Longitudinal Dynamics: Model-Based Cruise Control

In Lab 5 you designed a PI cruise controller. This week you will design a *model-based* LQR cruise controller for your BARC vehicle.

Task 1 Longitudinal Vehicle Dynamics Model - No Deliverables

A simple model for longitudinal vehicle dynamics (under a no-slip assumption) moving on an inclined road can be expressed according to Newton's 2nd law:

$$m\frac{dV}{dt} = F_x - R_x - mg\ \text{sin}(\theta) - F_{aero} \tag{1}$$

where F_x is the traction force, R_x is the sum of rolling resistance forces at the front and rear wheels, θ is the angle of inclination of the road, and F_{aero} is the equivalent aerodynamic drag force. **NOTE: In this lab F_x will be the control input to the system.** F_{aero} can be further represented as

$$F_{aero} = \frac{1}{2}\rho C_d A_F (V + V_{wind})^2, \tag{2}$$

where C_d is the vehicle's aerodynamic drag coefficient and A_F its frontal area. C_d can be roughly determined from a coast down test. In a coast down test, after an initial burst of acceleration to gain speed, the acceleration is reduced to zero and the vehicle is allowed to slow down solely under the effects of aerodynamic drag and rolling resistance. The longitudinal tire force applied by the motor under these conditions is small and can be assumed to be zero. Furthermore, the road incline, θ, and the wind velocity, V_{wind}, are both assumed to be zero. Therefore, the vehicle dynamics model can be simplified as:

$$-m\frac{dV}{dt} = \frac{1}{2}\rho V^2 A_F C_d + R_x. \tag{3}$$

You saw in last week's homework assignment that this can be further decomposed to yield

$$\frac{v(t)}{V_0} = \frac{1}{\beta}\ \text{tan}\left[(1 - \frac{t}{T})\ \text{tan}^{-1}\beta\right] \qquad \text{where } \beta = V_0\left(\frac{\rho A_F C_d}{2R_x}\right)^{\frac{1}{2}} \tag{4}$$

where T is the time the vehicle stops after the coast-down test (i.e. $V(T) = 0$).

Task 2 Parameter Estimation

In HW3, you solved the above eqn. (4) using experimental data from a full-scale vehicle. This week, you will perform your own coast-down tests in order to estimate the C_d and R_x values for your BARC. Coast-down tests should be performed on a horizontal ground with low to no wind velocity, such as an indoor hallway.

1. Perform five coast-down tests on horizontal ground. Apply a short burst of acceleration and do not apply any traction force after that. In our demo car we used a motor PWM of 1650 for 4 seconds followed by a motor PWM of 1500 after that.

   Let the vehicle come to a complete stop before ending each trial. Be sure to record your accelerometer or encoder data during each test (whichever method you prefer for estimating velocity).

2. Import your experimental data into Matlab. Using the `fminunc` or `lsqcurvefit` MATLAB functions on eqn. (4), determine the best-fit value for β. (NOTE: this is exactly the same code you wrote for your Homework 5 coast down test analysis). Use the following parameters in your estimation:

   - $\rho = 1.225$ kg/m^3 is the mass density of air,
   - Your BARC's mass (m) is 5 kg,
   - $g = 9.81$ m/s^2.
   - A_F is frontal area of the BARC vehicle. Use the empirical formula provided by the NSCEP: $A_f =$ height $\cdot$ width $\cdot\ 0.9$ m^2. Approximate the height as 0.15m and the width as 0.25m.
   - V_0 m/s is the vehicle's longitudinal velocity at the point in time you cut the acceleration signal. (4 seconds in our demo vehicle).

3. ***Deliverable**: Create and submit a plot of $\frac{t}{T}$ vs. $\frac{v(t)}{V_0}$, where T is the time when your vehicle comes to a complete rest, and V_0 is the vehicle's initial longitudinal velocity at the point in time you cut the acceleration signal. Plot your raw experimental data and line of best fit on the same plot, using normalized axes. Please include your resulting β value in the figure's title, legend or caption.

4. ***Deliverable**: Report the resulting C_d and R_x. You should have a very high C_d. Explain why the C_d is much higher than what one expects with a real, full-scale vehicle.

Task 3 Controller design

LQR aims to minimize a sum of quadratic costs along a time trajectory of a system $\dot{x} = Ax + Bu$:

$$J = \int_{t=0}^{\infty} \left(x(t)^T Q x(t) + u(t)^T R u(t)\right),$$

where Q penalizes deviations in state from the origin, and R penalizes large actuation inputs. The solution to the optimization problem above is

$$u(t) = -K \cdot x(t)$$

where the optimal feedback gain K is obtained in Matlab as `K=lqr(A,B,Q,R)`.

In reference tracking LQR, the controller works to minimize the gap between the true state x and the desired state x_d, so that $\lim_{t\to\infty} x(t) - x_d(t) = 0$. In this case the cost becomes

$$J = \int_{t=0}^{\infty} \left(e(t)^T Q e(t) + \delta u(t)^T R \delta u(t)\right),$$

where $e(t) = x(t) - x_d$, $\delta u(t) = u(t) - u_d(t)$, and (x_d, u_d) is an equilibrium point of the system. In this lab, the state is $x(t) = v(t)$, and the input is $u(t) = F_x(t)$.

1. ***Deliverable**: Compute the equilibrium point $(v_d, F_{x,d})$ of system in Eqn. 1 by applying the definition of an equilibrium point:

$$0 = F_{x,d} - R_x - mg \sin(\theta) - \frac{1}{2}\rho C_d A_F (v_d)^2 \tag{5}$$

where v_d is the desired speed (user chosen) and $F_{x,d}$ is the corresponding traction force input. (we set $V_{wind} = 0$). Solve Eq. 5 for $F_{x,d}$ as a function of v_d and θ to obtain

$$F_{x,d} = f_{eq}(v_d, \theta), \tag{6}$$

Note that in order to compute the equilibrium input $F_{x,d}$ we need the incline θ (which we estimate from our pitch sensor measurement) and the desired vehicle speed v_d.

2. ***Deliverable**: Linearize Eqn. 1 around $(v_d, F_{x,d})$ to obtain:

$$\dot{e} = A(v_d)e + B\delta u \tag{7}$$

where $e(t) = v(t) - v_d$, $\delta u(t) = F_x(t) - F_{x,d}$. Report your A and B matrices.

3. ***Deliverable**: Compute and report the LQR gain for the above system in Eqn. 7 with a desired reference of $v_d = 0.5$ m/s, with $Q = R = 1$.

4. **USEFUL NOTE.** Since the LQR gain has been computed for the error model in Eqn. 7. the controller is $\delta u(t) = -K \cdot e(t)$ which can be rewritten as:

$$F_x(t) = -K \cdot (v(t) - v_d) + F_{x,d} \tag{8}$$

Since K depends on A which depends on v_d, this control law is known as a "gain scheduled linear controller". In addition this controller has also a feedforward term $F_{x,d}$.

Task 4 Simulink Simulations

Test and tune your LQR controller by running the designed cruise control in Simulink. Your objective will be to track a piece-wise constant reference velocity.

1. ***Deliverable**: The LQR_Blank.slx Simulink model contains the following blocks:
    - An LQR controller, which you need to fill in
    - A vehicle model, which you need to fill in
    - Road incline angle profile:
        - (0) degrees from time $t = 0:3s$
        - (10) degrees from time $t = 3:8s$
        - (20) degrees from time $t = 8:13s$
        - (-15) degrees from time $t = 13:17s$

   - (0) degrees from time $t = 17 : 25s$

- Sensor noise, that can be switched on or off using a manual switch. The default setting is off (no sensor noise).
- A time-varying reference velocity:
   - 0.5 m/s from time $t = 0 : 10$s
   - 1 m/s from time $t = 10 : 25$ s
- A scope block, which plots the vehicle's velocity against the reference velocity.

You need to complete the Simulink model by filling in the "LQR Controller" and "Vehicle Model" blocks appropriately.

Hints:

- Use the `lqr` command to calculate the optimal feedback gain K.
- When you tune your LQR controller, start with $Q = R = 1$, and make changes from there. Remember that Q penalizes deviations from the reference state, and R penalizes high inputs.

Please submit your completed Simulink model.

2. Run simulations of your system. Start with $Q = R = 1$, and adjust until your controller tracks the reference signal.
3. ***__Deliverable__**: Submit a plot of your simulated vehicle's actual speed compared with the provided reference velocity signal. (Your scope's plot is fine).
4. ***__Deliverable__**: Switch your sensor noise on by double-clicking on the manual switch block in the noise subsystem. In the presence of noise, you will have to retune your LQR gains Q and R to get good performance. Once you have found appropriate gains, submit a plot of the vehicle's actual speed compared with the reference speed on the same graph.
5. ***__Deliverable__**: Explain why the controller is able to track the reference velocity perfectly without sensor noise, even with large disturbances. (Note that our controller does not have an integral action). You can use simulation to validate your claims.

You do not need to implement this LQR on your BARC car. If you are struggling with the BARC, at the end of the semester you can choose to implement this LQR on the BARC as your final project.

If you have time and want to test this on your car, please add plots of your experimental tests! (Note that there are a few additional steps to do for the BARC implementation since we do not control directly F_x).