of the rear axle will be optimized. If the coordinate frame is assigned appropriately, this corresponds to optimizing the path length in the $\mathbb{R}^2$ subspace of $\mathcal{C}$ while ignoring orientation. Keep in mind that the solutions given in this section depend heavily on the particular cost functional that is optimized.

Sections 15.3.1–15.3.3 cover the shortest paths for the Dubins car, the Reeds-Shepp car, and a differential-drive model, respectively. In each case, the paths can be elegantly described as combinations of a few motion primitives. Due to symmetries, it is sufficient to describe the optimal paths from a fixed initial configuration $q_I = (0, 0, 0)$ to any goal configuration $q_G \in \mathcal{C}$. If the optimal path is desired from a different $q_I \in \mathcal{C}$, then it can be recovered from rigid-body transformations applied to q_I and q_G (the whole path can easily be translated and rotated without effecting its optimality, provided that q_G does not move relative to q_I). Alternatively, it may be convenient to fix q_G and consider optimal paths from all possible q_I.

Once q_I (or q_G) is fixed, $\mathcal{C}$ can be partitioned into cells that correspond to sets of placements for q_G (or q_I). Inside of each cell, the optimal curve is described by a fixed sequence of parameterized motion primitives. For example, one cell for the Dubins car indicates "turn left," "go straight," and then "turn right." The curves are ideally suited for use as an LPM in a sampling-based planning algorithm.

This section mainly focuses on presenting the solutions. Establishing their correctness is quite involved and is based in part on Pontryagin's minimum principle from Section 15.2.3. Other important components are Filipov's existence theorem (see [903]) and Boltyanskii's sufficient condition for optimality (which also justifies dynamic programming) [130]. Substantially more details and justifications of the curves presented in Sections 15.3.1 and 15.3.2 appear in [903, 904, 923]. The corresponding details for the curves of Section 15.3.3 appear in [64].

15.3.1 Dubins Curves

Recall the Dubins version of the simple car given in Section 13.1.2. The system was specified in (13.15). It is assumed here that the car moves at constant forward speed, $u_s = 1$. The other important constraint is the maximum steering angle ϕ_{max}, which results in a minimum turning radius ρ_{min}. As the car travels, consider the length of the curve in $\mathcal{W} = \mathbb{R}^2$ traced out by a pencil attached to the center of the rear axle. This is the location of the body-frame origin in Figure 13.1. The task is to minimize the length of this curve as the car travels between any q_I and

Symbol	Steering: u
S	0
L	1
R	-1

Figure 15.3: The three motion primitives from which all optimal curves for the Dubins car can be constructed.

q_G. Due to ρ_{min}, this can be considered as a bounded-curvature shortest-path problem. If $\rho_{min} = 0$, then there is no curvature bound, and the shortest path follows a straight line in $\mathbb{R}^2$. In terms of a cost functional of the form (8.39), the criterion to optimize is

$$L(\tilde{q}, \tilde{u}) = \int_0^{t_F} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} dt, \tag{15.42}$$

in which t_F is the time at which q_G is reached, and a configuration is denoted as $q = (x, y, \theta)$. If q_G is not reached, then it is assumed that $L(\tilde{q}, \tilde{u}) = \infty$.

Since the speed is constant, the system can be simplified to

$$\begin{aligned} \dot{x} &= \cos\theta \\ \dot{y} &= \sin\theta \\ \dot{\theta} &= u, \end{aligned} \tag{15.43}$$

in which u is chosen from the interval $U = [-\tan\phi_{max}, \tan\phi_{max}]$. This implies that (15.42) reduces to optimizing the time t_F to reach q_G because the integrand reduces to 1. For simplicity, assume that $\tan\phi = 1$. The following results also hold for any $\phi_{max} \in (0, \pi/2)$.

It was shown in [294] that between any two configurations, the shortest path for the Dubins car can always be expressed as a combination of no more than three motion primitives. Each motion primitive applies a constant action over an interval of time. Furthermore, the only actions that are needed to traverse the shortest paths are $u \in \{-1, 0, 1\}$. The primitives and their associated symbols are shown in Figure 15.3. The S primitive drives the car straight ahead. The L and R primitives turn as sharply as possible to the left and right, respectively. Using these symbols, each possible kind of shortest path can be designated as a sequence of three symbols that corresponds to the order in which the primitives are applied. Let such a sequence be called a *word* . There is no need to have two consecutive primitives of the same kind because they can be merged into one. Under this observation, ten possible words of length three are possible. Dubins showed that only these six words are possibly optimal:

$$\{LRL,\ RLR,\ LSL,\ LSR,\ RSL,\ RSR\}. \tag{15.44}$$

The shortest path between any two configurations can always be characterized by one of these words. These are called the *Dubins curves*.

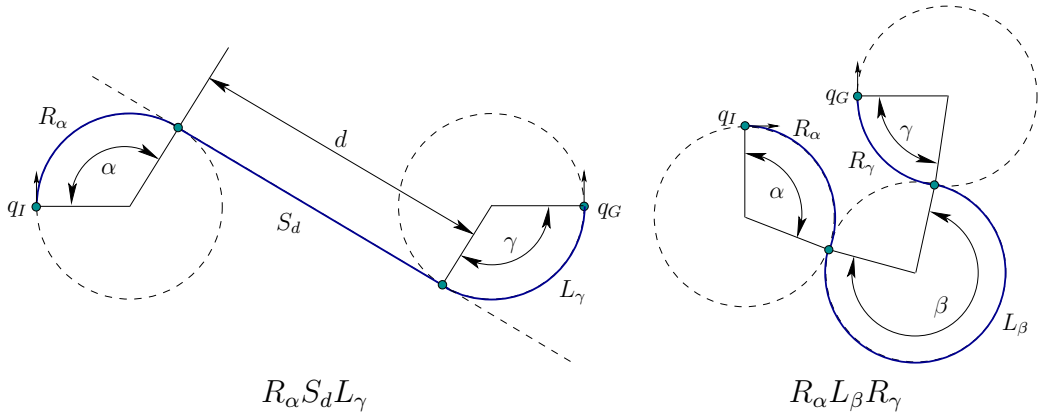

Figure 15.4: The trajectories for two words are shown in $\mathcal{W} = \mathbb{R}^2$.

To be more precise, the duration of each primitive should also be specified. For L or R, let a subscript denote the total amount of rotation that accumulates during the application of the primitive. For S, let a subscript denote the total distance traveled. Using such subscripts, the Dubins curves can be more precisely characterized as

$$\{L_\alpha\, R_\beta\, L_\gamma,\ R_\alpha\, L_\beta\, R_\gamma,\ L_\alpha\, S_d\, L_\gamma,\ L_\alpha\, S_d\, R_\gamma,\ R_\alpha\, S_d\, L_\gamma,\ R_\alpha\, S_d\, R_\gamma\}, \tag{15.45}$$

in which $\alpha, \gamma \in [0, 2\pi)$, $\beta \in (\pi, 2\pi)$, and $d \geq 0$. Figure 15.4 illustrates two cases. Note that β must be greater than π (if it is less, then some other word becomes optimal).

It will be convenient to invent a compressed form of the words to group together paths that are qualitatively similar. This will be particularly valuable when Reeds-Shepp curves are introduced in Section 15.3.2 because there are 46 of them, as opposed to 6 Dubins curves. Let C denote a symbol that means "curve," and represents either R or L. Using C, the six words in (15.44) can be compressed to only two *base words*:

$$\{CCC,\ CSC\}. \tag{15.46}$$

In this compressed form, remember that two consecutive Cs must be filled in by distinct turns (RR and LL are not allowed as subsequences). In compressed form, the base words can be specified more precisely as

$$\{C_\alpha\, C_\beta\, C_\gamma,\ C_\alpha\, S_d\, C_\gamma\}, \tag{15.47}$$

in which $\alpha, \gamma \in [0, 2\pi)$, $\beta \in (\pi, 2\pi)$, and $d \geq 0$.

Powerful information has been provided so far for characterizing the shortest paths; however, for a given q_I and q_G, two problems remain:

1. Which of the six words in (15.45) yields the shortest path between q_I and q_G?

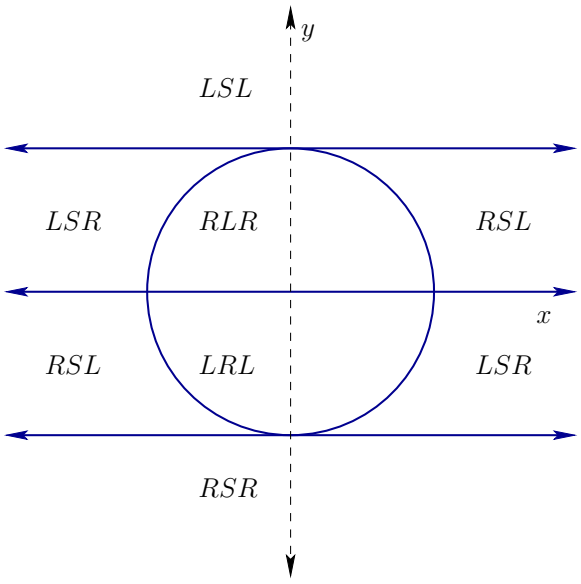

Figure 15.5: A slice at $\theta = \pi$ of the partition into word-invariant cells for the Dubins car. The circle is centered on the origin.

2. What are the values of the subscripts, α, β, γ, and d for the particular word?

To use the Dubins curves as an LPM, these questions should be answered efficiently. One simple approach is to try all six words and choose the shortest one. The parameters for each word can be determined by tracing out minimum-radius circles from q_I and q_G, as shown in Figure 14.23. Another way is to use the precise characterization of the regions over which a particular word is optimal. Suppose that q_G is fixed at $(0, 0, 0)$. Based on the possible placements of q_I, the C-space can be partitioned into cells for which the same word is optimal. The cells and their boundaries are given precisely in [903]. As an example, a slice of the cell decomposition for $\theta = \pi$ is shown in Figure 15.5.

In addition to use as an LPM, the resulting cost of the shortest path may be a useful distance function in many sampling-based planning algorithms. This is sometimes called the *Dubins metric* (it is not, however, a true metric because it violates the symmetry axiom). This can be considered as the optimal cost-to-go G^*. It could have been computed approximately using the dynamic programming approach in Section 14.5; however, thanks to careful analysis, the exact values are known. One interesting property of the Dubins metric is that it is discontinuous; see Figure 15.6. Compare the cost of traveling $\pi/2$ using the R primitive to the cost of traveling to a nearby point that would require a smaller turning radius than that achieved by the R primitive. The required action does not exist in U, and the point will have to be reached by a longer sequence of primitives. The discontinuity in G^* is enabled by the fact that the Dubins car fails to possess the STLC property from Section 15.1.3. For STLC systems, G^* is continuous.

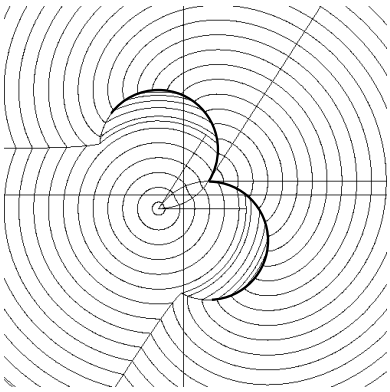

Figure 15.6: Level sets of the Dubins metric are shown in the plane. Along two circular arcs, the metric is discontinuous (courtesy of Philippe Souères).

15.3.2 Reeds-Shepp Curves

Now consider the shortest paths of the Reeds-Shepp car. The only difference in comparison to the Dubins car is that travel in the reverse direction is now allowed. The same criterion (15.42) is optimized, which is the distance traveled by the center of the rear axle. The shortest path is equivalent to the path that takes minimum time, as for the Dubins car. The simplified system in (15.43) can be enhanced to obtain

$$\begin{aligned} \dot{x} &= u_1 \cos\theta \\ \dot{y} &= u_1 \sin\theta \\ \dot{\theta} &= u_1 u_2, \end{aligned} \tag{15.48}$$

in which $u_1 \in \{-1, 1\}$ and $u_2 \in [-\tan\phi_{max}, \tan\phi_{max}]$. The first action variable, u_1, selects the gear, which is forward ($u_1 = 1$) or reverse ($u_1 = -1$). Once again, assume for simplicity that $u_2 \in [-1, 1]$. The results stated here apply to any $\phi_{max} \in (0, \pi/2)$.

It was shown in [814] that there are no more than 48 different words that describe the shortest paths for the Reeds-Shepp car. The base word notation from Section 15.3.1 can be extended to nicely express the shortest paths. A new symbol, "$|$", is used in the words to indicate that the "gear" is shifted from forward to reverse or reverse to forward. Reeds and Shepp showed that the shortest path for their car can always be expressed with one of the following base words:

$$\begin{aligned} \{&C|C|C,\ \ CC|C,\ \ C|CC,\ \ CSC,\ \ CC_\beta|C_\beta C,\ \ C|C_\beta C_\beta|C, \\ &C|C_{\pi/2}SC,\ \ CSC_{\pi/2}|C,\ \ C|C_{\pi/2}SC_{\pi/2}|C\}. \end{aligned} \tag{15.49}$$

As many as five primitives could be needed to execute the shortest path. A subscript of $\pi/2$ is given in some cases because the curve must be followed for precisely $\pi/2$ radians. For some others, β is given as a subscript to indicate that it must match the parameter of another primitive. The form given in (15.49)

Base	α	β	γ	d
$C_\alpha \vert C_\beta \vert C_\gamma$	$[0, \pi]$	$[0, \pi]$	$[0, \pi]$	$-$
$C_\alpha \vert C_\beta C_\gamma$	$[0, \beta]$	$[0, \pi/2]$	$[0, \beta]$	$-$
$C_\alpha C_\beta \vert C_\gamma$	$[0, \beta]$	$[0, \pi/2]$	$[0, \beta]$	$-$
$C_\alpha S_d C_\gamma$	$[0, \pi/2]$	-	$[0, \pi/2]$	$(0, \infty)$
$C_\alpha C_\beta \vert C_\beta C_\gamma$	$[0, \beta]$	$[0, \pi/2]$	$[0, \beta]$	$-$
$C_\alpha \vert C_\beta C_\beta \vert C_\gamma$	$[0, \beta]$	$[0, \pi/2]$	$[0, \beta]$	$-$
$C_\alpha \vert C_{\pi/2} S_d C_{\pi/2} \vert C_\gamma$	$[0, \pi/2]$	-	$[0, \pi/2]$	$(0, \infty)$
$C_\alpha \vert C_{\pi/2} S_d C_\gamma$	$[0, \pi/2]$	-	$[0, \pi/2]$	$(0, \infty)$
$C_\alpha S_d C_{\pi/2} \vert C_\gamma$	$[0, \pi/2]$	-	$[0, \pi/2]$	$(0, \infty)$

Figure 15.7: The interval ranges are shown for each motion primitive parameter for the Reeds-Shepp optimal curves.

Symbol	Gear: u_1	Steering: u_2
S^+	1	0
S^-	-1	0
L^+	1	1
L^-	-1	1
R^+	1	-1
R^-	-1	-1

Figure 15.8: The six motion primitives from which all optimal curves for the Reeds-Shepp car can be constructed.

is analogous to (15.46) for the Dubins car. The parameter ranges can also be specified, to yield a form analogous to (15.47). The result is shown in Figure 15.7. Example curves for two cases are shown in Figure 15.9.

Now the base words will be made more precise by specifying the particular motion primitive. Imagine constructing a list of words analogous to (15.44) for the Dubins car. There are six primitives as shown in Figure 15.8. The symbols S, L, and R are used again. To indicate the forward or reverse gear, $+$ and $-$ superscripts will be used as shown in Figure 15.8.[6]

Figure 15.10 shows 48 different words, which result from uncompressing the base words expressed using C, S, and "$\vert$" in (15.49). Each shortest path is a word with length at most five. There are substantially more words than for the Dubins car. Each base word in (15.49) expands into four or eight words using the motion primitives. To uncompress each base word, the rule that R and L cannot be applied consecutively is maintained. This yields four possibilities for the first

[6]This differs conceptually from the notation used in [903]. There, r^- corresponds to L^- here. The L here means that the steering wheel is positioned for a left turn, but the car is in reverse. This aids in implementing the rule that R and L cannot be consecutive in a word.

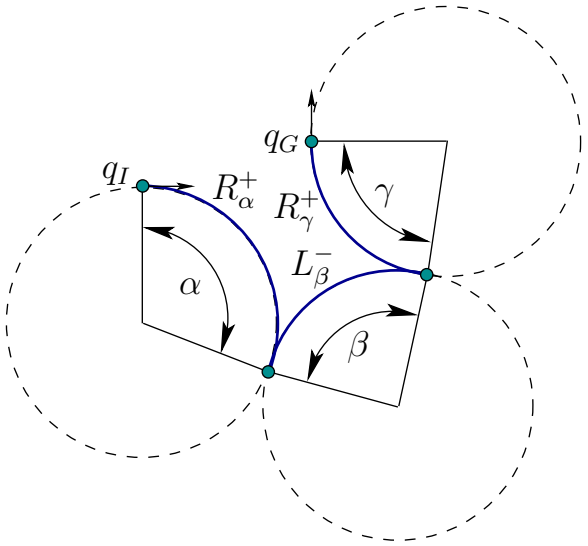

Figure 15.9: An example of the $R^+_\alpha L^-_\beta R^+_\gamma$ curve. This uses reverse to generate a curve that is shorter than the one in Figure 15.4b for the Dubins car.

six compressed words. The remaining three involve an intermediate S primitive, which allows eight possible sequences of Rs and Ls for each one. Two of the 48 words were eliminated in [923]. Each of the remaining 46 words can actually occur for a shortest path and are called the *Reeds-Shepp curves*.

For use as an LPM, the problem appears once again of determining the particular word and parameters for a given q_I and q_G. This was not difficult for Dubins curves, but now there are 46 possibilities. The naive approach of testing every word and choosing the shortest one may be too costly. The precise cell boundaries in $\mathcal{C}$ over which each word applies are given in [903]. The cell boundaries are unfortunately quite complicated, which makes the point location algorithm difficult to implement. A simple way to prune away many words from consideration is to use intervals of validity for θ. For some values of θ, certain compressed words are impossible as shortest paths. A convenient table of words that become active over ranges of θ is given in [903]. Once again, the length of the shortest path can serve as a distance function in sampling-based planning algorithms. The resulting *Reeds-Shepp metric* is continuous because the Reeds-Shepp car is STLC, which will be established in Section 15.4.

15.3.3 Balkcom-Mason Curves

In recent years, two more families of optimal curves have been determined [64, 211]. Recall the differential-drive system from Section 13.1.2, which appears in many mobile robot systems. In many ways, it appears that the differential drive is a special case of the simple car. The expression of the system given in (13.17) can be made to appear identical to the Reeds-Shepp car system in (15.48). For example, letting $r = 1$ and $L = 1$ makes them equivalent by assigning $u_\omega = u_1$ and

Base word	Sequences of motion primitives
$C\|C\|C$	$(L^+R^-L^+)(L^-R^+L^-)(R^+L^-R^+)(R^-L^+R^-)$
$CC\|C$	$(L^+R^+L^-)(L^-R^-L^+)(R^+L^+R^-)(R^-L^-R^+)$
$C\|CC$	$(L^+R^-L^-)(L^-R^+L^+)(R^+L^-R^-)(R^-L^+R^+)$
CSC	$(L^+S^+L^+)(L^-S^-L^-)(R^+S^+R^+)(R^-S^-R^-)$ $(L^+S^+R^+)(L^-S^-R^-)(R^+S^+L^+)(R^-S^-L^-)$
$CC_\beta\|C_\beta\,C$	$(L^+R_\beta^+L_\beta^-R^-)(L^-R_\beta^-L_\beta^+R^+)(R^+L_\beta^+R_\beta^-L^-)(R^-L_\beta^-R_\beta^+L^+)$
$C\|C_\beta\,C_\beta\|C$	$(L^+R_\beta^-L_\beta^-R^+)(L^-R_\beta^+L_\beta^+R^-)(R^+L_\beta^-R_\beta^-L^+)(R^-L_\beta^+R_\beta^+L^-)$
$C\|C_{\pi/2}SC$	$(L^+R_{\pi/2}^-S^-R^-)(L^-R_{\pi/2}^+S^+R^+)(R^+L_{\pi/2}^-S^-L^-)(R^-L_{\pi/2}^+S^+L^+)$ $(L^+R_{\pi/2}^-S^-L^-)(L^-R_{\pi/2}^+S^+L^+)$ $(R^+L_{\pi/2}^-S^-R^-)(R^-L_{\pi/2}^+S^+R^+)$
$CSC_{\pi/2}\|C$	$(L^+S^+L_{\pi/2}^+R^-)(L^-S^-L_{\pi/2}^-R^+)(R^+S^+R_{\pi/2}^+L^-)(R^-S^-R_{\pi/2}^-L^+)$ $(R^+S^+L_{\pi/2}^+R^-)(R^-S^-L_{\pi/2}^-R^+)(L^+S^+R_{\pi/2}^+L^-)(L^-S^-R_{\pi/2}^-L^+)$
$C\|C_{\pi/2}SC_{\pi/2}\|C$	$(L^+R_{\pi/2}^-S^-L_{\pi/2}^-R^+)(L^-R_{\pi/2}^+S^+L_{\pi/2}^+R^-)$ $(R^+L_{\pi/2}^-S^-R_{\pi/2}^-L^+)(R^-L_{\pi/2}^+S^+R_{\pi/2}^+L^-)$

Figure 15.10: The 48 curves of Reeds and Shepp. Sussmann and Tang [923] showed that $(L^-R^+L^-)$ and $(R^-L^+R^-)$, which appear in the first row, can be eliminated. Hence, only 46 words are needed to describe the shortest paths.

$u_\psi = u_1 u_2$. Consider the distance traveled by a point attached to the center of the differential-drive axle using (15.42). Minimizing this distance for any q_I and q_G is trivial, as shown in Figure 13.4 of Section 13.1.2. The center point can be made to travel in a straight line in $\mathcal{W} = \mathbb{R}^2$. This would be possible for the Reeds-Shepp car if $\rho_{min} = 0$, which implies that $\phi_{max} = \pi/2$. It therefore appeared for many years that no interesting curves exist for the differential drive.

The problem, however, with measuring the distance traveled by the axle center is that pure rotations are cost-free. This occurs when the wheels rotate at the same speed but with opposite angular velocities. The center does not move; however, the time duration, energy expenditure, and wheel rotations that occur are neglected. By incorporating one or more of these into the cost functional, a challenging optimization arises. Balkcom and Mason bounded the speed of the differential drive and minimized the total time that it takes to travel from q_I to q_G. Using (13.16), the action set is defined as $U = [-1, 1] \times [-1, 1]$, in which the maximum rotation rate of each wheel is one (an alternative bound can be used without loss of generality). The criterion to optimize is

$$L(\tilde{q}, \tilde{u}) = \int_0^{t_F} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} + |\dot{\theta}(t)| dt, \tag{15.50}$$

which takes θ into account, whereas it was neglected in (15.42). This criterion is once again equivalent to minimizing the time to reach q_G. The resulting model will be referred to as the *Balkcom-Mason drive*. An alternative criterion is the total amount of wheel rotation; this leads to an alternative family of optimal curves [211].

Symbol	Left wheel: u_l	Right wheel: u_r
$\Uparrow$	1	1
$\Downarrow$	-1	-1
$\curvearrowleft$	-1	1
$\curvearrowright$	1	-1

Figure 15.11: The four motion primitives from which all optimal curves for the differential-drive robot can be constructed.

It was shown in [64] that only the four motion primitives shown in Figure 15.11 are needed to express time-optimal paths for the differential-drive robot. Each primitive corresponds to holding one action variable fixed at its limit for an interval of time. Using the symbols in Figure 15.11 (which were used in [64]), words can be formed that describe the optimal path. It has been shown that the word length is no more than five. Thus, any shortest paths may be expressed as a piecewise-constant action trajectory in which there are no more than five pieces. Every piece corresponds to one of the primitives in Figure 15.11.

It is convenient in the case of the Balkcom-Mason drive to use the same symbols for both base words and for precise specification of primitives. Symmetry transformations will be applied to each base word to yield a family of eight words that precisely specify the sequences of motion primitives. Nine base words describe the shortest paths:

$$\{\curvearrowright,\ \Downarrow,\ \Downarrow\curvearrowright,\ \curvearrowright\Downarrow\curvearrowright,\ \Uparrow\curvearrowleft_{\pi}\Downarrow,\ \curvearrowleft\Downarrow\curvearrowleft,\ \Downarrow\curvearrowright\curvearrowright,\ \curvearrowleft\Downarrow\curvearrowright\Uparrow,\ \Uparrow\curvearrowleft\Downarrow\curvearrowright\Uparrow\}. \quad (15.51)$$

This is analogous to the compressed forms given in (15.46) and (15.49). The motions are depicted in Figure 15.12.

Figure 15.13 shows 40 distinct *Balkcom-Mason curves* that result from applying symmetry transformations to the base words of (15.51). There are 72 entries in Figure 15.13, but many are identical. The transformation T_1 indicates that the directions of $\Uparrow$ and $\Downarrow$ are flipped from the base word. The transformation T_2 reverses the order of the motion primitives. The transformation T_3 flips the directions of $\curvearrowleft$ and $\curvearrowright$. The transformations commute, and there are seven possible ways to combine them, which contributes to a row of Figure 15.13.

To construct an LPM or distance function, the same issues arise as for the Reeds-Shepp and Dubins cars. Rather than testing all 40 words to find the shortest path, simple tests can be defined over which a particular word becomes active [64]. A slice of the precise cell decomposition and the resulting optimal cost-to-go (which can be called the *Balkcom-Mason metric*) are shown in Figure 15.14.

15.4 Nonholonomic System Theory

This section gives some precision to the term *nonholonomic*, which was used loosely in Chapters 13 and 14. Furthermore, small-time controllability (STLC), which

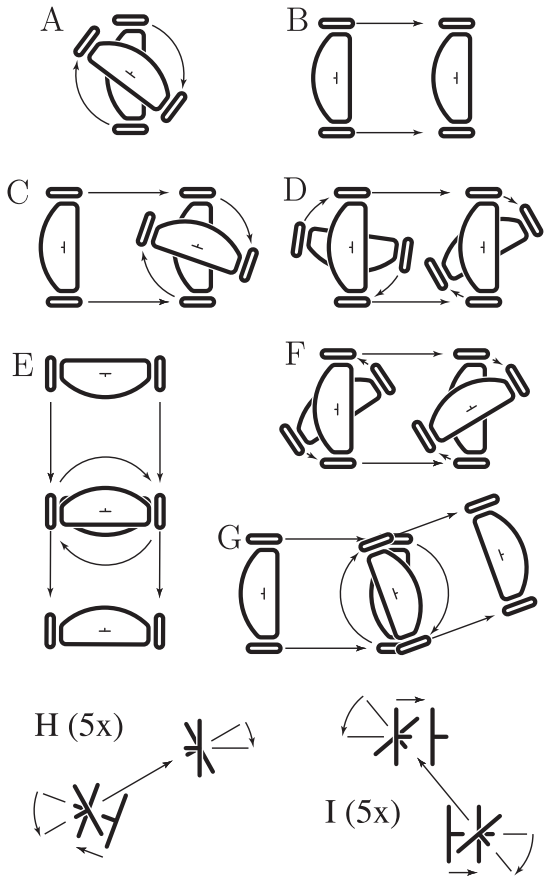

Figure 15.12: Each of the nine base words is depicted [64]. The last two are only valid for small motions; they are magnified five times and the robot outline is not drawn.

	Base	T_1	T_2	T_3	$T_2 \circ T_1$	$T_3 \circ T_1$	$T_3 \circ T_2$	$T_3 \circ T_2 \circ T_1$
A.	$\curvearrowright$	$\curvearrowright$	$\curvearrowright$	$\curvearrowleft$	$\curvearrowright$	$\curvearrowleft$	$\curvearrowleft$	$\curvearrowleft$
B.	$\Downarrow$	$\Uparrow$	$\Downarrow$	$\Downarrow$	$\Uparrow$	$\Uparrow$	$\Downarrow$	$\Uparrow$
C.	$\Downarrow\curvearrowright$	$\Uparrow\curvearrowright$	$\curvearrowright\Downarrow$	$\Downarrow\curvearrowleft$	$\curvearrowright\Uparrow$	$\Uparrow\curvearrowleft$	$\curvearrowleft\Downarrow$	$\curvearrowleft\Uparrow$
D.	$\curvearrowright\Downarrow\curvearrowright$	$\curvearrowright\Uparrow\curvearrowright$	$\curvearrowright\Downarrow\curvearrowright$	$\curvearrowleft\Downarrow\curvearrowleft$	$\curvearrowright\Uparrow\curvearrowright$	$\curvearrowleft\Uparrow\curvearrowleft$	$\curvearrowleft\Downarrow\curvearrowleft$	$\curvearrowleft\Uparrow\curvearrowleft$
E.	$\Uparrow\curvearrowleft_\pi\Downarrow$	$\Downarrow\curvearrowleft_\pi\Uparrow$	$\Downarrow\curvearrowleft_\pi\Uparrow$	$\Uparrow\curvearrowright_\pi\Downarrow$	$\Uparrow\curvearrowleft_\pi\Downarrow$	$\Downarrow\curvearrowright_\pi\Uparrow$	$\Downarrow\curvearrowright_\pi\Uparrow$	$\Uparrow\curvearrowright_\pi\Downarrow$
F.	$\curvearrowleft\Downarrow\curvearrowright$	$\curvearrowleft\Uparrow\curvearrowright$	$\curvearrowright\Downarrow\curvearrowleft$	$\curvearrowright\Downarrow\curvearrowleft$	$\curvearrowright\Uparrow\curvearrowleft$	$\curvearrowright\Uparrow\curvearrowleft$	$\curvearrowleft\Downarrow\curvearrowright$	$\curvearrowleft\Uparrow\curvearrowright$
G.	$\Downarrow\curvearrowright\Uparrow$	$\Uparrow\curvearrowright\Downarrow$	$\Uparrow\curvearrowright\Downarrow$	$\Downarrow\curvearrowleft\Uparrow$	$\Downarrow\curvearrowright\Uparrow$	$\Uparrow\curvearrowleft\Downarrow$	$\Uparrow\curvearrowleft\Downarrow$	$\Downarrow\curvearrowleft\Uparrow$
H.	$\curvearrowleft\Downarrow\curvearrowright\Uparrow$	$\curvearrowleft\Uparrow\curvearrowright\Downarrow$	$\Uparrow\curvearrowright\Downarrow\curvearrowleft$	$\curvearrowright\Downarrow\curvearrowleft\Uparrow$	$\Downarrow\curvearrowright\Uparrow\curvearrowleft$	$\curvearrowright\Uparrow\curvearrowleft\Downarrow$	$\Uparrow\curvearrowleft\Downarrow\curvearrowright$	$\Downarrow\curvearrowleft\Uparrow\curvearrowright$
I.	$\Uparrow\curvearrowright\Downarrow\curvearrowright\Uparrow$	$\Downarrow\curvearrowleft\Uparrow\curvearrowright\Downarrow$	$\Uparrow\curvearrowright\Downarrow\curvearrowleft\Uparrow$	$\Uparrow\curvearrowright\Downarrow\curvearrowleft\Uparrow$	$\Downarrow\curvearrowright\Uparrow\curvearrowleft\Downarrow$	$\Downarrow\curvearrowright\Uparrow\curvearrowleft\Downarrow$	$\Uparrow\curvearrowleft\Downarrow\curvearrowright\Uparrow$	$\Downarrow\curvearrowleft\Uparrow\curvearrowright\Downarrow$

Figure 15.13: The 40 optimal curve types for the differential-drive robot, sorted by symmetry class [64].

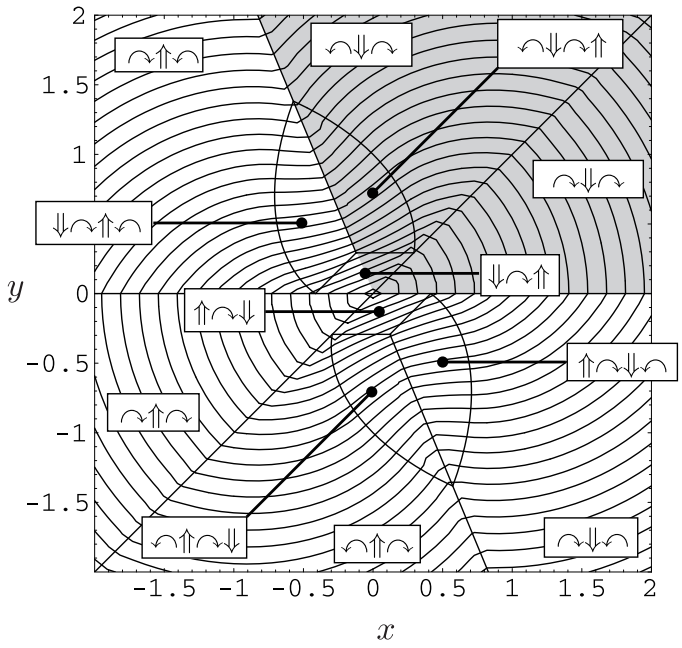

Figure 15.14: A slice of the optimal curves is shown for $q_I = (x, y, \frac{\pi}{4})$ and $q_G = (0, 0, 0)$ [64]. Level sets of the optimal cost-to-go G^* are displayed. The coordinates correspond to a differential drive with $r = L = 1$ in (13.16).

was defined in Section 15.1.3, is addressed. The presentation given here barely scratches the surface of this subject, which involves deep mathematical principles from differential geometry, algebra, control theory, and mechanics. The intention is to entice the reader to pursue further study of these topics; see the suggested literature at the end of the chapter.

15.4.1 Control-Affine Systems

Nonholonomic system theory is restricted to a special class of nonlinear systems. The techniques of Section 15.4 utilize ideas from linear algebra. The main concepts will be formulated in terms of linear combinations of vector fields on a smooth manifold X. Therefore, the formulation is restricted to *control-affine systems*, which were briefly introduced in Section 13.2.3. For these systems, $\dot{x} = f(x, u)$ is of the form

$$\dot{x} = h_0(x) + \sum_{i=1}^{m} h_i(x) u_i, \tag{15.52}$$

in which each h_i is a vector field on X.

The vector fields are expressed using a coordinate neighborhood of X. Usually, $m < n$, in which n is the dimension of X. Unless otherwise stated, assume that $U = \mathbb{R}^m$. In some cases, U may be restricted.

Each action variable $u_i \in \mathbb{R}$ can be imagined as a coefficient that determines how much of $h_i(x)$ is blended into the result $\dot{x}$. The *drift term* $h_0(x)$ always remains and is often such a nuisance that the *driftless* case will be the main focus. This means that $h_0(x) = 0$ for all $x \in X$, which yields

$$\dot{x} = \sum_{i=1}^{m} h_i(x) u_i. \tag{15.53}$$

The driftless case will be used throughout most of this section. The set $h_1, \ldots, h_m$, is referred to as the *system vector fields*. It is essential that U contains at least an open set that contains the origin of $\mathbb{R}^m$. If the origin is not contained in U, then the system is no longer driftless.[7]

Control-affine systems arise in many mechanical systems. Velocity constraints on the C-space frequently are of the Pfaffian form (13.5). In Section 13.1.1, it was explained that under such constraints, a configuration transition equation (13.6) can be derived that is linear if q is fixed. This is precisely the driftless form (15.53) using $X = \mathcal{C}$. Most of the models in Section 13.1.2 can be expressed in this form. The Pfaffian constraints on configuration are often called *kinematic constraints* because they arise due to the kinematics of bodies in contact, such as a wheel rolling. The more general case of (15.52) for a phase space X arises from *dynamic constraints* that are obtained from Euler-Lagrange equation (13.118) or Hamilton's equations (13.198) in the formulation of the mechanics. These constraints capture conservation laws, and the drift term usually appears due to momentum.

Example 15.5 (A Simplified Model for Differential Drives and Cars) Both the simple-car and the differential-drive models of Section 13.1.2 can be expressed in the form (15.53) after making simplifications. The simplified model, (15.48), can be adapted to conveniently express versions of both of them by using different restrictions to define U. The third equation of (15.48) can be reduced to $\dot{\theta} = u_2$ without affecting the set of velocities that can be achieved. To conform to (15.53), the equations can then be written in a linear-algebra form as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2. \tag{15.54}$$

This makes it clear that there are two system vector fields, which can be combined by selecting the scalar values u_1 and u_2. One vector field allows pure translation, and the other allows pure rotation. Without restrictions on U, this system behaves like a differential drive because the simple car cannot execute pure rotation. Simulating the simple car with (15.54) requires restrictions on U (such as requiring that u_1 be 1 or -1, as in Section 15.3.2). This is equivalent to the unicycle from Figure 13.5 and (13.18).

[7] Actually, if the convex hull of U contains an open set that contains the origin, then a driftless system can be simulated by rapid switching.

Note that (15.54) can equivalently be expressed as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{15.55}$$

by organizing the vector fields into a matrix. ■

In (15.54), the vector fields were written as column vectors that combine linearly using action variables. This suggested that control-affine systems can be alternatively expressed using matrix multiplication in (15.55). In general, the vector fields can be organized into an $n \times m$ matrix as

$$H(x) = \begin{bmatrix} h_1(x) & h_2(x) & \cdots & h_m(x) \end{bmatrix}. \tag{15.56}$$

In the driftless case, this yields

$$\dot{x} = H(x)\, u \tag{15.57}$$

as an equivalent way to express (15.53)

It is sometimes convenient to work with Pfaffian constraints,

$$g_1(x)\dot{x}_1 + g_2(x)\dot{x}_2 + \cdots + g_n(x)\dot{x}_n = 0, \tag{15.58}$$

instead of a state transition equation. As indicated in Section 13.1.1, a set of k independent Pfaffian constraints can be converted into a state transition equation with $m = (n - k)$ action variables. The resulting state transition equation is a driftless control-affine system. Thus, Pfaffian constraints provide a dual way of specifying driftless control-affine systems. The k Pfaffian constraints can be expressed in matrix form as

$$G(x)\, \dot{x} = 0, \tag{15.59}$$

which is the dual of (15.57), and $G(x)$ is a $k \times n$ matrix formed from the g_i coefficients of each Pfaffian constraint. Systems with drift can be expressed in a Pfaffian-like form by constraints

$$g_0(x) + g_1(x)\dot{x}_1 + g_2(x)\dot{x}_2 + \cdots + g_n(x)\dot{x}_n = 0. \tag{15.60}$$

15.4.2 Determining Whether a System Is Nonholonomic

The use of linear algebra in Section 15.4.1 suggests further development of algebraic concepts. This section briefly introduces concepts that resemble ordinary linear algebra but apply to linear combinations of vector fields. This provides the concepts and tools needed to characterize important system properties in the remainder of this section. This will enable the assessment of whether a system is nonholonomic and also whether it is STLC. Many of the constructions are named after Sophus Lie (pronounced "lee"), a mathematician who in the nineteenth century contributed many ideas to algebra and geometry that happen to be relevant in the study of nonholonomic systems (although that application came much later).

15.4.2.1 Completely integrable or nonholonomic?

Every control-affine system must be one or the other (not both) of the following:

1. **Completely integrable:** This means that the Pfaffian form (15.59) can be obtained by differentiating k equations of the form $f_i(x) = 0$ with respect to time. This case was interpreted as being trapped on a surface in Section 13.1.3. An example of being trapped on a circle in $\mathbb{R}^2$ was given in (13.22).

2. **Nonholonomic:** This means that the system is not completely integrable. In this case, it might even be possible to reach all of X, even if the number of action variables m is much smaller than n, the dimension of X.

In this context, the term *holonomic* is synonymous with completely integrable, and *nonintegrable* is synonymous with nonholonomic. The term nonholonomic is sometimes applied to non-Pfaffian constraints [588]; however, this will be avoided here, in accordance with mechanics literature [112].

The notion of integrability used here is quite different from that required for (14.1). In that case, the state transition equation needed to be integrable to obtain integral curves from any initial state. This was required for all systems considered in this book. By contrast, *complete integrability* implies that the system can be expressed without even using derivatives. This means that all integral curves can eventually be characterized by constraints that do not involve derivatives.

To help understand complete integrability, the notion of an integral curve will be generalized from one to m dimensions. A manifold $M \subseteq X$ is called an *integral manifold* of a set of Pfaffian constraints if at every $x \in M$, all vectors in the tangent space $T_x(M)$ satisfy the constraints. For a set of completely integrable Pfaffian constraints, a partition of X into integral manifolds can be obtained by defining maximal integral manifolds from every $x \in X$. The resulting partition is called a *foliation*, and the maximal integral manifolds are called *leaves* [872].

Example 15.6 (A Foliation with Spherical Leaves) As an example, suppose $X = \mathbb{R}^n$ and consider the Pfaffian constraint

$$x_1\dot{x}_1 + x_2\dot{x}_2 + \cdots x_n\dot{x}_n = 0. \tag{15.61}$$

This is completely integrable because it can be obtained by differentiating the equation of a sphere,

$$x_1^2 + x_2^2 + \cdots x_n^2 - r^2 = 0, \tag{15.62}$$

with respect to time (r is a constant). The particular sphere that is obtained via integration depends on an initial state. The foliation is the collection of all concentric spheres that are centered at the origin. For example, if $X = \mathbb{R}^3$, then a maximal integral manifold arises for each point of the form $(0, 0, r)$. In each case, it is a sphere of radius r. The foliation is generated by selecting every $r \in [0, \infty)$. ■

The task in this section is to determine whether a system is completely integrable. Imagine someone is playing a game with you. You are given an control-affine system and asked to determine whether it is completely integrable. The person playing the game with you can start with equations of the form $h_i(x) = 0$ and differentiate them to obtain Pfaffian constraints. These can then be converted into the parametric form to obtain the state transition equation (15.53). It is easy to construct challenging problems; however, it is very hard to solve them. The concepts in this section can be used to determine only whether it is possible to win such a game. The main tool will be the Frobenius theorem, which concludes whether a system is completely integrable. Unfortunately, the conclusion is obtained without producing the integrated constraints $h_i(x) = 0$. Therefore, it is important to keep in mind that "integrability" does not mean that *you* can integrate it to obtain a nice form. This is a challenging problem of reverse engineering. On the other hand, it is easy to go in the other direction by differentiating the constraints to make a challenging game for someone else to play.

15.4.2.2 Distributions

A distribution[8] expresses a set of vector fields on a smooth manifold. Suppose that a driftless control-affine system (15.53) is given. Recall the vector space definition from Section 8.3.1 or from linear algebra. Also recall that a state transition equation can be interpreted as a vector field if the actions are fixed and as a vector space if the state is instead fixed. For $U = \mathbb{R}^m$ and a fixed $x \in X$, the state transition equation defines a vector space in which each h_i evaluated at x is a basis vector and each u_i is a coefficient. For example, in (15.54), the vector fields h_1 and h_2 evaluated at $q = (0, 0, 0)$ become $[1 \;\; 0 \;\; 0]^T$ and $[0 \;\; 0 \;\; 1]^T$, respectively. These serve as the basis vectors. By selecting values of $u \in \mathbb{R}^2$, a 2D vector space results. Any vector of the form $[a \;\; 0 \;\; b]^T$ can be represented by setting $u_1 = a$ and $u_2 = b$. More generally, let $\triangle(x)$ denote the vector space obtained in this way for any $x \in X$.

The dimension of a vector space is the number of independent basis vectors. Therefore, the dimension of $\triangle(x)$ is the rank of $H(x)$ from (15.56) when evaluated at the particular $x \in X$. Now consider defining $\triangle(x)$ for every $x \in X$. This yields a parameterized family of vector spaces, one for each $x \in X$. The result could just as well be interpreted as a parameterized family of vector fields. For example, consider actions for i from 1 to m of the form $u_i = 1$ and $u_j = 0$ for all $i \neq j$. If the action is held constant over all $x \in X$, then it selects a single vector field h_i from the collection of m vector fields:

$$\dot{x} = h_i(x). \tag{15.63}$$

Using constant actions, an m-dimensional vector space can be defined in which each vector field h_i is a basis vector (assuming the h_i are linearly independent),

[8]This distribution has nothing to do with probability theory. It is just an unfortunate coincidence of terminology.

and the $u_i \in \mathbb{R}$ are the coefficients:

$$u_1 h_1(x) + u_2 h_2(x) + \cdots + u_m h_m(x). \tag{15.64}$$

This idea can be generalized to allow the u_i to vary over X. Thus, rather than having u constant, it can be interpreted as a feedback plan $\pi : X \rightarrow U$, in which the action at x is given by $u = \pi(x)$. The set of all vector fields that can be obtained as

$$\pi_1(x) h_1(x) + \pi_2(x) h_2(x) + \cdots + \pi_m(x) h_m(x) \tag{15.65}$$

is called the *distribution* of the set $\{h_1, \ldots, h_m\}$ of vector fields and is denoted as $\triangle$. If $\triangle$ is obtained from an control-affine system, then $\triangle$ is called the *system distribution*. The resulting set of vector fields is not quite a vector space because the nonzero coefficients π_i do not necessarily have a multiplicative inverse. This is required for the coefficients of a vector field and was satisfied by using $\mathbb{R}$ in the case of constant actions. A distribution is instead considered algebraically as a *module* [469]. In most circumstances, it is helpful to imagine it as a vector space (just do not try to invert the coefficients!). Since a distribution is almost a vector space, the span notation from linear algebra is often used to define it:

$$\triangle = \text{span}\{h_1, h_2, \ldots, h_m\}. \tag{15.66}$$

Furthermore, it *is* actually a vector space with respect to constant actions $u \in \mathbb{R}^m$. Note that for each fixed $x \in X$, the vector space $\triangle(x)$ is obtained, as defined earlier. A vector field f is said to *belong* to a distribution $\triangle$ if it can be expressed using (15.65). If for all $x \in X$, the dimension of $\triangle(x)$ is m, then $\triangle$ is called a *nonsingular distribution* (or *regular distribution*). Otherwise, $\triangle$ is called a *singular distribution*, and the points $x \in X$ for which the dimension of $\triangle(x)$ is less than m are called *singular points*. If the dimension of $\triangle(x)$ is a constant c over all $x \in X$, then c is called the *dimension* of the distribution and is denoted by $\dim(\triangle)$. If the vector fields are smooth, and if π is restricted to be smooth, then a *smooth distribution* is obtained, which is a subset of the original distribution.

As depicted in Figure 15.15, a nice interpretation of the distribution can be given in terms of the tangent bundle of a smooth manifold. The tangent bundle was defined for $X = \mathbb{R}^n$ in (8.9) and generalizes to any smooth manifold X to obtain

$$T(X) = \bigcup_{x \in X} T_x(X). \tag{15.67}$$

The tangent bundle is a $2n$-dimensional manifold in which n is the dimension of X. A phase space for which $x = (q, \dot{q})$ is actually $T(\mathcal{C})$. In the current setting, each element of $T(X)$ yields a state and a velocity, $(x, \dot{x})$. Which pairs are possible for a driftless control-affine system? Each $\triangle(x)$ indicates the set of possible $\dot{x}$ values for a fixed x. The point x is sometimes called the *base* and $\triangle(x)$ is called the *fiber* over x in $T(X)$. The distribution $\triangle$ simply specifies a subset of $T_x(X)$ for every $x \in X$. For a vector field f to belong to $\triangle$, it must satisfy $f(x) \in \triangle(x)$ for all $x \in X$. This is just a restriction to a subset of $T(X)$. If $m = n$ and the

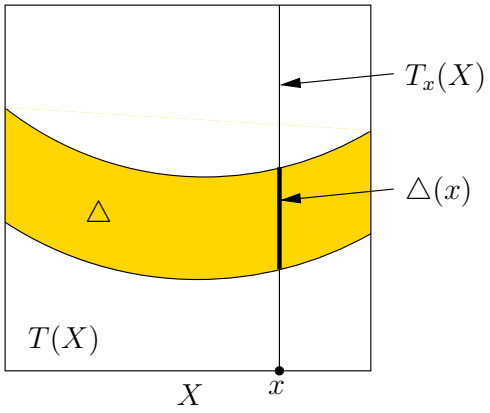

Figure 15.15: The distribution $\triangle$ can be imagined as a slice of the tangent bundle $T(X)$. It restricts the tangent space at every $x \in X$.

system vector fields are independent, then any vector field is allowed. In this case, $\triangle$ includes any vector field that can be constructed from the vectors in $T(X)$.

Example 15.7 (The Distribution for the Differential Drive) The system in (15.54) yields a two-dimensional distribution:

$$\triangle = \text{span}\{[\cos\theta \quad \sin\theta \quad 0]^T, \ [0 \quad 0 \quad 1]^T\}. \tag{15.68}$$

The distribution is nonsingular because for any (x, y, θ) in the coordinate neighborhood, the resulting vector space $\triangle(x, y, \theta)$ is two-dimensional. ■

Example 15.8 (A Singular Distribution) Consider the following system, which is given in [478]:

$$\begin{aligned} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} &= h_1(x)u_1 + h_2(x)u_2 + h_3(x)u_3 \\ &= \begin{pmatrix} x_1 \\ 1 + x_3 \\ 1 \end{pmatrix} u_1 + \begin{pmatrix} x_1 x_2 \\ (1 + x_3)x_2 \\ x_2 \end{pmatrix} u_2 + \begin{pmatrix} x_1 \\ x_1 \\ 0 \end{pmatrix} u_3. \end{aligned} \tag{15.69}$$

The distribution is

$$\triangle = \text{span}\{h_1, h_2, h_3\}. \tag{15.70}$$

The first issue is that for any $x \in \mathbb{R}^3$, $h_2(x) = h_1(x)x_2$, which implies that the vector fields are linearly dependent over all of $\mathbb{R}^3$. Hence, this distribution is singular because $m = 3$ and the dimension of $\Delta(x)$ is 2 if $x_1 \neq 0$. If $x_1 = 0$, then the dimension of $\Delta(x)$ drops to 1. The dimension of $\triangle$ is not defined because the dimension of $\Delta(x)$ depends on x. ■

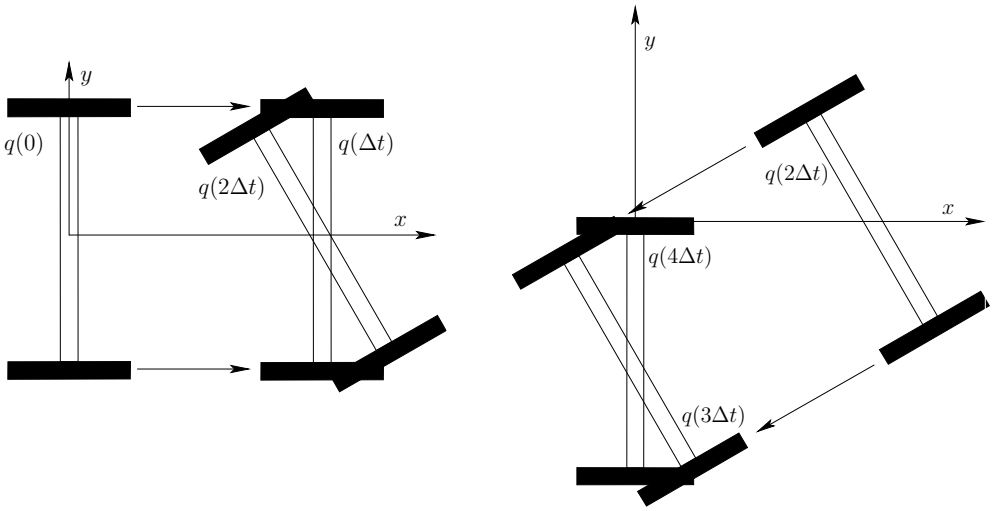

Figure 15.16: (a) The effect of the first two primitives. (b) The effect of the last two primitives.

A distribution can alternatively be defined directly from Pfaffian constraints. Each $g_i(x) = 0$ is called an *annihilator* because enforcing the constraint eliminates many vector fields from consideration. At each $x \in X$, $\triangle(x)$ is defined as the set of all velocity vectors that satisfy all k Pfaffian constraints. The constraints themselves can be used to form a *codistribution*, which is a kind of dual to the distribution. The codistribution can be interpreted as a vector space in which each constraint is a basis vector. Constraints can be added together or multiplied by any $c \in \mathbb{R}$, and there is no effect on the resulting distribution of allowable vector fields.

15.4.2.3 Lie brackets

The key to establishing whether a system is nonholonomic is to construct motions that combine the effects of two action variables, which may produce motions in a direction that seems impossible from the system distribution. To motivate the coming ideas, consider the differential-drive model from (15.54). Apply the following piecewise-constant action trajectory over the interval $[0, 4\Delta t]$:

$$u(t) = \begin{cases} (1,0) & \text{for } t \in [0, \Delta t) \\ (0,1) & \text{for } t \in [\Delta t, 2\Delta t) \\ (-1,0) & \text{for } t \in [2\Delta t, 3\Delta t) \\ (0,-1) & \text{for } t \in [3\Delta t, 4\Delta t] \ . \end{cases} \tag{15.71}$$

The action trajectory is a sequence of four motion primitives: 1) translate forward, 2) rotate forward, 3) translate backward, and 4) rotate backward.

The result of all four motion primitives in succession from $q_I = (0, 0, 0)$ is shown in Figure 15.16. It is fun to try this at home with an axle and two wheels (Tinkertoys work well, for example). The result is that the differential drive moves sideways![9] From the transition equation (15.54) such motions appear impossible. This is a beautiful property of nonlinear systems. The state may wiggle its way in directions that do not seem possible. A more familiar example is parallel parking a car. It is known that a car cannot directly move sideways; however, some wiggling motions can be performed to move it sideways into a tight parking space. The actions we perform while parking resemble the primitives in (15.71).

Algebraically, the motions of (15.71) appear to be checking for commutativity. Recall from Section 4.2.1 that a group G is called *commutative* (or *Abelian*) if $ab = ba$ for any $a, b \in G$. A *commutator* is a group element of the form $aba^{-1}b^{-1}$. If the group is commutative, then $aba^{-1}b^{-1} = e$ (the identity element) for any $a, b \in G$. If a nonidentity element of G is produced by the commutator, then the group is not commutative. Similarly, if the trajectory arising from (15.71) does not form a loop (by returning to the starting point), then the motion primitives do not commute. Therefore, a sequence of motion primitives in (15.71) will be referred to as the *commutator motion*. It will turn out that if the commutator motion cannot produce any velocities not allowed by the system distribution, then the system is completely integrable. This means that if we are trapped on a surface, then it is impossible to leave the surface by using commutator motions.

Now generalize the differential drive to any driftless control-affine system that has two action variables:

$$\dot{x} = f(x)u_1 + g(x)u_2. \tag{15.72}$$

Using the notation of (15.53), the vector fields would be h_1 and h_2; however, f and g are chosen here to allow subscripts to denote the components of the vector field in the coming explanation.

Suppose that the commutator motion (15.71) is applied to (15.72) as shown in Figure 15.17. Determining the resulting motion requires some general computations, as opposed to the simple geometric arguments that could be made for the differential drive. If would be convenient to have an expression for the velocity obtained in the limit as Δt approaches zero. This can be obtained by using Taylor series arguments. These are simplified by the fact that the action history is piecewise constant.

The coming derivation will require an expression for $\ddot{x}$ under the application of a constant action. For each action, a vector field of the form $\dot{x} = h(x)$ is obtained. Upon differentiation, this yields

$$\ddot{x} = \frac{dh}{dt} = \frac{\partial h}{\partial x}\frac{dx}{dt} = \frac{\partial h}{dx}\dot{x} = \frac{\partial h}{dx}h(x). \tag{15.73}$$

This follows from the chain rule because h is a function of x, which itself is a function of t. The derivative $\partial h/\partial x$ is actually an $n \times n$ Jacobian matrix, which

[9] It also moves slightly forward; however, this can be eliminated by either lengthening the time of the third primitive or by considering the limit as Δ approaches zero.

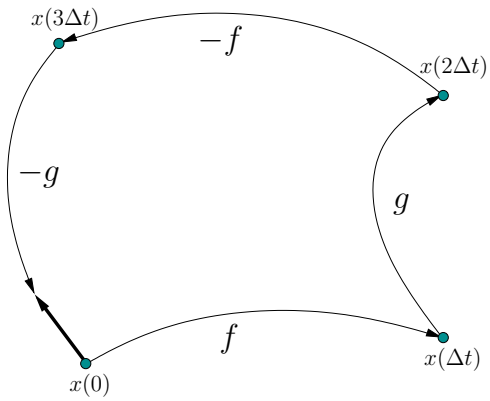

Figure 15.17: The velocity obtained by the Lie bracket can be approximated by a sequence of four motion primitives.

is multiplied by the vector $\dot{x}$. To further clarify (15.73), each component can be expressed as

$$\ddot{x}_i = \frac{d}{dt} h_i(x(t)) = \sum_{j=1}^{n} \frac{\partial h_i}{\partial x_j} h_j. \tag{15.74}$$

Now the state trajectory under the application of (15.71) will be determined using the Taylor series, which was given in (14.17). The state trajectory that results from the first motion primitive $u = (1, 0)$ can be expressed as

$$\begin{aligned} x(\Delta t) &= x(0) + \Delta t\ \dot{x}(0) + \tfrac{1}{2}(\Delta t)^2\ \ddot{x}(0) + \cdots \\ &= x(0) + \Delta t\ f(x(0)) + \frac{1}{2}(\Delta t)^2\ \frac{\partial f}{\partial x}\Big|_{x(0)}\ f(x(0)) + \cdots, \end{aligned} \tag{15.75}$$

which makes use of (15.73) in the second line. The Taylor series expansion for the second primitive is

$$x(2\Delta t) = x(\Delta t) + \Delta t\ g(x(\Delta t)) + \frac{1}{2}(\Delta t)^2\ \frac{\partial g}{\partial x}\Big|_{x(\Delta t)}\ g(x(\Delta t)) + \cdots. \tag{15.76}$$

An expression for $g(x(\Delta t))$ can be obtained by using the Taylor series expansion in (15.75) to express $x(\Delta t)$. The first terms after substitution and simplification are

$$x(2\Delta t) = x(0) + \Delta t\ (f + g) + (\Delta t)^2 \left(\frac{1}{2} \frac{\partial f}{\partial x} f + \frac{\partial g}{\partial x} f + \frac{1}{2} \frac{\partial g}{\partial x} g \right) + \cdots. \tag{15.77}$$

To simplify the expression, the evaluation at $x(0)$ has been dropped from every occurrence of f and g and their derivatives.

The idea of substituting previous Taylor series expansions as they are needed can be repeated for the remaining two motion primitives. The Taylor series expansion for the result after the third primitive is

$$x(3\Delta t) = x(0) + \Delta t\ g + (\Delta t)^2 \left(\frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g + \frac{1}{2} \frac{\partial g}{\partial x} g \right) + \cdots. \tag{15.78}$$

Finally, the Taylor series expansion after all four primitives have been applied is

$$x(4\Delta t) = x(0) + (\Delta t)^2 \left(\frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g \right) + \cdots . \tag{15.79}$$

Taking the limit yields

$$\lim_{\Delta t \to 0} \frac{x(4\Delta t) - x(0)}{(\Delta t)^2} = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g, \tag{15.80}$$

which is called the *Lie bracket* of f and g and is denoted by $[f, g]$. Similar to (15.74), the ith component can be expressed as

$$[f, g]_i = \sum_{j=1}^{n} \left(f_j \frac{\partial g_i}{\partial x_j} - g_j \frac{\partial f_i}{\partial x_j} \right). \tag{15.81}$$

The Lie bracket is an important operation in many subjects, and is related to the Poisson and Jacobi brackets that arise in physics and mathematics.

Example 15.9 (Lie Bracket for the Differential Drive) The Lie bracket should indicate that sideways motions are possible for the differential drive. Consider taking the Lie bracket of the two vector fields used in (15.54). Let $f = [\cos\theta \;\; \sin\theta \;\; 0]^T$ and $g = [0 \;\; 0 \;\; 1]^T$. Rename h_1 and h_2 to f and g to allow subscripts to denote the components of a vector field.

By applying (15.81), the Lie bracket $[f, g]$ is

$$\begin{aligned}
[f, g]_1 &= f_1 \frac{\partial g_1}{\partial x} - g_1 \frac{\partial f_1}{\partial x} + f_2 \frac{\partial g_1}{\partial y} - g_2 \frac{\partial f_1}{\partial y} + f_3 \frac{\partial g_1}{\partial \theta} - g_3 \frac{\partial f_1}{\partial \theta} = \sin\theta \\
[f, g]_2 &= f_1 \frac{\partial g_2}{\partial x} - g_1 \frac{\partial f_2}{\partial x} + f_2 \frac{\partial g_2}{\partial y} - g_2 \frac{\partial f_2}{\partial y} + f_3 \frac{\partial g_2}{\partial \theta} - g_3 \frac{\partial f_2}{\partial \theta} = -\cos\theta \\
[f, g]_3 &= f_1 \frac{\partial g_3}{\partial x} - g_1 \frac{\partial f_3}{\partial x} + f_2 \frac{\partial g_3}{\partial y} - g_2 \frac{\partial f_3}{\partial y} + f_3 \frac{\partial g_3}{\partial \theta} - g_3 \frac{\partial f_3}{\partial \theta} = 0.
\end{aligned} \tag{15.82}$$

The resulting vector field is $[f, g] = [\sin\theta \;\; -\cos\theta \;\; 0]^T$, which indicates the sideways motion, as desired. When evaluated at $q = (0, 0, 0)$, the vector $[0 \;\; -1 \;\; 0]^T$ is obtained. This means that performing short commutator motions wiggles the differential drive sideways in the $-y$ direction, which we already knew from Figure 15.16. ■

Example 15.10 (Lie Bracket of Linear Vector Fields) Suppose that each vector field is a linear function of x. The $n \times n$ Jacobians $\partial f / \partial x$ and $\partial g / \partial x$ are constant.

As a simple example, recall the nonholonomic integrator (13.43). In the linear-algebra form, the system is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -x_2 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 1 \\ x_1 \end{pmatrix} u_2. \tag{15.83}$$

Let $f = h_1$ and $g = h_2$. The Jacobian matrices are

$$\frac{\partial f}{\partial x} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial g}{\partial x} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{15.84}$$

Using (15.80),

$$\frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -x_2 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ -x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}. \tag{15.85}$$

This result can be verified using (15.81).

■

15.4.2.4 The Frobenius Theorem

The Lie bracket is the only tool needed to determine whether a system is completely integrable (holonomic) or nonholonomic (not integrable). Suppose that a system of the form (15.53) is given. Using the m system vector fields $h_1, \ldots, h_m$ there are $\binom{m}{2}$ Lie brackets of the form $[h_i, h_j]$ for $i < j$ that can be formed. A distribution $\triangle$ is called *involutive* [133] if for each of these brackets there exist m coefficients $c_k \in \mathbb{R}$ such that

$$[h_i, h_j] = \sum_{k=1}^{m} c_k h_k. \tag{15.86}$$

In other words, every Lie bracket can be expressed as a linear combination of the system vector fields, and therefore it already belongs to $\triangle$. The Lie brackets are unable to escape $\triangle$ and generate new directions of motion. We did not need to consider all n^2 possible Lie brackets of the system vector fields because it turns out that $[h_i, h_j] = -[h_j, h_i]$ and consequently $[h_i, h_i] = 0$. Therefore, the definition of involutive is not altered by looking only at the $\binom{m}{2}$ pairs.

If the system is smooth and the distribution is nonsingular, then the Frobenius theorem immediately characterizes integrability:

A system is completely integrable if and only if it is involutive.

Proofs of the Frobenius theorem appear in numerous differential geometry and control theory books [133, 156, 478, 846]. There also exist versions that do not require the distribution to be nonsingular.

Determining integrability involves performing Lie brackets and determining whether (15.86) is satisfied. The search for the coefficients can luckily be avoided by using linear algebra tests for linear independence. The $n \times m$ matrix $H(x)$, which was defined in (15.56), can be augmented into an $n \times (m+1)$ matrix $H'(x)$ by adding $[h_i, h_j]$ as a new column. If the rank of $H'(x)$ is $m+1$ for any pair h_i and h_j, then it is immediately known that the system is nonholonomic. If the rank of $H'(x)$ is m for all Lie brackets, then the system is completely integrable. Driftless linear systems, which are expressed as $\dot{x} = Bu$ for a fixed matrix B, are completely integrable because all Lie brackets are zero.

Example 15.11 (The Differential Drive Is Nonholonomic) For the differential drive model in (15.54), the Lie bracket $[f, g]$ was determined in Example 15.9 to be $[\sin\theta \quad -\cos\theta \quad 0]^T$. The matrix $H'(q)$, in which $q = (x, y, \theta)$, is

$$H'(q) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta & 0 & -\cos\theta \\ 0 & 1 & 0 \end{pmatrix}. \tag{15.87}$$

The rank of $H'(q)$ is 3 for all $q \in \mathcal{C}$ (the determinant of $H'(q)$ is 1). Therefore, by the Frobenius theorem, the system is nonholonomic. ■

Example 15.12 (The Nonholonomic Integrator Is Nonholonomic) We would hope that the nonholonomic integrator is nonholonomic. In Example 15.10, the Lie bracket was determined to be $[0 \;\; 0 \;\; 2]^T$. The matrix $H'(q)$ is

$$H'(q) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_2 & x_1 & 2 \end{pmatrix}, \tag{15.88}$$

which clearly has full rank for all $q \in \mathcal{C}$. ■

Example 15.13 (Trapped on a Sphere) Suppose that the following system is given:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} x_2 \\ -x_1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} x_3 \\ 0 \\ -x_1 \end{pmatrix} u_2, \tag{15.89}$$

for which $X = \mathbb{R}^3$ and $U = \mathbb{R}^2$. Since the vector fields are linear, the Jacobians are constant (as in Example 15.10):

$$\frac{\partial f}{\partial x} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial g}{\partial x} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}. \tag{15.90}$$

Using (15.80),

$$\frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ -x_1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_3 \\ 0 \\ -x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ x_3 \\ -x_2 \end{pmatrix}. \quad (15.91)$$

This yields the matrix

$$H'(x) = \begin{pmatrix} x_2 & -x_1 & 0 \\ x_3 & 0 & -x_1 \\ 0 & x_3 & -x_2 \end{pmatrix}. \quad (15.92)$$

The determinant is zero for all $x \in \mathbb{R}^3$, which means that $[f, g]$ is never linearly independent of f and g. Therefore, the system is completely integrable.[10]

The system can actually be constructed by differentiating the equation of a sphere. Let

$$f(x) = x_1^2 + x_2^2 + x_3^2 - r^2 = 0, \quad (15.93)$$

and differentiate with respect to time to obtain

$$x_1 \dot{x}_1 + x_2 \dot{x}_2 + x_3 \dot{x}_3 = 0, \quad (15.94)$$

which is a Pfaffian constraint. A parametric representation of the set of vectors that satisfy (15.94) is given by (15.89). For each $(u_1, u_2) \in \mathbb{R}^2$, (15.89) yields a vector that satisfies (15.94). Thus, this was an example of being trapped on a sphere, which we would expect to be completely integrable. It was difficult, however, to suspect this using only (15.89). ■

15.4.3 Determining Controllability

Determining complete integrability is the first step toward determining whether a driftless control-affine system is STLC. The Lie bracket attempts to produce motions in directions that do not seem to be allowed by the system distribution. At each q, a velocity not in $\triangle(q)$ may be produced by the Lie bracket. By working further with Lie brackets, it is possible to completely characterize *all* of the directions that are possible from each q. So far, the Lie brackets have only been applied to the system vector fields $h_1, \ldots, h_m$. It is possible to proceed further by applying Lie bracket operations on Lie brackets. For example, $[h_1, [h_1, h_2]]$ can be computed. This might generate a vector field that is linearly independent of all of the vector fields considered in Section 15.4.2 for the Frobenius theorem. The main idea in this section is to apply the Lie bracket recursively until no more independent vector fields can be found. The result is called the Lie algebra. If the number of independent vector fields obtained in this way is the dimension of X, then it turns out that the system is STLC.

[10]This system is singular at the origin. A variant of the Frobenius theorem given here is technically needed.

15.4.3.1 The Lie algebra

The notion of a Lie algebra is first established in general. Let V be any vector space with coefficients in $\mathbb{R}$. In V, the vectors can be added or multiplied by elements of $\mathbb{R}$; however, there is no way to "multiply" two vectors to obtain a third. The Lie algebra introduces a product operation to V. The product is called a *bracket* or *Lie bracket* (considered here as a generalization of the previous Lie bracket) and is denoted by $[\cdot, \cdot] : V \times V \rightarrow V$.

To be a *Lie algebra* obtained from V, the bracket must satisfy the following three axioms:

1. **Bilinearity:** For any $a, b \in \mathbb{R}$ and $u, v, w \in V$,

$$\begin{aligned} [au + bv, w] &= a[u, w] + b[v, w] \\ [u, av + bw] &= a[u, w] + b[u, w]. \end{aligned} \tag{15.95}$$

2. **Skew symmetry:** For any $u, v \in V$,

$$[u, v] = -[v, u]. \tag{15.96}$$

   This means that the bracket is anti-commutative.

3. **Jacobi identity:** For any $u, v, w \in V$,

$$[[u, v], w] + [[v, w], u] + [[w, u], v] = 0. \tag{15.97}$$

   Note that the bracket is not even associative.

Let $\mathcal{L}(V)$ denote the *Lie algebra* of V. This is a vector space that includes all elements of V and any new elements that can be obtained via Lie bracket operations. The Lie algebra $\mathcal{L}(V)$ includes every vector that can be obtained from any finite number of nested Lie bracket operations. Thus, describing a Lie algebra requires characterizing all vectors that are obtained under the algebraic closure of the bracket operation. Since $\mathcal{L}(V)$ is a vector space, this is accomplished by finding a basis of independent vectors of which all elements of $\mathcal{L}(V)$ can be expressed as a linear combination.

Example 15.14 (The Vector Cross Product) Let V be the vector space over $\mathbb{R}^3$ that is used in vector calculus. The basis elements are often denoted as $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$. A bracket for this vector space is simply the cross product

$$[u, v] = u \times v. \tag{15.98}$$

It can be verified that the required axioms of a Lie bracket are satisfied.

One interesting property of the cross product that is exploited often in analytic geometry is that it produces a vector outside of the span of u and v. For example, let W be the two-dimensional subspace of vectors

$$W = \text{span}\{\hat{\imath}, \hat{\jmath}\}. \tag{15.99}$$

The cross product always yields a vector that is a multiple of $\hat{k}$, which lies outside of V if the product is nonzero. This behavior is very similar to constructing vector fields that lie outside of $\triangle$ using the Lie bracket in Section 15.4.2. ■

Example 15.15 (Lie Algebra on Lie Groups) Lie groups are the most important application of the Lie algebra concepts. Recall from Section 4.2.1 the notion of a matrix group. Important examples throughout this book have been $SO(n)$ and $SE(n)$. If interpreted as a smooth manifold, these matrix groups are examples of *Lie groups* [63]. In general, a *Lie group* G is both a differentiable manifold and a group with respect to some operation $\circ$ if and only if:

1. The product $a \circ b$, interpreted as a function from $G \times G \rightarrow G$, is smooth.
2. The inverse a^{-1}, interpreted as a function from G to G, is smooth.

The two conditions are needed to prevent the group from destroying the nice properties that come with the smooth manifold. An important result in the study of Lie groups is that all compact finite-dimensional Lie groups can be represented as matrix groups.

For any Lie group, a Lie algebra can be defined on a special set of vector fields. These are defined using the *left translation* mapping $L_g : x \mapsto gx$. The vector field formed from the differential of L_g is called a *left-invariant vector field*. A Lie algebra can be defined on the set of these fields. The Lie bracket definition depends on the particular group. For the case of $GL(n)$, the Lie bracket is

$$[A, B] = AB - BA. \tag{15.100}$$

In this case, the Lie bracket clearly appears to be a test for commutativity. If the matrices commute with respect to multiplication, then the Lie bracket is zero. The Lie brackets for $SO(n)$ and $SE(n)$ are given in many texts on mechanics and control [156, 846]. The Lie algebra of left-invariant vector fields is an important structure in the study of nonlinear systems and mechanics. ■

15.4.3.2 Lie algebra of the system distribution

Now suppose that a set $h_1, \ldots, h_m$ of vector fields is given as a driftless control-affine system, as in (15.53). Its associated distribution $\triangle$ is interpreted as a vector space with coefficients in $\mathbb{R}$, and the Lie bracket operation was given by (15.81). It can be verified that the Lie bracket operation in (15.81) satisfies the required axioms for a Lie algebra.

As observed in Examples 15.9 and 15.10, the Lie bracket may produce vector fields outside of $\triangle$. By defining the Lie algebra of $\triangle$ to be all vector fields that can be obtained by applying Lie bracket operations, a potentially larger distribution $\mathcal{L}(\triangle)$ is obtained. The Lie algebra can be expressed using the span notation by including $h_1, \ldots, h_m$ and all independent vector fields generated by Lie brackets. Note that no more than n independent vector fields can possibly be produced.

Example 15.16 (The Lie Algebra of the Differential Drive) The Lie algebra of the differential drive (15.54) is

$$\mathcal{L}(\triangle) = \text{span}\{[\cos\theta \;\; \sin\theta \;\; 0]^T, \; [0 \;\; 0 \;\; 1]^T, \; [\sin\theta \;\; -\cos\theta \;\; 0]^T\}. \tag{15.101}$$

This uses the Lie bracket that was computed in (15.82) to obtain a three-dimensional Lie algebra. No further Lie brackets are needed because the maximum number of independent vector fields has been already obtained. ■

Example 15.17 (A Lie Algebra That Involves Nested Brackets) The previous example was not very interesting because the Lie algebra was generated after computing only one bracket. Suppose that $X = \mathbb{R}^5$ and $U = \mathbb{R}^2$. In this case, there is room to obtain up to three additional, linearly independent vector fields. The dimension of the Lie algebra may be any integer from 2 to 5.

Let the system be

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} u_2. \tag{15.102}$$

This is a *chained-form system*, which is a concept that becomes important in Section 15.5.2.

The first Lie bracket produces

$$[h_1, h_2] = [0 \;\; 0 \;\; -1 \;\; 0 \;\; 0]^T. \tag{15.103}$$

Other vector fields that can be obtained by Lie brackets are

$$[h_1, [h_1, h_2]] = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 0]^T \tag{15.104}$$

and

$$[h_1, [h_1, [h_1, h_2]]] = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 1]^T. \tag{15.105}$$

The resulting five vector fields are independent over all $x \in \mathbb{R}^5$. This includes h_1, h_2, and the three obtained from Lie bracket operations. Independence can be established by placing them into a 5×5 matrix,

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ x_2 & 0 & -1 & 0 & 0 \\ x_3 & 0 & 0 & 1 & 0 \\ x_4 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{15.106}$$

which has full rank for all $x \in \mathbb{R}^5$. No additional vector fields can possibly be independent. Therefore, the five-dimensional Lie algebra is

$$\mathcal{L}(\triangle) = \text{span}\{h_1, \; h_2, \; [h_1, h_2], \; [h_1, [h_1, h_2]], \; [h_1, [h_1, [h_1, h_2]]]\}. \tag{15.107}$$

■

15.4.3.3 Philip Hall basis of a Lie algebra

Determining the basis of a Lie algebra may be a long and tedious process. The combinations of Lie brackets in Example 15.17 were given; however, it is not known in advance which ones will produce independent vector fields. Numerous Lie brackets may be needed, including some that are nested, such as $[[h_1, h_2], h_3]$. The maximum depth of nested Lie bracket operations is not known a priori. Therefore, a systematic search must be performed (this can in fact be modeled as a discrete planning problem) by starting with h_1, ..., h_m and iteratively generating new, independent vector fields using Lie brackets.

One popular approach is to generate the *Philip Hall basis* (or *P. Hall basis*) of the Lie algebra $\mathcal{L}(\triangle)$. The construction of the basis essentially follows breadth-first search, in which the search depth is defined to be the number of nested levels of bracket operations. The *order* (or *depth*) d of a Lie product is defined recursively as follows. For the base case, let $d(h_i) = 1$ for any of the system vector fields. For any Lie product $[\phi_1, \phi_2]$, let

$$d([\phi_1, \phi_2]) = d(\phi_1) + d(\phi_2). \tag{15.108}$$

Thus, the order is just the nesting depth (plus one) of the Lie bracket operations. For example, $d([h_1, h_2]) = 2$ and $d([h_1, [h_2, h_3]]) = 3$.

In addition to standard breadth-first search, pruning should be automatically performed to ensure that the skew symmetry and Jacobi identities are always utilized to eliminate redundancy. A *P. Hall basis* is a sequence, $\mathcal{PH} = (\phi_1, \phi_2, \ldots)$, of Lie products for which:

1. The system vector fields h_i are the first m elements of $\mathcal{PH}$.
2. If $d(\phi_i) < d(\phi_j)$, then $i < j$.
3. Each $[\phi_i, \phi_j] \in \mathcal{PH}$ if and only if: a) $\phi_i, \phi_j \in \mathcal{PH}$ and $i < j$, and b) either $\phi_j = h_i$ for some i or $\phi_j = [\phi_l, \phi_r]$ for some $\phi_l, \phi_r \in \mathcal{PH}$ such that $l \leq i$.

It is shown in many algebra books (e.g., [861]) that this procedure results in a basis for the Lie algebra $\mathcal{L}(\triangle)$. Various algorithms for computing the basis are evaluated in [299].

Example 15.18 (P. Hall Basis Up to Depth Three) The P. Hall basis sorts the Lie products into the following sequence, which is obtained up to depth $d = 3$:

$$\begin{array}{llll} h_1, & h_2, & h_3, & \\ [h_1, h_2], & [h_2, h_3], & [h_1, h_3], & \\ [h_1, [h_1, h_2]], & [h_1, [h_1, h_3]], & [h_2, [h_1, h_2]], & [h_2, [h_1, h_3]], \\ [h_2, [h_2, h_3]], & [h_3, [h_1, h_2]], & [h_3, [h_1, h_3]], & [h_3, [h_2, h_3]] \ . \end{array}$$

So far, the only Lie product eliminated by the Jacobi identity is $[h_1, [h_2, h_3]]$ because

$$[h_1, [h_2, h_3]] = [h_2, [h_1, h_3]] - [h_3, [h_1, h_2]]. \tag{15.109}$$

Note that all of the Lie products given here may not be linearly independent vector fields. For a particular system, linear independence tests should be performed to delete any linearly dependent vector fields from the basis. ■

When does the sequence $\mathcal{PH}$ terminate? Recall that $\dim(\mathcal{L}(\triangle))$ can be no greater than n, because $\mathcal{L}_x(\triangle) \subseteq T_x(X)$. In other words, at every state $x \in X$, the number of possible independent velocity vectors is no more than the dimension of the tangent space at x. Therefore, $\mathcal{PH}$ can be terminated once n independent vector fields are obtained because there is no possibility of finding more. For some systems, there may be a depth k after which all Lie brackets are zero. Such systems are called *nilpotent* of order k. This occurs, for example, if all components of all vector fields are polynomials. If the system is not nilpotent, then achieving termination may be difficult. It may be the case that $\dim(\mathcal{L}(\triangle))$ is strictly less than n, but this is usually not known in advance. It is difficult to determine whether more Lie brackets are needed to increase the dimension or the limit has already been reached.

15.4.3.4 Controllability of driftless systems

The controllability of a driftless control-affine system (15.53) can be characterized using the *Lie algebra rank condition* (or *LARC*). Recall the definition of STLC from Section 15.1.3. Assume that either $U = \mathbb{R}^m$ or U at least contains an open set that contains the origin of $\mathbb{R}^m$. The *Chow-Rashevskii theorem* [112, 156, 846] states:

A driftless control-affine system, (15.53), is small-time locally controllable (STLC) at a point $x \in X$ if and only if $\dim(\mathcal{L}_x(\triangle)) = n$, the dimension of X.

If the condition holds for every $x \in X$, then the whole system is STLC. Integrability can also be expressed in terms of $\dim(\mathcal{L}(\triangle))$. Assume as usual that $m < n$. The three cases are:

1. $\dim(\mathcal{L}(\triangle)) = m$ — the system is completely integrable;
2. $m < \dim(\mathcal{L}(\triangle)) < n$ — the system is nonholonomic, but not STLC;
3. $\dim(\mathcal{L}(\triangle)) = n$ — the system is nonholonomic and STLC.

(15.110)

Example 15.19 (Controllability Examples) The differential drive, nonholonomic integrator, and the system from Example 15.17 are all STLC by the Chow-Rashevskii theorem because $\dim(\mathcal{L}(\triangle)) = n$. This implies that the state can be changed in any direction, even though there are differential constraints. The state can be made to follow arbitrarily close to any smooth curve in X. A method that achieves this based on the Lie algebra is given in Section 15.5.1. The fact that these systems are STLC assures the existence of an LPM that satisfies the

topological property of Section 14.6.2. ■

15.4.3.5 Handling Control-Affine Systems with Drift

Determining whether a system with drift (15.52), is STLC is substantially more difficult. Imagine a mechanical system, such as a hovercraft, that is moving at a high speed. Due to momentum, it is impossible from most states to move in certain directions during an arbitrarily small interval of time. One can, however, ask whether a system is STLC from a state $x \in X$ for which $h_0(x) = 0$. For a mechanical system, this usually means that it starts at rest. If a system with drift is STLC, this intuitively means that it can move in any direction by hovering around states that are close to zero velocity for the mechanical system.

The Lie algebra techniques can be extended to determine controllability for systems with drift; however, the tools needed are far more complicated. See Chapter 7 of [156] for more complete coverage. Even if $\dim(\mathcal{L}(\triangle)) = n$, it does not necessarily imply that the system is STLC. It does at least imply that the system is accessible, which motivates the definition given in Section 15.1.3. Thus, the set of achievable velocities still has dimension n; however, motions in all directions may not be possible due to drift. To obtain STLC, a sufficient condition is that the set of possible values for $\dot{x}$ contains an open set that contains the origin.

The following example clearly illustrates the main difficultly with establishing whether a system with drift is STLC.

Example 15.20 (Accessible, Not STLC) The following simple system clearly illustrates the difficulty caused by drift and was considered in [741]. Let $X = \mathbb{R}^2$, $U = \mathbb{R}$, and the state transition equation be

$$\begin{aligned} \dot{x}_1 &= u \\ \dot{x}_2 &= x_1^2. \end{aligned} \tag{15.111}$$

This system is clearly not controllable in any sense because x_2 cannot be decreased. The vector fields are $h_0(x) = [0 \;\; x_1^2]^T$ and $h_1(x) = [1 \;\; 0]^T$. The first independent Lie bracket is

$$[h_1, [h_0, h_1]] = [0 \;\; -2]. \tag{15.112}$$

The two-dimensional Lie algebra is

$$\mathcal{L}(\triangle) = \text{span}\{h_1, [h_1, [h_0, h_1]]\}, \tag{15.113}$$

which implies that the system is accessible. It is not STLC, however, because the bracket $[h_1, [h_0, h_1]]$ was constructed using h_0 and was combined in an unfortunate way. This bracket is indicating that changing x_2 is possible; however, we already know that it is not possible to decrease x_2. Thus, some of the vector fields obtained from Lie brackets that involve h_0 may have directional constraints. ■

In Example 15.20, $[h_1, [h_0, h_1]]$ was an example of a *bad bracket* [925] because it obstructed controllability. A method of classifying brackets as *good* or *bad* has been developed, and there exist theorems that imply whether a system with drift is STLC by satisfying certain conditions on the good and bad brackets. Intuitively, there must be enough good brackets to neutralize the obstructions imposed by the bad brackets [156, 925].

15.5 Steering Methods for Nonholonomic Systems

This section briefly surveys some methods that solve the BVP for nonholonomic systems. This can be considered as a motion planning problem under differential constraints but in the absence of obstacles. For linear systems, optimal control techniques can be used, as covered in Section 15.2.2. For mechanical systems that are fully actuated, standard control techniques such as the acceleration-based control model in (8.47) can be applied. If a mechanical system is underactuated, then it is likely to be nonholonomic. As observed in Section 15.4, it is possible to generate motions that appear at first to be prohibited. Suppose that by the Chow-Rashevskii theorem, it is shown that a driftless system is STLC. This indicates that it should be possible to design an LPM that successfully connects any pair of initial and goal states. The next challenge is to find an action trajectory $\tilde{u}$ that actually causes x_I to reach x_G upon integration in (14.1). Many methods in Chapter 14 could actually be used, but it is assumed that these would be too slow. The methods in this section exploit the structure of the system (e.g, its Lie algebra) and the fact that there are no obstacles to more efficiently solve the planning problem.

15.5.1 Using the P. Hall Basis

The steering method presented in this section is due to Lafferriere and Sussmann [574]. It is assumed here that a driftless control-affine system is given, in which X is a Lie group, as introduced in Example 15.15. Furthermore, the system is assumed to be STLC. The steering method sketched in this section follows from the Lie algebra $\mathcal{L}(\triangle)$. The idea is to apply piecewise-constant motion primitives to move in directions given by the P. Hall basis. If the system is nilpotent, then this method reaches the goal state exactly. Otherwise, it leads to an approximate method that can be iterated to get arbitrarily close to the goal. Furthermore, some systems are *nilpotentizable* by using feedback [442].

The main idea is to start with (15.53) and construct an *extended system*

$$\dot{x} = \sum_{i=1}^{s} b_i(x) v_i, \tag{15.114}$$

in which each v_i is an action variable, and b_i is a vector field in $\mathcal{PH}$, the P. Hall basis. For every $i \leq m$, each term of (15.114) is $b_i(x)v_i = h_i(x)u_i$, which comes from the original system. For $i > m$, each b_i represents a Lie product in $\mathcal{PH}$, and v_i is a *fictitious action variable*. It is called fictitious because the velocity given by b_i for $i > m$ cannot necessarily be achieved by using a single action variable of the system. In general, s may be larger than n because at each $x \in X$ a different subset of $\mathcal{PH}$ may be needed to obtain n independent vectors. Also, including more basis elements simplifies some of the coming computations.

Example 15.21 (Extended System for the Nonholonomic Integrator) The extended system for the nonholonomic integrator (15.83) is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -x_2 \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 1 \\ x_1 \end{pmatrix} v_2 + \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} v_3. \tag{15.115}$$

The first two terms correspond to the original system. The last term arises from the Lie bracket $[h_1, h_2]$. Only one fictitious action variable is needed because the three P. Hall vector fields are independent at every $x \in X$.

It is straightforward to move this system along a grid-based path in $\mathbb{R}^3$. Motions in the x_1 and x_2 directions are obtained by applying $v_1 = u_1$ and $v_2 = u_2$, respectively. To move the system in the x_3 direction, the commutator motion in (15.71) should be performed. This corresponds to applying v_3. The steering method described in this section yields a generalization of this approach. Higher degree Lie products can be used, and motion in any direction can be achieved. ■

Suppose some x_I and x_G are given. There are two phases to the steering method:

1. Determine an action trajectory $\tilde{v}$ for the extended system, for which $x(0) = x_I$ and $x(t_F) = x_G$ for some $t_F > 0$.

2. Convert $\tilde{v}$ into an action trajectory $\tilde{u}$ that eliminates the fictitious variables and uses the actual m action variables $u_1, \ldots, u_m$.

The first phase is straightforward. For the extended system, any velocity in the tangent space, $T_x(X)$, can be generated. Start with any smooth path $\tau : [0, 1] \rightarrow X$ such that $\tau(0) = x_I$ and $\tau(1) = x_G$. The velocity $\dot{\tau}(t)$ along the path τ is a velocity vector in $T_{\tau(t)}(X)$ that can be expressed as a linear combination of the $b_i(\tau(t))$ vectors using linear algebra. The coefficients of this combination are the v_i values. The second phase is much more complicated and will be described shortly. If the system is nilpotent, then $\tilde{u}$ should bring the system precisely from x_I to x_G. By the way it is constructed, it will also be clear how to refine $\tilde{u}$ to come as close as desired to the trajectory produced by $\tilde{v}$.

Formal calculations The second phase is solved using formal algebraic computations. This means that the particular vector fields, differentiation, manifolds, and so on, can be ignored. The concepts involve pure algebraic manipulation. To avoid confusion with previous definitions, the term *formal* will be added to many coming definitions. Recall from Section 4.4.1 the formal definitions of the algebra of polynomials (e.g., $\mathbb{F}[x_1, \ldots, x_n]$). Let $A(y_1, \ldots, y_m)$ denote the *formal noncommutative algebra*[11] of polynomials in the variables y_1, ..., y_m. The y_i here are treated as symbols and have no other assumed properties (e.g, they are not necessarily vector fields). When polynomials are multiplied in this algebra, no simplifications can be made based on commutativity. The algebra can be converted into a Lie algebra by defining a Lie bracket. For any two polynomials $p, q \in A(y_1, \ldots, y_m)$, define the *formal Lie bracket* to be $[p, q] = pq - qp$. The formal Lie bracket yields an equivalence relation on the algebra; this results in a *formal Lie algebra* $L(y_1, \ldots, y_m)$ (there are many equivalent expressions for the same elements of the algebra when the formal Lie bracket is applied). Nilpotent versions of the formal algebra and formal Lie algebra can be made by forcing all monomials of degree $k + 1$ to be zero. Let these be denoted by $A_k(y_1, \ldots, y_m)$ and $L_k(y_1, \ldots, y_m)$, respectively. The P. Hall basis can be applied to obtain a basis of the formal Lie algebra. Example 15.18 actually corresponds to the basis of $L_3(h_1, h_2, h_3)$ using formal calculations.

The exponential map The steering problem will be solved by performing calculations on $L_k(y_1, \ldots, y_m)$. The *formal power series* of $A(y_1, \ldots, y_m)$ is the set of all linear combinations of monomials, including those that have an infinite number of terms. Similarly, the *formal Lie series* of $L(y_1, \ldots, y_m)$ can be defined.

The *formal exponential map* is defined for any $p \in A(y_1, \ldots, y_m)$ as

$$e^p = 1 + p + \frac{1}{2!}p^2 + \frac{1}{3!}p^3 + \cdots . \tag{15.116}$$

In the nilpotent case, the *formal exponential map* is defined for any $p \in A_k(y_1, \ldots, y_m)$ as

$$e^p = \sum_{i=0}^{k} \frac{p^i}{i!}. \tag{15.117}$$

The formal series is truncated because all terms with exponents larger than k vanish.

A *formal Lie group* is constructed as

$$G_k(y_1, \ldots, y_m) = \{e^p \mid p \in L_k(y_1, \ldots, y_m)\}. \tag{15.118}$$

If the formal Lie algebra is not nilpotent, then a formal Lie group $G(y_1, \ldots, y_m)$ can be defined as the set of all e^p, in which p is represented using a formal Lie series.

The following example is taken from [574]:

[11] Intuitively, being an algebra means that polynomials can be added and multiplied; for all of the required axioms, see [469].

Example 15.22 (Formal Lie Groups) Suppose that the generators x and y are given. Some elements of the formal Lie group $G(x, y)$ are

$$e^x = I + x + \tfrac{1}{2}x^2 + \tfrac{1}{6}x^3 + \cdots , \tag{15.119}$$

$$e^{[x,y]} = I + [x, y] + \tfrac{1}{2}[x, y]^2 + \cdots , \tag{15.120}$$

and

$$e^{x-y+3[x,y]} = I + x - y + 3[x, y] + \cdots , \tag{15.121}$$

in which I is the formal Lie group identity. Some elements of the formal Lie group $G_2(x, y)$ are

$$e^x = I + x + \tfrac{1}{2}x^2, \tag{15.122}$$

$$e^{[x,y]} = I + [x, y], \tag{15.123}$$

and

$$e^{x-y+3[x,y]} = I + x - y + 3[x, y] + \tfrac{1}{2}(x - y)^2. \tag{15.124}$$

■

To be a group, the axioms given in Section 4.2.1 must be satisfied. The identity is I, and associativity clearly follows from the series representations. Each e^p has an inverse, e^{-p}, because $e^p e^{-p} = I$. The only remaining axiom to satisfy is closure. This is given by the *Campbell-Baker-Hausdorff-Dynkin formula* (or *CBHD formula*), for which the first terms for any $p, q \in G(y_1, \ldots, y_m)$ are

$$\exp(p)\exp(q) = \exp(p+q+\tfrac{1}{2}[p, q]+\tfrac{1}{12}[[p, q], q]-\tfrac{1}{12}[[p, q], p]+\tfrac{1}{24}[p, [q, [p, q]]]+\cdots), \tag{15.125}$$

in which $\exp(x)$ alternatively denotes e^x for any x. The formula also applies to $G_k(y_1, \ldots, y_m)$, but it becomes truncated into a finite series. This fact will be utilized later. Note that $e^p e^q \neq e^{p+q}$, which differs from the standard definition of exponentiation.

The CBHD formula is often expressed as

$$e^p e^q e^{-p} = \exp\left(\sum_{i=0}^{\infty} \frac{\mathrm{Ad}_p^i\, q}{i!}\right), \tag{15.126}$$

in which $\mathrm{Ad}_p^0\, q = q$, and $\mathrm{Ad}_p^i\, q = [p, \mathrm{Ad}_p^{i-1}\, q]$. The operator Ad provides a compact way to express some nested Lie bracket operations. Additional terms of (15.125) can be obtained using (15.126).

The Chen-Fliess series The P. Hall basis from Section 15.4.3 applies in general to any Lie algebra. Let B_1, ..., B_s denote a P. Hall basis for the nilpotent formal Lie algebra $L_k(y_1, \ldots, y_m)$. An important theorem in the study of formal Lie

groups is that every $S \in G_k(y_1, \ldots, y_m)$ can be expressed in terms of the P. Hall basis of its formal Lie algebra as

$$S = e^{z_s B_s} e^{z_{s-1} B_{s-1}} \cdots e^{z_2 B_2} e^{z_1 B_1}, \tag{15.127}$$

which is called the *Chen-Fliess series*. The z_i are sometimes called the backward P. Hall coordinates of S (there is a forward version, for which the terms in (15.127) go from 1 to s, instead of s to 1).

Returning to the system vector fields Now the formal algebra concepts can be applied to the steering problem. The variables become the system vector fields: $y_i = h_i$ for all i from 1 to m. For the P. Hall basis elements, each B_i becomes b_i. The Lie group becomes the state space X, and the Lie algebra is the familiar Lie algebra over the vector fields, which was introduced in Section 15.4.3. Consider how an element of the Lie group must evolve over time. This can be expressed using the differential equation

$$\dot{S}(t) = S(t)(v_1 b_1 + v_2 b_2 + \cdots + v_s b_s), \tag{15.128}$$

which is initialized with $S(0) = I$. Here, S can be interpreted as a matrix, which may, for example, belong to $SE(3)$.

The solution at every time $t > 0$ can be written using the Chen-Fliess series, (15.127):

$$S(t) = e^{z_s(t) b_s} e^{z_{s-1}(t) b_{s-1}} \cdots e^{z_2(t) b_2} e^{z_1(t) b_1}. \tag{15.129}$$

This indicates that $S(t)$ can be obtained by integrating b_1 for time $z_1(t)$, followed by b_2 for time $z_2(t)$, and so on until b_s is integrated for time $z_s(t)$. Note that the backward P. Hall coordinates now vary over time. If we determine how they evolve over time, then the differential equation in (15.128) is solved.

The next step is to figure out how the backward P. Hall coordinates evolve. Differentiating (15.129) with respect to time yields

$$\dot{S}(t) = \sum_{j=1}^{s} e^{z_s b_s} \cdots e^{z_{j+1} b_{j+1}} \dot{z}_j b_j e^{z_j b_j} \cdots e^{z_1 b_1}. \tag{15.130}$$

The Chen-Fliess-Sussmann equation There are now two expressions for $\dot{S}$, which are given by (15.128) and (15.130). By equating them, s equations of the form

$$\sum_{j=1}^{s} p_{j,k} \dot{z}_j = v_k \tag{15.131}$$

are obtained, in which $p_{j,k}$ is a polynomial in z_i variables. This makes use of the series representation for each exponential; see Example 15.23.

The evolution of the backward P. Hall coordinates is therefore given by the *Chen-Fliess-Sussmann (CFS) equation*:

$$\dot{z} = Q(z) v, \tag{15.132}$$

in which $Q(z)$ is an $s \times s$ matrix, and $z(0) = 0$. The entries in $Q(z)$ are polynomials; hence, it is possible to integrate the system analytically to obtain expressions for the $z_i(t)$.

A simple example is given, which was worked out in [299]:

Example 15.23 (The CFS Equation for the Nonholonomic Integrator) The extended system for the nonholonomic integrator was given in (15.115). The differential equation (15.128) for the Lie group is

$$\dot{S}(t) = S(t)(v_1 b_1 + v_2 b_2 + v_3 b_3), \tag{15.133}$$

because $s = 3$.

There are two expressions for its solution. The Chen-Fliess series (15.129) becomes

$$S(t) = e^{z_3(t)b_3} e^{z_2(t)b_2} e^{z_1(t)b_1}. \tag{15.134}$$

The initial condition $S(0) = I$ is satisfied if $z_i(0) = 0$ for i from 1 to 3. The second expression for $\dot{S}(t)$ is (15.130), which in the case of the nonholonomic integrator becomes

$$\begin{aligned} \dot{S}(t) = & \dot{z}_3(t) b_3 e^{z_3(t)b_3} e^{z_2(t)b_2} e^{z_1(t)b_1} + \\ & e^{z_3(t)b_3} \dot{z}_2(t) b_2 e^{z_2(t)b_2} e^{z_1(t)b_1} + \\ & e^{z_3(t)b_3} e^{z_2(t)b_2} \dot{z}_1(t) b_1 e^{z_1(t)b_1}. \end{aligned} \tag{15.135}$$

Note that

$$S^{-1}(t) = e^{-z_1(t)b_1} e^{-z_2(t)b_2} e^{-z_3(t)b_3}. \tag{15.136}$$

Equating (15.133) and (15.135) yields

$$\begin{aligned} S^{-1}\dot{S} = v_1 b_1 + v_2 b_2 + v_3 b_3 = & e^{-z_1 b_1} e^{-z_2 b_2} e^{-z_3 b_3} \dot{z}_3 b_3 e^{z_3 b_3} e^{z_2 b_2} e^{z_1 b_1} + \\ & e^{-z_1 b_1} e^{-z_2 b_2} \dot{z}_2 b_2 e^{z_2 b_2} e^{z_1 b_1} + \\ & e^{-z_1 b_1} \dot{z}_1 b_1 e^{z_1 b_1}, \end{aligned} \tag{15.137}$$

in which the time dependencies have been suppressed to shorten the expression. The formal Lie series expansions, appropriately for the exponentials, are now used. For $i = 1, 2$,

$$e^{z_i b_i} = (I + z_i b_i + \tfrac{1}{2} z_i^2 b_i^2) \tag{15.138}$$

and

$$e^{-z_i b_i} = (I - z_i b_i - \tfrac{1}{2} z_i^2 b_i^2). \tag{15.139}$$

Also,

$$e^{z_3 b_3} = (I + z_3 b_3) \tag{15.140}$$

and

$$e^{-z_3 b_3} = (I - z_3 b_3). \tag{15.141}$$

The truncation is clearly visible in (15.140) and (15.141). The b_3^2 terms are absent because b_3 is a polynomial of degree two, and its square would be of degree four.

Substitution into (15.137), performing noncommutative multiplication, and applying the Lie bracket definition yields

$$\dot{z}_1 b_1 + \dot{z}_2(b_2 - z_1 b_3) + \dot{z}_3 b_3 = v_1 b_1 + v_2 b_2 + v_3 b_3. \tag{15.142}$$

Equating like terms yields the Chen-Fliess-Sussmann equation

$$\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= v_3 + z_1 v_2. \end{aligned} \tag{15.143}$$

Recall that $\tilde{v}$ is given. By integrating (15.143) from $z(0) = 0$, the backward P. Hall coordinate trajectory $\tilde{z}$ is obtained. ■

Using the original action variables Once the CFS equation has been determined, the problem is almost solved. The action trajectory $\tilde{v}$ was determined from the given state trajectory $\tilde{v}$ and the backward P. Hall coordinate trajectory $\tilde{z}$ is determined by (15.143). The only remaining problem is that the action variables from v_{m+1} to v_s are fictitious because their associated vector fields are not part of the system. They were instead obtained from Lie bracket operations. When these are applied, they interfere with each other because many of them may try to use the same u_i variables from the original system at the same time.

The CBHD formula is used to determine the solution in terms of the system action variables $u_1, \ldots, u_m$. The differential equation now becomes

$$\dot{S}(t) = S(t)(u_1 h_1 + u_2 h_2 + \cdots + u_m h_m), \tag{15.144}$$

which is initialized with $S(0) = I$ and uses the original system instead of the extended system.

When applying vector fields over time, the CBHD formula becomes

$$\begin{aligned} &\exp(tf)\exp(tg) = \\ &\exp(tf + tg + \frac{t^2}{2}[f,g] + \frac{t^3}{12}[[f,g],g] - \frac{t^3}{12}[[f,g],f] + \frac{t^4}{24}[f,[g,[f,g]]] + \cdots). \end{aligned} \tag{15.145}$$

If the system is nilpotent, then this series is truncated, and the exact effect of sequentially combining constant motion primitives can be determined. This leads to a procedure for determining a finite sequence of constant motion primitives that generate a motion in the same direction as prescribed by the extended system and the action trajectory $\tilde{v}$.

15.5.2 Using Sinusoidal Action Trajectories

The steering method presented in this section is based on initial work by Brockett [142] and a substantial generalization of it by Murray and Sastry [727]. The approach applies to several classes of systems for which the growth of independent vector fields occurs as quickly as possible. This means that when the P. Hall basis is constructed, no elements need to be removed due to linear dependency on previous Lie products or system vector fields. For these systems, the approach applies sinusoids of integrally related frequencies to some action variables. This changes some state variables while others are automatically fixed. For more details beyond the presentation here, see [596, 725, 727, 846].

15.5.2.1 Steering the nonholonomic integrator

The main idea of the method can be clearly illustrated for the nonholonomic integrator,

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1, \end{aligned} \tag{15.146}$$

which was considered throughout Section 15.5.1. This case will be explained in detail, and the methods obtained by generalizing the principles will subsequently be stated. The presentation given here is based on [727, 846].

As was previously indicated, growing independent vector fields as quickly as possible is important. For the nonholonomic integrator, $[h_1, h_2]$, is linearly independent of h_1 and h_2, as observed in Example 15.12; thus, it satisfies this property. Consider steering the system from some $x_I = x(0)$ to some $x_G = x(1)$ while optimizing the cost functional

$$\int_0^1 \big(u_1(t)^2 + u_2(t)^2\big) dt. \tag{15.147}$$

The problem can be solved by using the constrained Lagrangian formulation, which was given in Section 13.4.3. The first step is to eliminate the u variables. From (15.146), the cost can be expressed in terms of $\dot{x}_1$ and $\dot{x}_2$ by using $\dot{x}_1 = u_1$ and $\dot{x}_2 = u_2$. The third equation in (15.146) can be written as

$$\dot{x}_3 = x_1 \dot{x}_2 - x_2 \dot{x}_1 \tag{15.148}$$

and will be interpreted as a constraint on the Lagrangian, which is combined using a (scalar) Lagrange multiplier as explained in Section 13.4.3. Define the Lagrangian as

$$L(x, \dot{x}) = (\dot{x}_1^2 + \dot{x}_2^2) + \lambda\big(\dot{x}_3 - x_1 \dot{x}_2 + x_2 \dot{x}_1\big), \tag{15.149}$$

in which the first term comes from the integrand of (15.147), and the second term comes from (15.148).

The Euler-Lagrange equation (13.118) yields

$$\begin{aligned} \ddot{x}_1 + \lambda \dot{x}_2 &= 0 \\ \ddot{x}_2 - \lambda \dot{x}_1 &= 0 \\ \dot{\lambda} &= 0. \end{aligned} \tag{15.150}$$

Note that $\dot{\lambda} = 0$ implies that $\lambda(t)$ is constant for all time. To obtain a differential equation that characterizes the optimal action trajectory, use the fact that for $i = 1, 2$, $\dot{x}_i = u_i$ and $\ddot{x}_i = \dot{u}_i$. This yields the equations $\dot{u}_1 = -\lambda \dot{u}_2$ and $\dot{u}_2 = \lambda \dot{u}_1$. These can be represented as second-order linear differential equations. Based on its roots, the solution is

$$\begin{aligned} u_1(t) &= u_1(0) \cos \lambda t - u_2(0) \sin \lambda t \\ u_2(t) &= u_1(0) \sin \lambda t + u_2(0) \cos \lambda t. \end{aligned} \tag{15.151}$$

Given initial and goal states, the optimal action trajectory is found by determining $u_1(0)$, $u_2(0)$, and λ. Suppose that $x_I = x(0) = (0, 0, 0)$ and $x_G = x(1) = (0, 0, a)$ for some $a \in \mathbb{R}$. Other cases can be obtained by applying transformations in $SE(3)$ to the solution.

The state trajectories for x_1 and x_2 can be obtained by integration of (15.151) because $u_i = \dot{x}_i$ for $i = 1$ and $i = 2$. Starting from $x_1(0) = x_2(0) = 0$, this yields

$$\begin{aligned} x_1(t) &= \frac{1}{\lambda}\big(u_1(0) \sin \lambda t + u_2(0) \cos \lambda t - u_2(0)\big) \\ x_2(t) &= \frac{1}{\lambda}\big(- u_1(0) \cos \lambda t + u_2(0) \sin \lambda t + u_1(0)\big). \end{aligned} \tag{15.152}$$

To maintain the constraint that $x_1(1) = x_2(1) = 0$, λ must be chosen as $\lambda = 2k\pi$ for some integer n. Integration of $\dot{x}_3$ yields

$$x_3(t) = \int_0^1 \big(x_1 u_2 - x_2 u_1\big) dt = \frac{1}{\lambda}\big(u_1^2(0) + u_2^2(0)\big) = a. \tag{15.153}$$

The cost is

$$\int_0^1 \big(u_1^2(t) + u_2^2(t)\big) dt = u_1^2(0) + u_2^2(0) = \lambda a. \tag{15.154}$$

The minimum cost is therefore achieved for $k = -1$, which yields $\lambda = 2\pi$ and $\|u\| = 2\pi a$. This fixes the magnitude of $u(0)$, but any direction may be chosen.

The steering problem can be solved in two phases:

1. Apply any action trajectory to steer x_1 and x_2 to their desired values while neglecting to consider x_3.

2. Apply the solution just developed to steer x_3 to the goal while x_1 and x_2 return to their values obtained in the first phase.

This idea can be generalized to other systems.

15.5.2.2 First-order controllable systems

The approach developed for the nonholonomic integrator generalizes to systems of the form

$$\begin{aligned} \dot{x}_i &= u_i && \text{for } i \text{ from 1 to } m \\ \dot{x}_{ij} &= x_i u_j - x_j u_i && \text{for all } i, j \text{ so that } i < j \text{ and } 1 \leq j \leq m \end{aligned} \tag{15.155}$$

and

$$\begin{aligned} \dot{x}_i &= u_i && \text{for } i \text{ from 1 to } m \\ \dot{x}_{ij} &= x_i u_j && \text{for all } i, j \text{ such that } i < j \text{ and } 1 \leq j \leq m. \end{aligned} \tag{15.156}$$

Brockett showed in [142] that for such *first-order controllable systems*, the optimal action trajectory is obtained by applying a sum of sinusoids with integrally related frequencies for each of the m action variables. If m is even, then the trajectory for each variable is a sum of $m/2$ sinusoids at frequencies 2π, $2 \cdot 2\pi$, ..., $(m/2) \cdot 2\pi$. If m is odd, there are instead $(m-1)/2$ sinusoids; the sequence of frequencies remains the same. Suppose m is even (the odd case is similar). Each action is selected as

$$u_i = \sum_{k=1}^{m/2} a_{ik} \sin 2\pi k t + b_{ik} \cos 2\pi k t. \tag{15.157}$$

The other state variables evolve as

$$x_{ij} = x_{ij}(0) + \frac{1}{2} \sum_{k=1}^{m/2} \frac{1}{k} (a_{jk} b_{ik} - a_{ik} b_{jk}), \tag{15.158}$$

which provides a constraint similar to (15.153). The periodic behavior of these action trajectories causes the x_i variables to return to their original values while steering the x_{ij} to their desired values. In a sense this is a vector-based generalization in which the scalar case was the nonholonomic integrator.

Once again, a two-phase steering approach is obtained:

1. Apply any action trajectory that brings every x_i to its goal value. The evolution of the x_{ij} states is ignored in this stage.

2. Apply sinusoids of integrally related frequencies to the action variables. Choose each $u_i(0)$ so that x_{ij} reaches its goal value. In this stage, the x_i variables are ignored because they will return to their values obtained in the first stage.

This method has been generalized even further to *second-order controllable systems*:

$$\begin{aligned} \dot{x}_i &= u_i && \text{for } i \text{ from 1 to } m \\ \dot{x}_{ij} &= x_i u_j && \text{for all } i, j \text{ such that } i < j \text{ and } 1 \leq j \leq m \\ \dot{x}_{ijk} &= x_{ij} u_k && \text{for all } (i, j, k) \in J, \end{aligned} \tag{15.159}$$

in which J is the set of all unique triples formed from distinct $i, j, k \in \{1, \ldots, m\}$ and removing unnecessary permutations due to the Jacobi identity for Lie brackets. For this problem, a three-phase steering method can be developed by using ideas similar to the first-order controllable case. The first phase determines x_i, the second handles x_{ij}, and the third resolves x_{ijk}. See [727, 846] for more details.

15.5.2.3 Chained-form systems

Example 15.17 considered a special case of a *chained-form system*. The system in (15.102) can be generalized to any n as

$$\begin{aligned} \dot{x}_1 &= u_1 & \qquad \dot{x}_4 &= x_3 u_1 \\ \dot{x}_2 &= u_2 & \qquad &\vdots \\ \dot{x}_3 &= x_2 u_1 & \qquad \dot{x}_n &= x_{n-1} u_1. \end{aligned} \tag{15.160}$$

This can be considered as a system with *higher order controllability*. For these systems, a multi-phase approach is obtained:

1. Apply any action trajectory for u_1 and u_2 that brings x_1 and x_2 to their goal values. The evolution of the other states is ignored in this stage.

2. This phase is repeated for each k from 3 to n. Steer x_k to its desired value by applying

$$u_1 = a \sin 2\pi k t \quad \text{and} \quad u_2 = b \cos 2\pi k t, \tag{15.161}$$

in which a and b are chosen to satisfy the constraint

$$x_k(1) = x_k(0) + \left(\frac{a}{4\pi}\right)^{(k-2)} \frac{b}{(k-2)!}. \tag{15.162}$$

Each execution of this phase causes the previous $k-1$ state variables to return to their previous values.

For a proof of the correctness of the second phase, and more information in general, see [727, 846]. It may appear that very few systems fit the forms given in this section; however, it is sometimes possible to transform systems to fit this form. Recall that the original simple car model in (13.15) was simplified to (15.54). Transformation methods for putting systems into chained form have been developed. For systems that still cannot be put in this form, Fourier techniques can be used to obtain approximate steering methods that are similar in spirit to the methods in this section. When the chained-form system is expressed using Pfaffian constraints, the result is often referred to as the *Goursat normal form*. The method can be extended even further to *multi-chained-form systems*.

15.5.3 Other Steering Methods

The steering methods presented so far are perhaps the most widely known; however, several other alternatives exist. Most of these follow in the spirit of the methods in Sections 15.5.1 and 15.5.2 by exploiting the properties of a specific class of systems. Some alternatives are briefly surveyed here. This is an active field of research; it is likely that new methods will be developed in the coming years.

Differentially flat systems Differential flatness has become an important concept in the development of steering methods. It was introduced by Fliess, Lévine, Martin, and Rouchon in [344]; see also [726]. Intuitively, a system is said to be *differentially flat* if a set of variables called *flat outputs* can be found for which all states and actions can be determined from them without integration. This specifically means that for a system $\dot{x} = f(x, u)$ with $X = \mathbb{R}^n$ and $U = \mathbb{R}^m$, there exist *flat outputs* of the form

$$y = h(x, u, \dot{u}, \ldots, u^{(k)}) \tag{15.163}$$

such that there exist functions g and g' for which

$$x = g(y, \dot{y}, \ldots, y^{(j)}) \tag{15.164}$$

and

$$u = g'(y, \dot{y}, \ldots, y^{(j)}). \tag{15.165}$$

One example is the simple car pulling trailers, expressed in (13.19); the flat outputs are the position in $\mathcal{W} = \mathbb{R}^2$ of the last trailer. This property was used for motion planning in [578]. Recent works on the steering of differentially flat systems include [578, 813, 833].

Decoupling vector fields For mechanical systems in which dynamics is considered, the steering problem becomes complicated by drift. One recent approach is based on establishing that a system is *kinematically controllable*, which means that the system is STLC on the C-space, if traversed using trajectories that start and stop at zero velocity states [157]. The method finds *decoupling vector fields* on the C-space. Any path that is the integral curve of a decoupling vector field in the C-space is executable by the full system with dynamics. If a mechanical system admits such vector fields, then it was proved in [157] that a steering method for $\mathcal{C}$ can be lifted into one for X, the phase space of the mechanical system. This idea was applied to generate an efficient LPM in an RRT planner in [224].

Averaging methods By decomposing the state trajectory into a low-frequency part that accounts for the long-range evolution of states and a high-frequency part that accounts for small oscillations over short ranges, *averaging methods* enable perturbations to be systematically made to state trajectories. This yields other steering methods based on sinusoidal action trajectories [112, 420, 623, 624].

Variational techniques As might be expected, the general-purpose gradient-based optimization techniques of Section 14.7 can be applied to the steering of nonholonomic systems. Such methods are based on Newton iterations on the space of possible state trajectories. This leads to a gradient descent that arrives at a local optimum while satisfying the differential constraints. For details on applying such techniques to steer nonholonomic systems, see [276, 334, 596, 901, 926].

Pontryagin's minimum principle The minimum principle can be helpful in developing a steering method. Due to the close connection between the Euler-Lagrange equation and Hamilton's equations, as mentioned in Section 13.4.4, this should not be surprising. The Euler-Lagrange equation was used in Section 15.5.2 to determine an optimal steering method for the nonholonomic integrator. A steering methodology based on the minimum principle is described in [846]. The optimal curves of Section 15.3 actually represent steering methods obtained from the minimum principle. Unfortunately, for the vast majority of problems, numerical techniques are needed to solve the resulting differential equations. It is generally expected that techniques developed for specific classes, such as the nilpotent, chained-form, or differentially flat systems, perform much better than general-purpose numerical techniques applied to the Euler-Lagrange equation, Hamilton's equations or Pontryagin's minimum principle.

Dynamic programming The numerical dynamic programming approach of Section 14.5 can be applied to provide optimal steering for virtual any system. To apply it here, the obstacle region X_{free} is empty. The main drawback, however, is that the computational cost is usually too high, particularly if the dimension of X is high. On the other hand, it applies in a very general setting, and Lie group symmetries can be used to apply precomputed trajectories from any initial state. This is certainly a viable approach with systems for which the state space is $SE(2)$ or $SO(3)$.

Further Reading

The basic stability and controllability concepts from Section 15.1 appear in many control textbooks, especially ones that specialize in nonlinear control; see [523, 846] for an introduction to nonlinear control. More advanced concepts appear in [156]. For illustrations of many convergence properties in vector fields, see [44]. For linear system theory, see [192]. Brockett's condition and its generalization appeared in [143, 996]. For more on stabilization and feedback control of nonholonomic systems, see [156, 846, 964]. For Lyapunov-based design for feedback control, see [278].

For further reading on the Hamilton-Jacobi-Bellman equation, see [85, 95, 492, 789, 912]. For numerical approaches to its solution (aside from value iteration), see [2, 253, 707]. Linear-quadratic problems are covered in [28, 570]. Pontryagin's original works provide an unusually clear explanation of the minimum principle [801]. For other sources, see [95, 410, 789]. A generalization that incorporates state-space constraints appears in [927].

Works on which Section 15.3 is based are [64, 127, 211, 294, 814, 903, 904, 923]. Optimal curves have been partially characterized in other cases; see [227, 903]. One complication is that optimal curves often involve infinite switching [370, 1000]. There is also interest in nonoptimal curves that nevertheless have good properties, especially for use as a local planning method for car-like robots [31, 358, 520, 794, 848]. For feedback control of car-like robots, see [112, 663].

For further reading on nonholonomic system theory beyond Section 15.4, there are many excellent sources: [83, 112, 113, 156, 478, 725, 741, 846]. A generalization of the Chow-Rashevskii theorem to hybrid systems is presented in [724]. Controllability of a car pulling trailers is studied in [594]. Controllability of a planar hovercraft with thrusters is considered in [669]. The term holonomic is formed from two Greek words meaning "integrable" and "law" [135].

Section 15.5 is based mainly on the steering methods in [574] (Section 15.5.1) and [142, 727] (Section 15.5.2). The method of Section 15.5.1 is extended to time-varying systems in [299]. A multi-rate version is developed in [713]. In [480], it was improved by using a Lyndon basis, as opposed to the P. Hall basis. Another steering method that involves series appears in [154, 155]. For more on chained-form systems, see [858, 902]. For a variant that uses polynomials and the Goursat normal form, instead of sinusoids, see [846]. For other steering methods, see the references suggested in Section 15.5.3.

Exercises

1. Characterize the stability at $(0, 0)$ of the vector field on $X = \mathbb{R}^2$, given by $\dot{x}_1 = x_2$ and $\dot{x}_2 = -x_2^2 - x_1$. Use the Lyapunov function $\phi(x_1, x_2) = x_1^2 + x_2^2$.

2. Repeat Example 15.4, but instead use the cost term $l(x, u) = u^2$.

3. Repeat Example 15.4, but instead for a triple integrator $q^{(3)} = u$ and $U = [-1, 1]$.

4. Determine the precise conditions under which each of the four cases of Example 15.4 occurs. Define a feedback motion plan that causes time-optimal motions.

5. Note that some of the six optimal words for the Dubins car do not appear for the Reeds-Shepp car. For each of these, illustrate why it does not appear.

6. Retrace the steps of the Taylor series argument for deriving the Lie bracket in Section 15.4.2. Arrive at (15.81) by showing all steps in detail (smaller steps are skipped in Section 15.4.2).

7. Determine whether the following system is nonholonomic and STLC:

$$\begin{aligned} \dot{q}_1 &= u_1 & \qquad \dot{q}_4 &= q_2^2 u_1 \\ \dot{q}_2 &= u_2 & \qquad \dot{q}_5 &= q_1^2 u_2 \\ \dot{q}_3 &= q_1 u_2 - q_2 u_1. \end{aligned} \tag{15.166}$$

8. Prove that linear systems $\dot{x} = Ax + Bu$ for constant matrices A and B cannot be nonholonomic.

9. Determine whether the following system is nonholonomic and STLC:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ -\sin\psi \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} u_2. \tag{15.167}$$

10. Using the commutator motion and constant actions for the differential drive, develop a lattice over its configuration space.

11. Consider a smooth nonlinear system that has only one action variable and an n-dimensional state space for $n > 1$. Are such systems always completely integrable, always nonholonomic, or is either possible?

12. Generalize Example 15.17 to $\mathbb{R}^n$ with two action variables. Determine whether the system is STLC for any $n > 5$.

13. Show that the vector cross product on $\mathbb{R}^3$ indeed produces a Lie algebra when used as a bracket operation.

14. Derive the CFS equation for the following system:

$$\begin{aligned} \dot{q}_1 &= u_1 & \dot{q}_3 &= q_1 u_2 - q_2 u_1 \\ \dot{q}_2 &= u_2 & \dot{q}_4 &= q_2^2 u_1. \end{aligned} \tag{15.168}$$

Implementations

15. Implement software that computes the P. Hall basis up to any desired order (this is only symbolic computation; the Lie brackets are not expanded).

16. Implement software that displays the appropriate optimal path for the Dubins car, between any given q_I and q_G.

17. Apply the planning algorithm in Section 14.4.2 to numerically determine the Dubins curves. Use Dijkstra's algorithm for the search, and use a high-resolution grid. Can your software obtain the same set of curves as Dubins?

18. Experiment with using Dubins curves as a local planning method (LPM) and metric in an RRT-based planning algorithm. Does using the curves improve execution time? Do they lead to better solutions?