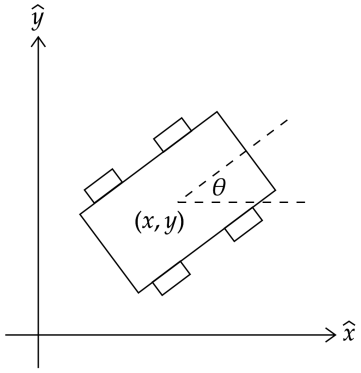
EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2020**

Lecture 8: Differential Geometry and Nonholonomic Control

Scribes: Eric Hu, Jonathan Robinsons

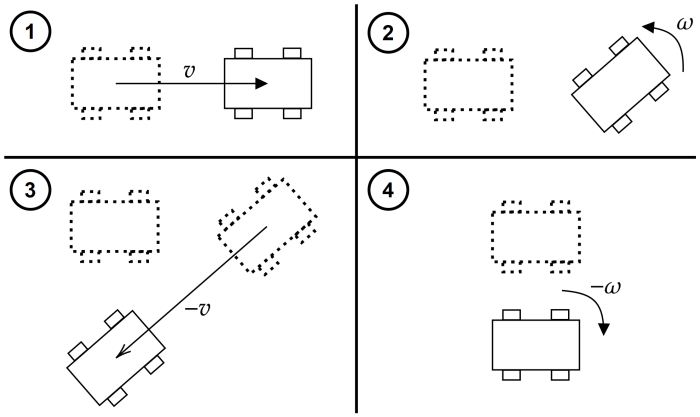
8.1 A motivating example: parallel parking

Consider a very simple Dubin's car model: that is, a car fixed in a 2D plane, which is constrained to roll without slipping, moving only along its current heading.



$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

The inputs v and ω allow us to directly control the forward and angular velocities, respectively. However, what if we need to move our car a small distance to the right, for example when parallel parking? It is clear that this system will not allow us to do so directly, but intuitively, we know there are many sets of inputs that might achieve this: one such solution is to drive forward, turn left, drive backward, and turn right.



Of course, there are many, many systems which are not so intuitive. Instead of relying on intuition, can we develop a mathematical formulation that will allow us to determine the extent to which these constrained

systems can be controlled? This will help form the foundation of **nonholonomic motion planning and control**, or planning in the presence of nonholonomic constraints.

8.2 A somewhat long and tedious derivation

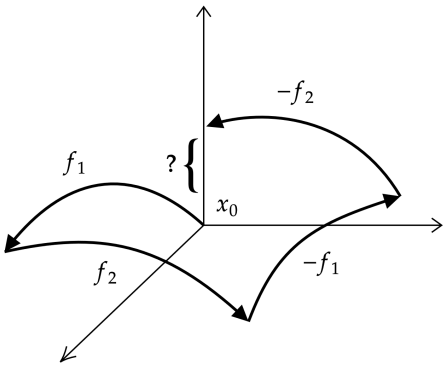
Let us introduce the notation $\phi_t^f(x_0)$ — known as the *flow* of the vector field f — to represent the state of the solution to the differential equation $\dot{x} = f(x)$ at time t (with $x = x_0$ at $t = 0$), such that

$$\frac{d}{dt}\phi_t^f(x) = f\left(\phi_t^f(x)\right) \qquad x \in \mathbb{R}^n \tag{8.1}$$

Now, we define the following control system:

$$\dot{x} = f_1(x) * u_1 + f_2(x) * u_2 \tag{8.2}$$

To visualize the problem, we imagine starting at some state x_0, and travelling along the flow of both vector fields in a "square" pattern: specifically, we will let $f_1(x)$ control our state for a short time, then switch to $f_2(x)$ for a short time, then back to $f_1(x)$ in the opposite direction, and finally back to $f_2(x)$ in the opposite direction. Because we're alternating the flow we're following, we won't necessarily end up back where we started, although we'll be pretty close; that small translation is what we're looking to measure, since it represents a direction in which our system can be controlled, which we probably couldn't have achieved by merely riding f_1 or f_2 alone. Think about this in the context of the parallel-parking problem!



We allow each stage of the trajectory to last for some very small time ϵ. From our control system, we can determine the endpoint of each stage:

$t = [0, \epsilon]$	$\dot{x} = f_1(x(t))$	$x(\epsilon) = \phi_\epsilon^{f_1}(x_0)$
$t = [\epsilon, 2\epsilon]$	$\dot{x} = f_2(x(t))$	$x(2\epsilon) = \phi_\epsilon^{f_2}(\phi_\epsilon^{f_1}(x_0)) = \phi_\epsilon^{f_2} \circ \phi_\epsilon^{f_1}(x_0)$
$t = [2\epsilon, 3\epsilon]$	$\dot{x} = -f_1(x(t))$	$x(3\epsilon) = \phi_\epsilon^{-f_1} \circ \phi_\epsilon^{f_2} \circ \phi_\epsilon^{f_1}(x_0)$
$t = [3\epsilon, 4\epsilon]$	$\dot{x} = -f_2(x(t))$	$x(4\epsilon) = \phi_\epsilon^{-f_2} \circ \phi_\epsilon^{-f_1} \circ \phi_\epsilon^{f_2} \circ \phi_\epsilon^{f_1}(x_0)$

The value we're looking for is $x(4\epsilon) - x_0$. To find this, we will consider the Taylor series approximation of each endpoint, starting with $x(\epsilon)$, and representing all terms of order n and greater as $O(\epsilon^n)$.

$$\begin{aligned}
x(\epsilon) &= \phi_\epsilon^{f_1}(x_0) \\
&= x_0 + \epsilon\dot{x}(0) + \frac{\epsilon^2}{2}\ddot{x}(0) + O(\epsilon^3) \\
&= x_0 + \epsilon\dot{\phi}_\epsilon^{f_1}(x(t))\Big|_{t=0} + \frac{\epsilon^2}{2}\ddot{\phi}_\epsilon^{f_1}(x(t))\Big|_{t=0} + O(\epsilon^3) \\
&= x_0 + \epsilon f_1(x_0) + \frac{\epsilon^2}{2}\frac{\partial f_1}{\partial x} f_1(x_0) + O(\epsilon^3)
\end{aligned}$$

$$\begin{aligned}
x(2\epsilon) &= \phi_\epsilon^{f_2}(x(\epsilon)) \\
&= x(\epsilon) + \epsilon f_2(x(\epsilon)) + \frac{\epsilon^2}{2}\frac{\partial f_2}{\partial x} f_2(x(\epsilon)) + O(\epsilon^3) \\
&= \Big(x_0 + \epsilon f_1(x_0) + \frac{\epsilon^2}{2}\frac{\partial f_1}{\partial x} f_1(x_0) + O(\epsilon^3)\Big) + \epsilon f_2\Big(x_0 + \epsilon f_1(x_0) + O(\epsilon^2)\Big) \\
&\quad + \frac{\epsilon^2}{2}\frac{\partial f_2}{\partial x}\Big(x_0 + O(\epsilon)\Big) f_2\Big(x_0 + O(\epsilon)\Big) + O(\epsilon^3) \\
&= x_0 + \epsilon\Big(f_1(x_0) + f_2\Big(x_0 + \epsilon f_1(x_0)\Big)\Big) + \frac{\epsilon^2}{2}\Big(\frac{\partial f_1}{\partial x} f_1(x_0) + \frac{\partial f_2}{\partial x} f_2(x_0)\Big) + O(\epsilon^3) \\
&= x_0 + \epsilon\Big(f_1(x_0) + f_2(x_0)\Big) + \frac{\epsilon^2}{2}\Big(\frac{\partial f_1}{\partial x} f_1(x_0) + \frac{\partial f_2}{\partial x} f_2(x_0) + 2\frac{\partial f_2}{\partial x} f_1(x_0)\Big) + O(\epsilon^3)
\end{aligned}$$

Note: here, we used the Taylor series approximation: $f_2(x_0 + \epsilon f_1(x)) = f_2(x_0) + \epsilon\frac{\partial f_2}{\partial x} f_1(x_0) + O(\epsilon^2)$.
We'll skip the pain of deriving the other two endpoints:

$$\begin{aligned}
x(3\epsilon) &= \phi_\epsilon^{-f_1}(x(2\epsilon)) \\
&= x_0 + \epsilon f_2(x_0) + \frac{\epsilon^2}{2}\Big(\frac{\partial f_2}{\partial x} f_2(x_0) + 2\frac{\partial f_2}{\partial x} f_1(x_0) - 2\frac{\partial f_1}{\partial x} f_2(x_0)\Big) + O(\epsilon^3)
\end{aligned}$$

$$\begin{aligned}
x(4\epsilon) &= \phi_\epsilon^{-f_2}(x(3\epsilon)) \\
&= x_0 + \epsilon^2\Big(\frac{\partial f_2}{\partial x} f_1(x_0) - \frac{\partial f_1}{\partial x} f_2(x_0)\Big) + O(\epsilon^3)
\end{aligned}$$

These two reverse stages cancel out all the terms of order ϵ! If ϵ is very small, we can assume that $O(\epsilon^3)$ is sufficiently smaller than the other terms, and we get:

$$x(4\epsilon) - x_0 = \epsilon^2\Big(\frac{\partial f_2}{\partial x} f_1(x_0) - \frac{\partial f_1}{\partial x} f_2(x_0)\Big) \tag{8.3}$$

Remember that this is in general a vector equation, and since each f represents a vector field, each $\frac{\partial f}{\partial x}$ is a Jacobian matrix.

Obviously, after all that derivation with those disgustingly nested Taylor series, the nice, simple, symmetric-looking quantity at the end deserves a special name...

8.3 The Lie bracket

We define the **Lie bracket** of two vector fields f and g, over the space parameterized by q, as

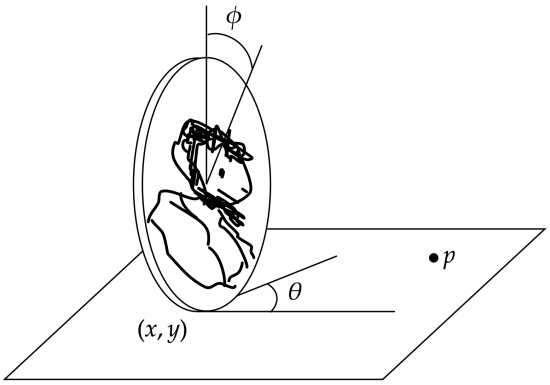
$$\boxed{[f,g](q) = \frac{\partial g}{\partial q} f(q) - \frac{\partial f}{\partial q} g(q)} \tag{8.4}$$

As we saw in Eq 1.3, this vector quantity represents the infinitesimal translation that arises as a consequence of traversing our "square". Note that this motion occurs on the order of ϵ^2, meaning it's a much slower motion to carry out than those directly controllable via our inputs, which are on the order of ϵ.

If $[f,g](q) = 0$, f and g are said to *commute*, meaning that travelling a square will always bring you back to where you started, no matter which field you pick first.

8.4 Example: Disk rolling on a plane

Imagine a penny of radius r constrained to roll on its edge (without slipping) along a flat plane, as depicted. If you can only control the rates of rolling ($\dot{\phi}$) and turning ($\dot{\theta}$), can you steer it to any arbitrary position p on the plane, such that when it arrives, it is right-side up — that is, $\phi = 0$?



With the no-slip constraint, we can write the following equations:

$$\dot{x} = r\dot{\phi}\cos\theta$$
$$\dot{y} = r\dot{\phi}\sin\theta$$

Define the state vector $q = [x, y, \theta, \phi]^T$.

$$\begin{bmatrix} 1 & 0 & 0 & -r\cos\theta \\ 0 & 1 & 0 & -r\sin\theta \end{bmatrix} \dot{q} = 0$$

Our control system lies in the nullspace of this matrix.

$$\dot{q} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_2 \qquad \text{where} \qquad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} \tag{8.5}$$

Let $f(q)$ be the vector field controlled by u_1, and $g(q)$ by u_2. Can alternating between these two allow us to control more of the penny's state? We use the Lie bracket to find out.

$$[f,g](q) = \frac{\partial g}{\partial q} f(q) - \frac{\partial f}{\partial q} g(q) = 0 \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -r\sin\theta & 0 \\ 0 & 0 & r\cos\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} r\sin\theta \\ -r\cos\theta \\ 0 \\ 0 \end{bmatrix} := h(q) \tag{8.6}$$

$h(q)$ is clearly linearly independent from f and g, meaning we've discovered a new direction in which to control our system! But we're not done yet. Using $h(q)$, we can calculate two more Lie brackets:

$$[f,h](q) = \frac{\partial h}{\partial q} f(q) - \frac{\partial f}{\partial q} h(q) = \begin{bmatrix} 0 & 0 & r\cos\theta & 0 \\ 0 & 0 & r\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -r\sin\theta & 0 \\ 0 & 0 & r\cos\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r\sin\theta \\ -r\cos\theta \\ 0 \\ 0 \end{bmatrix} = 0 \tag{8.7}$$

$$[g,h](q) = \frac{\partial h}{\partial q} g(q) - \frac{\partial g}{\partial q} h(q) = \begin{bmatrix} 0 & 0 & r\cos\theta & 0 \\ 0 & 0 & r\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} - 0 \begin{bmatrix} r\sin\theta \\ -r\cos\theta \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \\ 0 \end{bmatrix} := k(q) \tag{8.8}$$

As it turns out, $[f,h]$ didn't do us much good, but $[g,h]$ gave us yet another linearly independent vector field $k(q)$! Let's put the four vector fields into a matrix.

$$[f(q), g(q), h(q), k(q)] = \begin{bmatrix} r\cos\theta & 0 & r\sin\theta & r\cos\theta \\ r\sin\theta & 0 & -r\cos\theta & r\sin\theta \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{8.9}$$

As you can see, this matrix is nonsingular. This means, that, with only the two inputs u_1 and u_2 in Eq 8.5, we can theoretically control $\dot{q}$ to any vector in $\mathbb{R}^4$. While controlling h and k will ultimately be a lot slower than the others, the Lie bracket has provided a good test to prove that this range of motion is possible within our system.