EN530.603 Applied Optimal Control
Homework #4
October 4, 2021

Due: October 11, 2021 (before class)

Prof: Marin Kobilarov

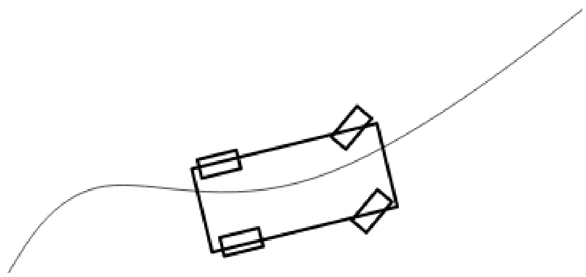

Figure 1: A car-like robot tracking a trajectory

1. Given the first-order system with quadratic criterion

$$\dot{x} = ax - bu, \quad x(t_0) \quad \text{given}, \tag{1}$$

$$J = \frac{1}{2}c[x(t_f)]^2 + \frac{1}{2}\int_{t_0}^{t_f}[u(t)]^2 dt, \tag{2}$$

where x, u and scalar variables and a, b, c are constant. Compute analytically the optimal control $u(t)$ minimizing J using the optimal control Euler-Lagrange equations and transversality conditions.

2. Given the first-order system with quadratic criterion

$$\dot{x} = ax - bu, \quad x(t_0) \quad \text{given}, \tag{3}$$

$$J = \frac{1}{2}c[x(t_f)]^2 + \frac{1}{2}\int_{t_0}^{t_f}[u(t)]^2 dt, \tag{4}$$

where x, u and scalar variables and a, b, c are constant, similarly to Problem 1. Notice that it has linear dynamics and a quadratic cost. This time solve for the optimal control $u(t)$ that minimizes J using the Ricatti equation, rather than directly from the necessary conditions given by variational calculus.

3. Consider the second-order system

$$\dot{x}_1 = x_2, \tag{5}$$

$$\dot{x}_2 = 2x_1 - x_2 + u, \tag{6}$$

$$\tag{7}$$

with cost function

$$J = \int_0^{t_f} [x_1^2 + \frac{1}{2}x_2^2 + \frac{1}{2}u^2]dt, \tag{8}$$

Find the optimal control law by finding the Riccati ODE. Implement the control law from initial condition $x(0) = [-5,\ 5]^T$ until final time $t_f = 20$ using Matlab (you can either integrate $P(t)$ analytically or numerically backwards in time using e.g. `ode45`, whichever is applicable). Plot the resulting elements of the matrix $P(t)$, the control input $u(t)$ and state histories $x(t)$.

4. Consider a trajectory tracking problem for an autonomous car (see Figure 1). The simplified dynamics for a car are given by

$$\dot{x} = f(x,u) = \begin{pmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{\theta} \\ \dot{v} \\ \dot{\delta} \end{pmatrix} = \begin{pmatrix} v\cos\theta \\ v\sin\theta \\ v\tan\delta \\ u_1 \\ u_2 \end{pmatrix}, \tag{9}$$

where (p_1, p_2) is the planar position of the car, v is the forward velocity, θ is the angle of the car with respect to the p_1-axis, and δ is the steering angle. The control inputs are the acceleration u_1 and the steering angle rate u_2. Consider a desired reference trajectory defined by

$$x_d(t) = \begin{pmatrix} t \\ 2t \\ \arctan 2 \\ \sqrt{5} \\ 0 \end{pmatrix} \tag{10}$$

with desired controls

$$u_d(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{11}$$

The trajectory tracking error is then defined as

$$e(t) = x(t) - x_d(t) \tag{12}$$

with virtual control signal

$$s(t) = u(t) - u_d(t). \tag{13}$$

Given a standard quadratic cost function

$$J = \int_0^{\infty} \left[\frac{1}{2}e^\top Q e + \frac{1}{2}s^\top R s \right] dt, \tag{14}$$

find the locally optimal LQR control law for tracking the desired trajectory. Note: the error dynamics are non-linear, so we cannot simply use the Ricatti equation to solve for a closed-form control law. We can, however, linearize the dynamics about the desired trajectory and formulate an LQR controller that is valid in a local region around the trajectory.

(a) Write the error dynamics $\dot{e}(t)$ for this trajectory tracking problem.

(b) Linearize $\dot{e}(t)$ so that it is in the form $\dot{e} \approx Ae + Bs$, with $A = f_x \in \mathbb{R}^{5\times5}, B = f_u \in \mathbb{R}^{5\times2}$. Write A and B explicitly.

(c) Implement the locally optimal control law for the error dynamics linearized about the reference trajectory using Matlab (feel free to use the `lqr` function here). Simulate a car tracking the given reference trajectory using the implemented control law. Start at state $x(0) = (0, 0, 0, 0, 0)^\top$ and simulate the system for 5 seconds. Use $Q =$ `diag([5, 5, 0.01, 0.1, 0.1])` and $R =$ `diag([0.5, 0.1])`.

(d) Plot the resulting control input $u(t)$. Plot the position history $p_1(t), p_2(t)$ along with the reference trajectory.

Note: upload your code as a single zip file (please name it as *LastName_FirstName_HW4.zip*) to the File upload link on the class webpage; in addition attach a printout of the code and plots to your homework solutions.