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

Lecture 10: Sinusoids and Optimal Control

Scribes: Tarkan Al-Kazily and Josephine Koe

10.1 Sinusoids

Consider the following system of three states q_1, q_2, q_3 with dynamics given by

$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix} = \begin{bmatrix}u_1\\ u_2\\ q_1u_2 - q_2u_1\end{bmatrix} = \begin{bmatrix}1\\ 0\\ -q_2\end{bmatrix}u_1 + \begin{bmatrix}0\\ 1\\ q_1\end{bmatrix}u_2 = g_1(q)u_1 + g_2(q)u_2.$$

This is an underactuated system, where $\dot{q}_3 = q_1\dot{q}_2 - q_2\dot{q}_1$. Put this another way, we get the constraint $\dot{q}_3 - q_1\dot{q}_2 + q_2\dot{q}_1 = 0$. We can examine the Lie Bracket of the control flows to find potential motion in different directions.

$$[g_1, g_2] = \frac{\partial g_2}{\partial q}g_1 - \frac{\partial g_1}{\partial q}g_2 = \begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix}\begin{bmatrix}1\\0\\-q_2\end{bmatrix} - \begin{bmatrix}0&0&0\\0&0&0\\0&-1&0\end{bmatrix}\begin{bmatrix}0\\1\\q_1\end{bmatrix} = \begin{bmatrix}0\\0\\2\end{bmatrix}.$$

The Lie Bracket then indicates that the system is completely non-holonomic, since our distribution generated by g_1 and g_2 satisfies

$$\Delta = \text{Span}\{g_1, g_2, [g_1, g_2]\} = \mathbb{R}^3.$$

Now we ask ourselves how to steer from two states in the direction of our Lie Bracket flow, from $q(0)$ to $q(1)$, with some optimality in regards to our control input. Formalized,

$$q(0) = \begin{bmatrix}0\\0\\0\end{bmatrix},\ q(1) = \begin{bmatrix}0\\0\\a\end{bmatrix},\ \text{ find } u(t) \text{ for } \min_{u(t)} \int_0^1 \|u(t)\|^2 dt.$$

(Professor Sastry went on an only-somewhat-factual tangent about the Queen Dido of Carthage and described her as the founder of control theory. She figured out how to enclose the maximum area next to the sea given a length of cow hide.)

To solve this optimization, we use the fact that $u(t) = [u_1, u_2]^T$ gives us the time derivatives of our state $[q_1, q_2]^T$. Then we are trying to minimize $\dot{q}_1^2 + \dot{q}_2^2$. We also want to account for the constraint on $\dot{q}_3$, which we will weight using a Lagrange multiplier $\lambda(t)$. Our resulting equation takes the form of a Lagrangian (just like the Lagrangian we are used to, it is some function on the state and dynamics of our system)

$$L(q, \dot{q}) = \dot{q}_1^2 + \dot{q}_2^2 + \lambda(t)\left(\dot{q}_3 - q_1\dot{q}_2 + q_2\dot{q}_1\right).$$

It turns out that the same Euler-Lagrange equations will come in handy here to optimize the results. Applying these equations gives

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \frac{d}{dt}\left(\begin{bmatrix}2\dot{q}_1 + \lambda q_2\\ 2\dot{q}_2 - \lambda q_1\\ \lambda\end{bmatrix}\right) - \begin{bmatrix}-\lambda\dot{q}_2\\ \lambda\dot{q}_1\\ 0\end{bmatrix} = 2\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \dot{\lambda}\end{bmatrix} + 2\begin{bmatrix}\lambda\dot{q}_2\\ -\lambda\dot{q}_1\\ 0\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}.$$

Already this tells us a lot. We know that λ doesn't vary with time, and should therefore be a constant ($\dot{\lambda} = 0$). Additionally, we know that our optimal control law will need to satisfy the following differential equation

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\lambda \\ \lambda & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

See that this differential equation features a skew symmetric matrix $\Lambda = \begin{bmatrix} 0 & -\lambda \\ \lambda & 0 \end{bmatrix} = \hat{\lambda}$ in $so(2)$. We know that the solution to this equation is given by the matrix exponential, and better yet we have an easy closed form solution for this matrix exponential!

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = e^{\Lambda t} \begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} = \begin{bmatrix} \cos(\lambda t) & -\sin(\lambda t) \\ \sin(\lambda t) & \cos(\lambda t) \end{bmatrix} \begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} \tag{10.1}$$

Now we need to determine what value of λ provides optimal control. First, in order to keep the constraints that $q_1(0) = q_1(1) = 0$ and $q_2(0) = q_2(1) = 0$, we integrate the control input to find $q_1(t)$ and $q_2(t)$

$$\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} = \begin{bmatrix} q_1(0) \\ q_2(0) \end{bmatrix} + \int_0^t e^{\Lambda t} u(0) dt = \Lambda^{-1} \left(e^{\Lambda t} - I \right) u(0) \tag{10.2}$$

For $q_1(1) = q_2(1) = 0$, we must have that $e^{\Lambda} = I$, which forces λ to be of the form $2k\pi$. Obviously we need $\lambda > 0$, but what integer k will give us the best possible frequency?

We find this by looking at the constraint on q_3. Again integrating the dynamics to find $q_3(t)$, we get

$$q_3(t) = q_3(0) + \int_0^t q_1\dot{q}_2 - q_2\dot{q}_1 d\tau = \int_0^t \begin{bmatrix} q_1 & q_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} d\tau.$$

Using our closed form of $[q_1(t), q_2(t)]^T$ in equation (10.2) along with our closed form equation for $[\dot{q}_1(t), \dot{q}_2(t)]^T$ in equation (10.1), we can simplify our integrand.

$$\begin{bmatrix} q_1(t) & q_2(t) \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \left(\Lambda^{-1} \left(e^{\Lambda t} - I \right) u(0)\right)^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} e^{\Lambda t} u(0) = -\frac{1}{\lambda} u(0)^T (e^{\Lambda t} - I) u(0) \tag{10.3}$$

Integrating equation 10.3 to use the constraint that $q_3(1) = a$ gives us the equation

$$q_3(1) = \frac{1}{\lambda} \int_0^1 u(0)^T (I - e^{\Lambda \tau}) u(0) d\tau = \frac{1}{\lambda} \|u(0)\|^2 = a.$$

Since we want to minimize our control function which we've shown to satisfy $\|u(0)\|^2 = \lambda a$, the minimum cost is achieved when λ is smallest, which must be $\lambda = 2\pi$.

In this way, we've shown that the optimal control in the direction of $[g_1, g_2]$, which results in a flow $\dot{q}_3$, is given by the sinusoidal control law

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} \cos(2\pi t) & -\sin(2\pi t) \\ \sin(2\pi t) & \cos(2\pi t) \end{bmatrix} \begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix}.$$

See MLS Chapter 8 Section 2 for more details on this problem.

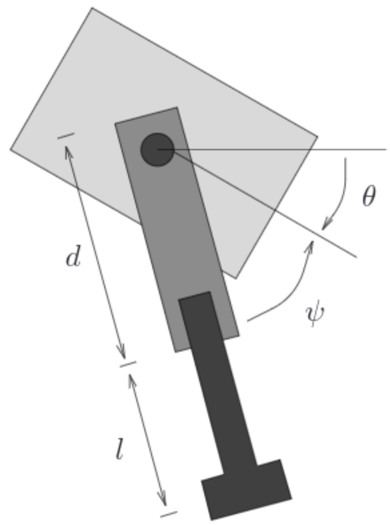

Figure 7.5: A simple hopping robot.

10.2 Hopper Example

Let's return to the example of a flipping hopping robot from the previous lectures. As derived from the conservation of rotational intertia, the system dynamics can be given by

$$\begin{bmatrix} \dot{\psi} \\ \dot{l} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \frac{m(l+d)^2}{I+m(l+d)^2} u_1 \end{bmatrix}$$

However, with a clever change of states to using $\alpha = \theta + \frac{md^2}{I+md^2}\psi$, we get the following system model

$$\begin{bmatrix} \dot{\psi} \\ \dot{l} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \frac{2mdI}{(I+md^2)^2} l u_1 + O(l^2) u_1 \end{bmatrix}$$

Clearly we've got a much nicer form of control law for $\dot{\alpha}$ than we had for $\dot{\theta}$. We'll let $f(l)$ represent the coefficient to u_1 so that $\dot{\alpha} = f(l)u_1$. What we want to do with this hopping robot is to get it to do midair somersaults. In terms of the system state $q = [\psi, l, \alpha]^T$, that means going from $q(0) = [0,0,0]^T$ to $q(1) = [0,0,2\pi]^T$. We have direct control over $\dot{\psi}$ and $\dot{l}$ with u_1 and u_2 respectively, so in order to optimally go in the direction of $\dot{\alpha}$ we'll need to incorporate sinusoids.

Let $u_1 = a\cos(2\pi t)$ and $u_2 = a\sin(2\pi t)$ - these control sinusoids have frequency specifically chosen to complete the maneuver in one second. Now, these controls satisfy our requirements on ψ and l due to the periodicity of cos and sin:

$$\psi(1) = \psi(0) + \int_0^1 a\cos(2\pi\tau)d\tau = 0$$
$$l(1) = l(0) + \int_0^1 a\sin(2\pi\tau)d\tau = 0$$

To check that the requirement on $\alpha(1) = 2\pi$ is satisfied, we'll use the Fourier Series representation for $f(l)$, given by coefficients α_i and β_i

$$f(l) = \alpha_0 + \alpha_1 \cos(2\pi t) + \beta_1 \sin(2\pi t) + \alpha_2 \cos(4\pi t) + \beta_2 \sin(4\pi t) + \dots.$$

Since $u_1 = a\cos(2\pi t)$, we can use the Fourier series representation of $f(l)$ to quickly integrate $\dot{\alpha}$ to find $\alpha(1)$:

$$\begin{aligned}
\alpha(1) &= \alpha(0) + \int_0^1 f(l) u_1 d\tau \\
&= \int_0^1 (\alpha_0 + \alpha_1 \cos(2\pi\tau) + \beta_1 \sin(2\pi\tau) + \alpha_2 \cos(4\pi\tau) + \beta_2 \sin(4\pi\tau) + \dots) a \cos(2\pi\tau) d\tau \\
&= a\alpha_1 \int_0^1 \cos^2(2\pi\tau) d\tau + a \int_0^1 \alpha_0 \cos(2\pi\tau) + \beta_1 \sin(2\pi\tau)\cos(2\pi\tau) + \alpha_2 \cos(4\pi\tau)\cos(2\pi\tau) + \dots d\tau \\
&= \frac{a\alpha_1}{2} + 0.
\end{aligned}$$

Therefore we can establish that $\alpha(1) = 2\pi$ with appropriate choice of amplitude a, given that we know the Fourier coefficients of $f(l)$.

10.3 Generalization of Sinusoids for Multiple Frequencies

Now we can use these methods to generalize the steering problem to cars with n trailers using sinusoids of multiple frequencies. Model the system as follows:

$$\begin{aligned}
x &= \dot{q}_1 = u_1 \\
\phi &= \dot{q}_2 = u_2 \\
\theta &= \dot{q}_3 = q_2 u_1 \\
y &= \dot{q}_4 = q_3 u_1
\end{aligned}$$

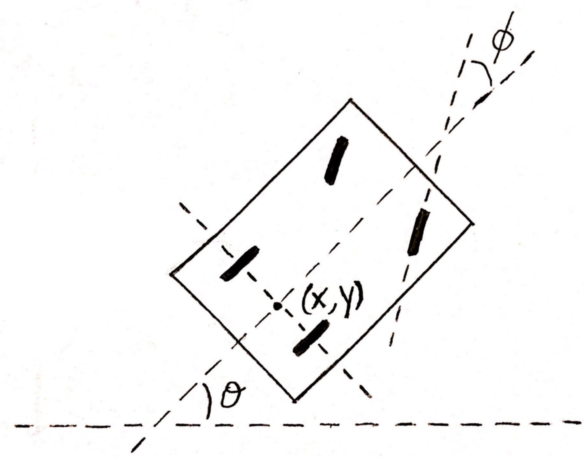

Figure 10.1: Model of a car described by the position of the back axle's midpoint, the angle of the back wheels to the frame, and the steering angle of the front wheels.

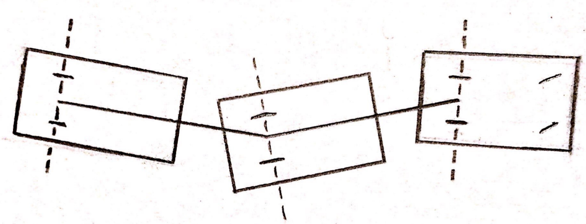

Figure 10.2: Model of a car with trailers.

The system can be described with the equation $\dot{q} = g_1 u_1 + g_2 u_2$, where $g_1 = \begin{bmatrix} 1 & 0 & q_2 & q_3 \end{bmatrix}^T$ and $g_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$. First compute the Lie bracket of $[g_1, g_2]$, then compute $[g_1, [g_1, g_2]]$

$$[g_1, g_2] = -\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$[g_1, [g_1, g_2]] = -\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Putting them all together, the matrix $\begin{bmatrix} g_1 & g_2 & [g_1, g_2] & [g_1, [g_1, g_2]] \end{bmatrix}$ has been built to a dimension of 4 and the columns are linearly independent. The system has 2 constraints and is completely non-holonomic.

How do you steer this system? Using sinusoids, you can construct the following plan:

$$
\begin{aligned}
&q_1(0) \xrightarrow{1} q_1(1) \xrightarrow{a\cos(2\pi t)} q_1(2) \\
&q_2(0) \xrightarrow{1} q_2(1) \xrightarrow{a\sin(2\pi t)} q_2(2) \\
&q_3(0) \to q_3'(1) \xrightarrow{a\cos(2\pi t)} q_3(2) = q_3'(1) + \frac{a^2}{2} \qquad (q_3(2) \text{ is now the desired final value of } q_3) \\
&q_4(0) \to q_4'(1) \to q_4''(2)
\end{aligned}
$$

q_3 and q_4 are drift variables of the system. Using sinusoids of different frequencies, you can get motion in the direction of the Lie brackets but have no net motion in q_1 and q_2. This is demonstrated in later steps, when you choose $u_1 = a\cos(2\pi t)$ and $u_2 = b\sin(4\pi t)$.

$$
\begin{aligned}
q_1 &= \frac{a}{2\pi}\sin(2\pi t) \\
q_1(4) &= \int_0^1 \frac{a}{2\pi}\sin(2\pi\tau)d\tau + q_1(3) \\
&= 0 + q_1(3) \\
&= q_1(3) \\
q_2(4) &= \int_0^1 -\frac{b}{4\pi}\cos(2\pi\tau)d\tau + q_2(3) \\
&= 0 + q_2(3) \\
&= q_2(3) \\
q_3(4) &= \int_0^1 \frac{ab}{(4\pi)(2\pi)}\cos(2\pi\tau)\cos(4\pi\tau)d\tau + q_3(3) \\
&= \int_0^1 \frac{ab}{(4\pi)(2\pi)(2)}(\cos(2\pi\tau) + \cos(6\pi\tau))d\tau + q_3(3) \\
&= 0 + q_3(3) \\
&= q_3(3) \\
q_4(4) &= \int_0^1 \frac{a^2 b}{(4\pi)^2(2\pi)}\cos^2(2\pi\tau)\cos(4\pi\tau)d\tau + q_4(3)
\end{aligned}
$$

The $\cos^2$ term in q_4 prevents the integral from becoming 0 and instead becomes a nonzero DC term. When you double the frequency, you do not need to keep track of the previous variables that you drove to their desired locations because the integral becomes 0 and they end up back in the place you drove them to (as seen with q_1, q_2, and q_3). Using sinusoids results in traversing in the directions of the Lie brackets in a smooth way because it is like a continuous version of moving in positive and negative directions.

For a general case, first steer q_1 and q_2 to their desired values. Then for each q_{k+2} where $k \geq 1$, steer q_{k+2} to its final value using $u_1 = a\sin(2\pi t), u_2 = b\cos(2\pi kt)$ such that $q_{k+2}(1) = (\frac{a}{4\pi})^k \frac{b}{k!} + q_{k+2}(0)$.

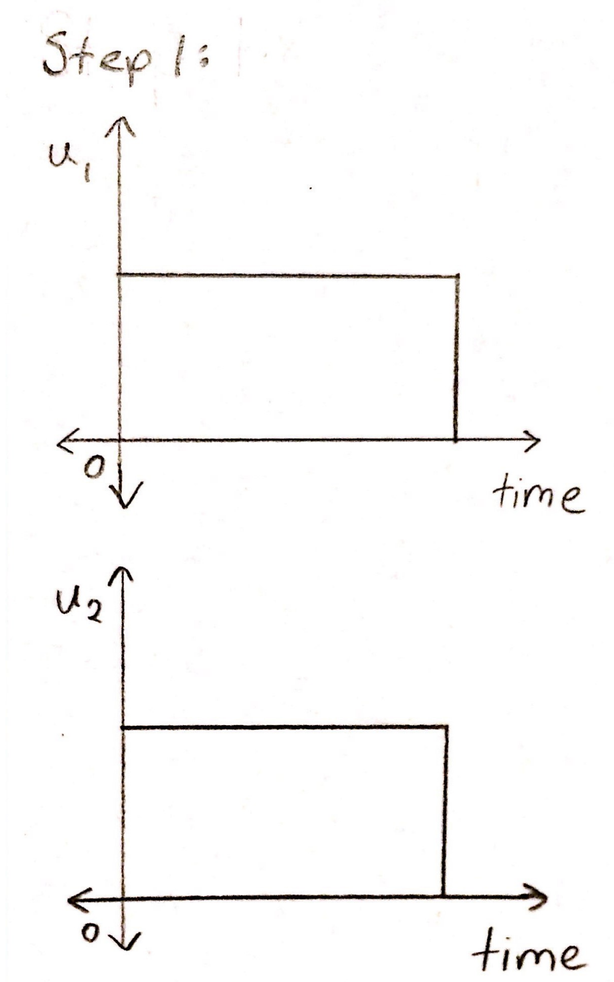

Figure 10.3: Graphs of inputs u_1 and u_2 over time when moving in the directions of g_1 and g_2.

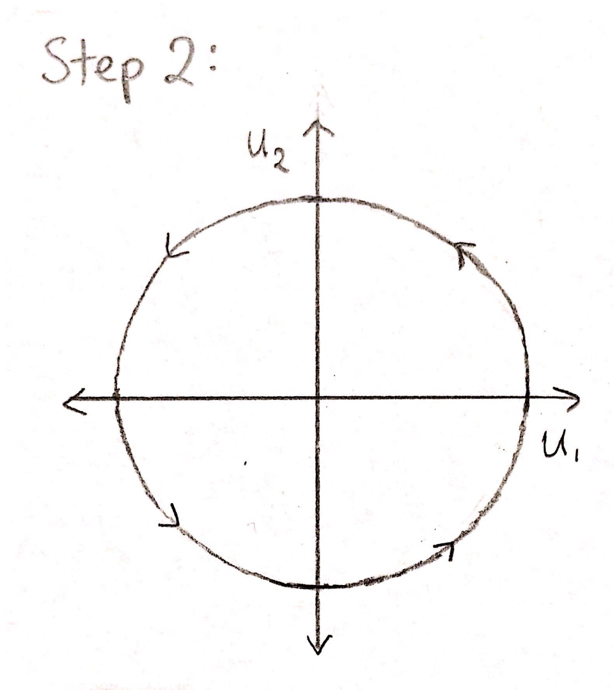

Figure 10.4: Graph of input u_1 vs input u_2 over one second when moving in the direction of $[g_1, g_2]$.

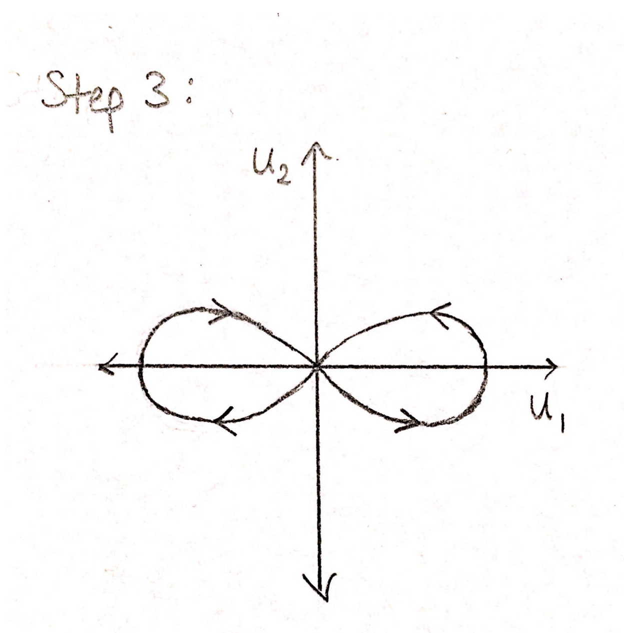

Figure 10.5: Graph of input u_1 vs input u_2 over one second when moving in the direction of $[g_1, [g_1, g_2]]$. u_2 is moving twice as fast as u_1.

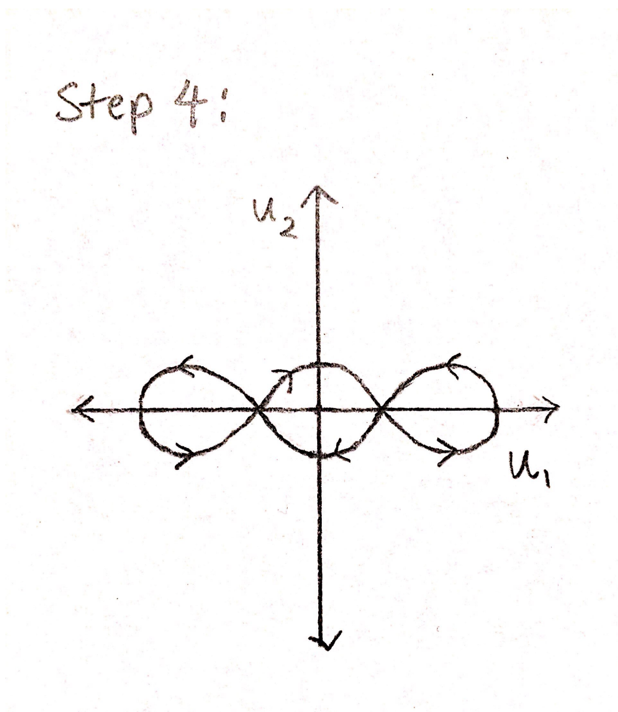

Figure 10.6: Graph of input u_1 vs input u_2 over one second when moving in the direction of an additional Lie bracket. u_2 is moving three times as fast as u_1.

The when moving in the directions of the Lie brackets, the plots of the inputs become Lissajous curves. The system is an example of a chain form system.

Definition *Chained form* systems are higher order systems that can be steered using sinusoids at integrally related frequencies. They are the duals of the Goursat form.

This method can also be applied to the parallel parking problem. In parallel parking, you move to next to where you want to be and line up with the parking spot. Then you make small movements that result in movement in the perpendicular (constrained) direction.

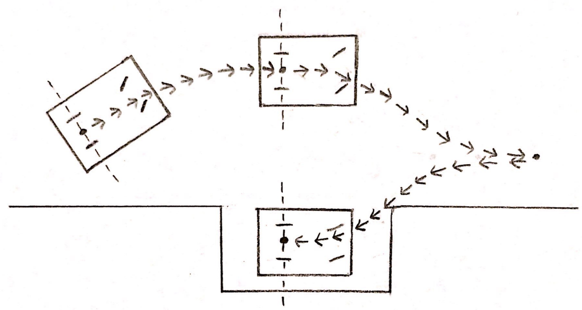

Figure 10.7: Diagram of parallel parking trajectory.

The resulting paths are in the shape of a cusp, which is where the sin and cos functions cross each other. The inputs would be in the following form:

$$\begin{aligned} u_1 &= a_0 + a_1 \cos(2\pi t) \\ u_2 &= b_0 + b_1 \sin(2\pi t) + b_2 \sin(4\pi t) \end{aligned}$$

10.4 The Big Questions

How do we go from the standard car model to this chained form?

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\theta)u_1 \\ \sin(\theta)u_1 \\ \frac{1}{\ell}\tan(\phi)u_1 \\ u_2 \end{bmatrix} \overset{?}{\rightarrow} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ q_2 v_1 \\ q_3 v_1 \end{bmatrix} \tag{10.4}$$

The kinematic model of the car becomes a combination of a nonlinear change of coordinates and state feedback and looks like this:

$$\begin{aligned} \dot{x} &= \cos(\theta)u_1 && \text{(looks like } \theta \text{ when } \theta \approx 0) \\ \dot{y} &= \sin(\theta)u_1 && \text{(looks like 0 when } \theta \approx 0) \\ \dot{\theta} &= \frac{1}{l}\tan(\phi)u_1 && \text{(as long as } \phi \neq \frac{\pi}{2}) \\ \dot{\phi} &= u_2 \end{aligned}$$

If there exists a $v = \alpha(x) + B(x)u$, then $q = \phi(x)$ where ϕ is the transformation that maps $\mathbb{R}^n \rightarrow \mathbb{R}^n$ and moves the nonholonomic control system to chained form. While it isn't immediately obvious, some non-linear coordinate transformation will be necessary to go from the first model to the second.

For the system with trailers, different trailers will be at different angles, but the point of interest is the midpoint of the end (last) trailer. Parallel parking of the last trailer is the hardest, but following the sinusoidal trajectory should not affect the previously positioned trailers.

Consider the system

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

If it turns out that

$$\Delta_0 = \{g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]]\}$$

has a rank of n and is involute, and

$$\Delta_1 = \{g_2, [g_1, g_2], [g_1, [g_1, g_2]]\}$$

has a rank of $(n-1)$ and is involute, and

$$\Delta_2 = \{[g_1, g_2], [g_1, [g_1, g_2]]\}$$

has a rank of $(n-2)$ and is involute, then there exists a nonlinear change of coordinates that converts the system into one-chain form.

Is this the same as feedback linearization? No. The first 2 conditions apply to feedback linearization, but the third form is a requirement for the system to be in chain form.

Definition Systems that are not linearizable on their own but can be linearized through dynamic extension are called *flat systems*.

The problem with traditional RRTs is that they don't take the dynamics of the system into account and assume you can move in any directions, even ones which have constraints on them. To fix this, you can choose coarse pieces of the trajectory to be compatible to the dynamics and have an "alphabet" with the system's kinematic constraints. The local planners in RRTs assume that you can steer in the direction of the goal and get closer to it, but you have to make sure it is dynamically feasible. After that, you have to reprove that it does not crash into any obstacles.