AE353 Homework #9:
Control Design — Differential Drive Robot Race

(due at the beginning of class on Wednesday, April 29)

Goal

This week, you will design a control system that makes a differential drive robot race along a road as fast as possible. You will implement your control system by modifying the script:

`hw9_ControlLoopTemplate.m`

You will test your control system with the script:

`hw9_Simulation.m`

Instructions on exactly how to do these two things will be posted to piazza.

Model

The dynamics of the robot are:

$$
\begin{aligned}
\dot{q}_1 &= v \cos q_3 \\
\dot{q}_2 &= v \sin q_3 \\
\dot{q}_3 &= w \\
\dot{w}_R &= k_\tau \tau_R \\
\dot{w}_L &= k_\tau \tau_L
\end{aligned}
$$

where

$$v = \frac{r}{2}\left(w_R + w_L\right)$$

is the *forward speed*,

$$w = \frac{r}{2b}\left(w_R - w_L\right)$$

is the *turning rate*, and

- $r = 0.15$ is the radius of each wheel;
- $b = 0.25$ is half the distance between each wheel (i.e., half the length of the axle);
- $k_\tau = 1$ is a torque constant;
- w_R and w_L are the right and left wheel angular velocities, respectively;
- (q_1, q_2) is the position of the robot (i.e., of the center of the axle);
- q_3 is the heading angle of the robot (in radians).

One way to make the robot follow a road is to make the robot track a *given* trajectory q_{des} that itself follows the road and that satisfies

$$\begin{aligned}
\dot{q}_{\text{des},1}(t) &= v_{\text{des}}(t)\cos\left(q_{\text{des},3}(t)\right) \\
\dot{q}_{\text{des},2}(t) &= v_{\text{des}}(t)\sin\left(q_{\text{des},3}(t)\right) \\
\dot{q}_{\text{des},3}(t) &= w_{\text{des}}(t)
\end{aligned}$$

for some forward speed v_{des} and turning rate w_{des}, both of which can be functions of time. If we apply a coordinate transformation to define

$$e = \begin{bmatrix} \cos q_3 & \sin q_3 & 0 \\ -\sin q_3 & \cos q_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(q - q_{\text{des}}\right)$$

then e_1 is the lateral position error (i.e., perpendicular to centerline of road at q_{des}), e_2 is the longitudinal position error (i.e., parallel to centerline of road at q_{des}), and e_3 is the heading error. The error dynamics can be written as

$$\begin{aligned}
\dot{e}_1 &= v - v_{\text{des}}\cos e_3 + e_2 w \\
\dot{e}_2 &= v_{\text{des}}\sin e_3 - e_1 w \\
\dot{e}_3 &= w - w_{\text{des}}.
\end{aligned}$$

If we linearize about $e \approx 0$ and assume both v_{des} and w_{des} are piecewise-constant, we have

$$\begin{aligned}
\dot{e}_1 &= v - v_{\text{des}} \\
\dot{e}_2 &= v_{\text{des}} e_3 \\
\dot{e}_3 &= w - w_{\text{des}}.
\end{aligned} \tag{1}$$

If we define

$$\begin{aligned}
e_4 &= v - v_{\text{des}} \\
e_5 &= w - w_{\text{des}},
\end{aligned} \tag{2}$$

then it is easy to show that

$$\begin{aligned}
\dot{e}_4 &= \frac{r k_\tau}{2}\left(\tau_R + \tau_L\right) \\
\dot{e}_5 &= \frac{r k_\tau}{2b}\left(\tau_R - \tau_L\right).
\end{aligned} \tag{3}$$

Equations (1) and (3) can be combined and—eliminating v and w from (1) using (2)—put into state space form (where the state is the 5×1 vector e and the inputs are τ_L and τ_R). The result is a model that you can use to design a controller. Note that you may assume knowledge of v_{des} and w_{des} (but only at the current time, not in the future). Given access to measurements of e_1, e_2, w_R, and w_L, you can also design an observer. Implement both controller and observer in `hw9_ControlLoopTemplate.m`, and you have yourself a control system.

How To Go Fast

The script `hw9_ControlLoopTemplate.m` gives you the ability, if you want, to change the desired forward speed v_{des}. If you increase it, your robot will try to go faster — but be careful! Try to go too fast and you'll run off the road. You'll have to figure out for yourself what to do with v_{des}. A simple strategy for the race would be to test your control design many times, setting v_{des} to the highest constant value that doesn't often result in a crash.

What to Turn In

You may work, if you like, with one partner. (You may also, as usual, work and share code with other people, as long as you include a list of all the people with whom you collaborated.) The two of you should submit the following:

- A single MATLAB script to replace `hw9_ControlLoopTemplate.m`, with your final control system. You must call this script `hw9_NAME1_NAME2.m`, where "NAME1" and "NAME2" are replaced with the first five letters of your first name (in capitals) followed by the first letter of your last name (in capitals) and—if you are working in a group—your teammate. For example, if I was working alone, I would submit a script with the file name `hw9_TIMB`. Details on how to submit your code will be posted to piazza.

- A brief description of your design process. If you use state space methods, this process should at least include the following:

  - Derivation of a state space model of the system (starting from what is described above).
  - Analysis of controllability and observability.
  - Design of controller and observer, either by eigenvalue placement or by optimality.
  - Simulation results, and their use to refine or validate your control system.

  If you use classical methods, the design process should at least include the following:

  - A block diagram model of the system.
  - Analysis of time response.
  - Analysis of frequency response.
  - Simulation results, and their use to refine or validate your control system.

  Please be very concise. A formal report is not required.

As discussed in class, we will have a contest in class on the due date. Details of this contest—and of opportunities for extra credit—will be posted online.