ME131 Vehicle Dynamics and Control

HW7: Path Tracking

Assigned: 4/24/2019 Due: 5/3/2019, 11:59pm (On bCourses)

Please submit your homework solutions on bCourses as a single PDF of your solutions. When videos are required, please only submit the link as part of the solution PDF document. Late homeworks will be penalized.

Problem 1 Lanekeeping, Stability, and Transient Response (20pt)

For the following question, refer to *lecture_Path_Tracking_Equations_of_Motion.pdf*. When a vehicle is referenced relative to the road, it goes from being a two-state (β, r) system to a four-state $(\dot{e}, \ddot{e}, \Delta\dot{\Psi}, \Delta\ddot{\Psi})$ system (Eq. 5.48). As a result, there are now two pairs of poles that influence the system dynamics. This problem asks you to look at the poles of the system for a number of different cases.

We define the following parameter values for our system:

$$
\begin{aligned}
C_f &= 20000 \text{ [N/rad]} \\
C_r &= 20000 \text{ [N/rad]} \\
m &= 1650 \text{[kg]} \\
I_z &= 2235 \text{ [kg m}^2\text{]} \\
L &= 2.468 \text{ [m]} \\
W_f &= 0.57 \\
a &= L(1 - W_f) \text{ [m]} \\
b &= LW_f \text{ [m]}
\end{aligned}
$$

1.1 (8pt) Write a function that calculates the A matrix with inputs of (U_x, k_p, x_{LA}). Find the poles of the system when the lanekeeping gain k_p is set to 0 and there is no lookahead. These are the open loop poles of the system.

Deliverable: Plot the poles on the real-imaginary axis for speeds of 10 m/s and 20 m/s, all on the same figure.

1.2 (a) (3pt) Now consider steering feedback proportional to the lateral error (i.e. $\delta_{FB} = -k_p e$) from Eq. 5.51. Set k_p such that the feedback steering angle is one degree for a lateral error of 1 meter (remember to convert to radians).

Deliverable: Plot the poles of the system for speeds of 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s, and 30 m/s, all on the same figure.

(b) (3pt) You should see the poles forming two pairs - one associated with the underlying vehicle dynamics and one associated with the error states.

Deliverable: This gain is actually fairly strong - does it manage to move the poles far from the origin? For what speeds is the system stable?

1.3 (a) (3pt) Keeping k_p at 1 degree/meter, let's add lookahead so the control law becomes:

$$\delta = -k_p(e + x_{LA}\Delta\Psi) \tag{1}$$

Deliverable: For a speed of 25 m/s, plot the system poles for lookahead distances of 1m, 10 m, 20m, and 100m, all on the same figure.

(b) (3pt) You should see a lot of change takes place between 20 and 100 meters.

Deliverable: Plot the poles of this system for lookahead distances between 20 and 100 meters using 5 meter increments. If we divide the poles into those associated with the vehicle dynamics and those associated with position, what happens to each of these pairs of poles as the lookahead increases?

Problem 2 Path Tracking (15pt)

Now let's move on to a simulation of a path that is not always straight (refer to *lecture_Driving_on_a_Curved_Path.pdf*). The following has been implemented for you: First modify your equation of motion for $\Delta\Psi$ to include the effect of curvature (Eq. 6.4). Since our curvature $\kappa(s)$ will vary along the length of the path, you should also now add in a state for s as well. Your simulation should now have states for U_y, r, s, e, and $\Delta\Psi$. Use *animateSim.m* to animate a vehicle driving on an oval track.

2.1 (5pt) Assume the vehicle starts at $s = 0$, with 0 heading and lateral error, and drives at a constant speed of $U_x = 8$ m/s. Using only the feedback controller without lookahead from Problem 1, simulate the vehicle until the end of the path. Set k_p at 1 degree/meter.

Deliverable: Plot the lateral error and heading error as functions of time as the car traverses the path.

2.2 (5pt) Now add the lookahead term. Set k_p at 1 degree/meter and x_{LA} to 10 meters. How does this change the trajectory of the car?

Deliverable: Plot the lateral error and heading error. Compare this to the plot from 2.1.a.

2.3 (5pt) Now add the feedforward steering with steady-state sideslip term in Eq. 6.30 and Eq 6.31. You should see much better tracking.

Deliverable: Plot the lateral error and heading error. Compare this to the plot from 2.1.a and 2.1.b.

Problem 3 Electronic Stability Control Design (65pt)

In this problem you will design an Electronics Stability Control System in MATLAB (Note: there is no `SIMULINK` file). All available files are in the provided .zip file. They include tire model data (`tireModel.mat`) and the following "template" files.

- `main.m`
- `referenceGen.m`, `ESCdlqr.m`, `brackingLogic.m`, `vehicleDyn.m`

The .m files represent functions in the following ESC control block diagram.

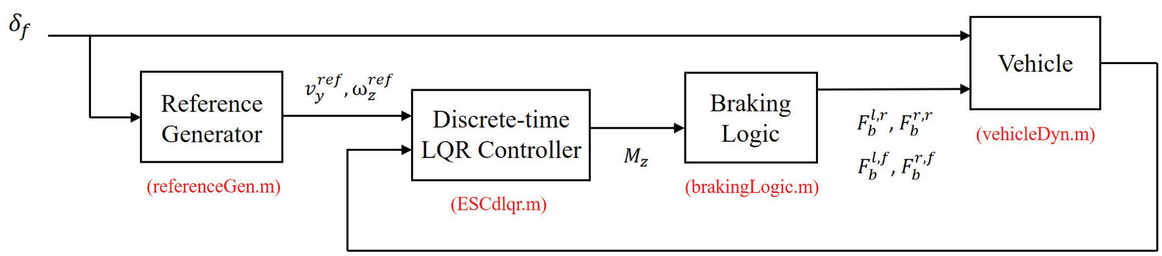

Figure 1: Control strategy for ESC (`main.m`)

Define v_x, v_y as the longitudinal and lateral velocities of the vehicle; ψ, ω_z as the yaw angle and the yaw rate of the vehicle; X, Y as the global position of the vehicle in the inertial frame.
We study the following scenario. The vehicle starts from the initial condition $v_x = 10$ m/s , $v_y = 0$ m/s, $\psi = \pi/2$ rad, $\omega_z = 0$ rad/s, $X = 0$ m and $Y = 0$ m and the driver provides a steering input shown in the following figure. We want to design an ESC controller and compare what happens when it is on or off.

The main code (`main.m`) has 7 parts: (1) Load vehicle parameters, (2) Compute driver's steering input, (3) Reference test, (4) Set initial conditions, (5) Run simulation with ESC off, (6) Run close-loop simulation with ESC on, (7) Plot routines. Note that the code does not run as it is currently written. You will first need to fill in some parts of it.

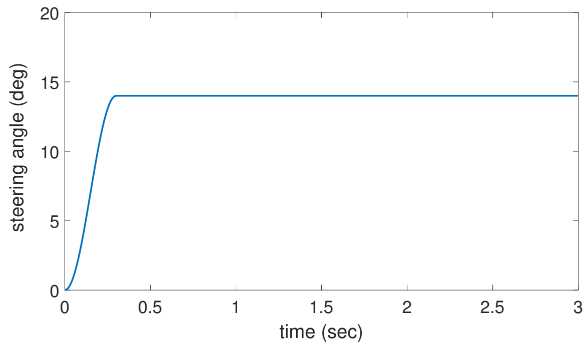

Figure 2: Driver steering input

3.1 (10pt) Open the main function `main.m` and look at blocks (1)-(3). Complete the function `referenceGen.m` to output the correct v_y^{ref} ("`vyRef`" in the code) and ω_z^{ref} ("`yawRateRef`" in the code). The equations for reference computation can be found in your notes. Recall that $\beta = v_y/v_x$.

Deliverable:

- Run blocks (1)-(3) of `main.m` and look at block (3). In block (3) the function `referenceGen.m` is called with inputs ($\delta_f = 5 \times \frac{\pi}{180}$, $v_x = 10$). Report the two outputs.
- In block (3) the function `referenceGen.m` is called with inputs ($\delta_f = 30 \times \frac{\pi}{180}$, $v_x = 10$). Report the two outputs.

3.2 (25pt) You will design a discrete-time ESC system by using the following lateral dynamics bicycle model:

$$\begin{bmatrix} \dot{v}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} v_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} \delta_f + \begin{bmatrix} c_{11} \\ c_{21} \end{bmatrix} M_z \tag{2}$$

with

$$\begin{aligned} a_{11} &= -\frac{2(C_\alpha^f + C_\alpha^r)}{mv_x}, & a_{12} &= -\frac{2(C_\alpha^f l_f - C_\alpha^r l_r)}{mv_x} - v_x \\ a_{21} &= -\frac{2(C_\alpha^f l_f - C_\alpha^r l_r)}{I_z v_x}, & a_{22} &= -\frac{2(C_\alpha^f l_f^2 + C_\alpha^r l_r^2)}{I_z v_x} \\ b_{11} &= \frac{2C_\alpha^f}{m}, & b_{21} &= \frac{2 l_f C_\alpha^f}{I_z} \\ c_{11} &= 0, & c_{21} &= \frac{1}{I_z} \end{aligned} \tag{3}$$

where

- $m = 2237$ Kg is the mass of the vehicle.
- v_x is the longitudinal velocity.
- $C_\alpha^f = 59563$ N/rad is front tire cornering stiffness.
- $C_\alpha^r = 88998$ N/rad is rear tire cornering stiffness.
- $l_f = 1.8$ m is the distance between c.g. and front axle.
- $l_r = 1.22$ m is the distance between c.g. and rear axle.
- $I_z = 5550.045$ Kg m^2 is the yaw moment of inertia.
- δ_f is the front wheel steering angle.

Open the function `ESCdlqr.m`. Most of the code is already all in there. Implement a discrete-time LQR controller with sampling rate $\mathbf{dt = 0.01}$ sec. Run part (4)-(7) of the file `main.m`. You should see three plots:

1. Tracking performance (ESC on vs ESC off): the v_y [m/s] and v_y^{ref} [m/s] vs time [sec] and the ω_z [deg/s] and ω_z^{ref} [deg/s] vs time [sec] in one plot.
2. Vehicle trajectory (ESC on vs ESC off): the X [m] vs Y [m/s] in one plot.
3. Vehicle yaw angle (ESC on vs ESC off): the ψ [deg] vs time [sec] in one plot.

Deliverable: Tune the LQR controller and submit the three plots described above. A well-tuned ESC should give you a difference of one meter or more on lateral position at the end of the maneuver when compared to a maneuver without ESC.

3.3 (10pt) According to the the above plots, what type of behavior are you observing (and correcting with the ESC): Understeer or oversteer?

Deliverable: Understeer or oversteer answer.

3.4 (20pt) Confirm your answer at the previous point with a short explanation and by attaching the plots of front and rear tire slip angles during the maneuver with ESC on and ESC off. (Note you need to compute the slip angles AND need to add a plot in your main function)

Deliverable:

- A short explanation
- One plot of front and rear tire slip angles [rad] vs time [sec] for ESC on and ESC off. With a legend which describes the four signals.