ECPS 208: Control Systems for Cyber Physical Systems
Dubin's car project
Spring 2025

Dubin's car is a simple model to study (nonlinear) control design.

Step 1: Model. Consider a car with mass 1 (one) moving on a frictionless surface. Denote the two horizontal coordinates and p_x and p_y. It can have 2 actuators. The first one controls the rate of change of heading (i.e., $\dot{\theta} = w = u_1(t)$, θ being the angle that the direction of velocity makes with the x-axis. The second one would be (later on this quarter) the actual velocity: $u_2(t) = v$. We will start with the initial assumption that v is fixed (i.e., we set it a specific value, via cruise control).

What is the third order (and nonlinear) model of this motion: use p_x and p_y and θ as the state vector.

Step 2: Simple Control. (see Example-1.m) Design u_1, assuming we can measure p_x, $_y$ and θ, to deliver the car to target point g_x, and g_y (target positions are known). Start with, $(0, 0)$ or $(0, p_{y,o})$ and get to position $(5, 5)$.

- Gains to play with v , and gain of the control law. For initial values ...
- Plot the results. Make is cure/pretty/interesting
- What happens when it gets to the target? Do you need a kill switch? What are a few good ways to do so?
- Can you reverse the target and start positions?
- What happens if you increase the gain a lot OR use small gains?

Step 3: Second Actuator (for this step only). Suppose you wanted to also control velocity based on your position. What do you suggest? How does that change the speed of reaching target?

Step 4: Avoiding an Obstacle: (see Example-2):

- Work out the details of why this is a 'relative degree 2' problem (i.e., show you need to $\psi_(s)$ instead of $h(x)$ to get the control w to show up (answer is actually in code too!)
- Can you force an standstill such that it stops? Can you add a dither to fix it?

Step 5: Two Robots Criss-crossing (see Example-3) : Play with the size of the safe distance. Change direction of motion, can you make it stop (to be safe)?

- Work out the details of why the new safety constraint is also a 'relative degree 2' problem (i.e., show you need to $\psi_(s)$ instead of $h(x)$ to get the control w to show up (answer is actually in code too!)

- Can you force it to fail by using a large safety bound? What if you shrink the size of the safe distance?
- Can you make it do weird things? Can you explain why it behave strange? Can you fix it?

Step 6: Two robots criss-crossing and avoiding obstacle (see Example 4) :

Repeat the steps for Step 5.

Step 7: Many (3) robots, criss-crossing toward targets (extend Example 3) :

- Need to code the third agent in Example-3
- Can you get it to work reliably OR need to fine tune safe distance, targets, etc.

Step 8: Many (3) robots criss-crossing and avoiding obstacle (extend Example 4):

- Need to code the third agent in Example-4
- Can you get it to work reliably OR need to fine tune safe distance, targets, etc.