Spring 2022

Stanford
AA 203: Optimal and Learning-based Control
Homework #1, due April 18 by 11:59 pm

Problem 1: Backstepping

Consider the strict-feedback system

$$\begin{aligned}\dot{x} &= f(x) + B(x)z \\ \dot{z} &= u\end{aligned},$$

with $x \in \mathbb{R}^n$ and $z, u \in \mathbb{R}^m$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $B : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are known smooth functions, and $f(0) = 0$.

Suppose the subsystem $\dot{x} = f(x) + B(x)z$ can be stabilized by a smooth feedback law $z = \phi_0(x)$ with $\phi_0(0) = 0$, i.e., the closed-loop system $\dot{x} = f(x) + B(x)\phi_0(x)$ is *globally asymptotically stable* with respect to the origin $x = 0$. Moreover, suppose we know a smooth, positive-definite, radially unbounded Lyapunov function $V_0 : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and positive definite function $\rho : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ satisfying

$$\dot{V}_0(x) = \nabla V_0(x)^\mathsf{T}(f(x) + B(x)\phi_0(x)) \leq -\rho(x),$$

for all $x \in \mathbb{R}^n$.

We now consider the entire (x, z)-system, which we can only control through $u \in \mathbb{R}^m$. We want to use our knowledge of a stabilizing controller for the inner x-dynamics and the strict-feedback form of the (x, z)-dynamics to "back out" a stabilizing controller for the entire system.

Use the Lyapunov candidate function

$$V_1(x, z) = V_0(x) + \frac{1}{2}\|z - \phi_0(x)\|_2^2$$

to find a stabilizing controller $u = \phi_1(x, z)$ for some function $\phi_1 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ that ensures $(x, z) \to (0, 0)$. Notice that V_1 comprises the "inner" Lyapunov function V_0 and a penalty term for the difference between z and the value of the "inner" stabilizing control. Explicitly derive the function ϕ_1 and rigorously describe why it stabilizes the (x, z)-system using Lyapunov theory (i.e., prove $V_1(x, z)$ is positive-definite and radially unbounded, and $\dot{V}(x, z)$ is negative-definite along trajectories of the (x, z)-subsystem in closed-loop with $u = \phi_1(x, z)$).

Problem 2: Model reference adaptive control

Consider the continuous-time system

$$\dot{y}(t) + \alpha y(t) = \beta u(t).$$

We want to control this system, but we do not know the true plant parameters $\alpha, \beta \in \mathbb{R}$. In this problem, we will use *direct Model-Reference Adaptive Control (MRAC)* to match the behaviour of the true plant with that of the reference model

$$\dot{y}_m(t) + \alpha_m y_m(t) = \beta_m r(t)$$

where $\alpha_m, \beta_m \in \mathbb{R}$ are *known* constant parameters, and $r(t)$ is a chosen *bounded* reference signal.

(a) Consider the control law

$$u(t) = k_r(t)r(t) + k_y(t)y(t)$$

where $k_r(t)$ and $k_y(t)$ are time-varying feedback gains. Write out the differential equation for the resulting closed-loop dynamics. Use this to verify that, if $y(0) = y_m(0)$ and we knew α and β, the constant control gains

$$k_r^* := \frac{\beta_m}{\beta}, \quad k_y^* := \frac{\alpha - \alpha_m}{\beta}$$

would make the true plant dynamics perfectly match the reference model.

(b) When we do not know α and β, we adaptively update our controller over time in response to measurements of $y(t)$. Specifically, we want an *adaptation law* for $k_r(t)$ and $k_y(t)$ to make $y(t)$ tend towards $y_m(t)$ asymptotically. For this, we define the tracking error $e(t) := y(t) - y_m(t)$ and the parameter errors

$$\delta_r(t) := k_r(t) - k_r^*, \quad \delta_y(t) := k_y(t) - k_y^*.$$

Determine a differential equation for e in terms of e, $\dot{e}$, y, r, δ_y, δ_r, and suitable constants.

We consider the adaptation law for k_r and k_y described by

$$\begin{aligned} \dot{k}_r(t) &= -\operatorname{sign}(\beta)\gamma e(t)r(t) \\ \dot{k}_y(t) &= -\operatorname{sign}(\beta)\gamma e(t)y(t) \end{aligned},$$

where $\gamma > 0$ is a chosen constant *adaptation gain*. Since we are adapting the gains k_r and k_y of our controller directly, rather than estimates of the system parameters α and β, this is a *direct* adaptation law. We must at least know the sign of β, which indicates in what direction the input $u(t)$ "pushes" the output $y(t)$. For example, when modeling a car, you could reasonably assume that an increased braking force slows down the car. To show that the tracking error and parameter errors are stabilized by our chosen control law and adaptation law, we use Lyapunov theory.

Theorem 1 (Lyapunov)**.** *Consider the continuous-time system $\dot{x} = f(x,t)$, where $x = 0$ is an equilibrium point, i.e., $f(0,t) \equiv 0$. Suppose there exists a continuously differentiable scalar function $V(x,t)$ such that V is positive-definite in x for each $t \geq 0$, and $\dot{V}$ is negative semi-definite in x for each $t \geq 0$. Then $x = 0$ is a stable point in the sense of Lyapunov, i.e., $\|x(t)\|_2$ remains bounded as long as $\|x(0)\|_2$ is bounded.*

(c) Consider the state $x := (e, \delta_r, \delta_y)$ and the Lyapunov function candidate

$$V(x) = \frac{1}{2}e^2 + \frac{|\beta|}{2\gamma}(\delta_r^2 + \delta_y^2).$$

Show that $\dot{V} = -\alpha_m e^2$. Based on Lyapunov theory, what can you say about $e(t)$, $\delta_r(t)$, and $\delta_y(t)$ for all $t \geq 0$ if $\alpha_m > 0$?

In general, adaptive controllers yield time-varying closed-loop dynamics, even for LTI systems. As a result, we require more mathematical machinery beyond basic Lyapunov theory to establish anything stronger than Lyapunov stability. To this end, we use Barbalat's Lemma.

Theorem 2 (Barbalat's Lemma)**.** *Suppose $g : \mathbb{R} \to \mathbb{R}$ is differentiable. If g has a finite limit as $t \to \infty$ and $\dot{g}$ is uniformly continuous, then* $\lim_{t\to\infty} \dot{g}(t) = 0$.

Boundedness of the derivative of a function is a sufficient condition for Lipschitz continuity and hence uniform continuity. As a result, we have the following corollary.

Corollary 1. *Suppose $g : \mathbb{R} \to \mathbb{R}$ is twice-differentiable. If g has a finite limit as $t \to \infty$ and $\ddot{g}$ is bounded, then* $\lim_{t\to\infty} \dot{g}(t) = 0$.

(d) Apply Barbalat's Lemma to V to prove a stronger statement about $e(t)$ than we could originally make with basic Lyapunov theory in part (c).

With the given control law and adaptation law, MRAC proceeds as follows. We choose a reference signal $r(t)$ to excite the reference output $y_m(t)$ and construct the input signal $u(t)$. We use $u(t)$ to excite the true model, from which the output $y(t)$ and tracking error $e(t)$ are observed. The output $y(t)$ is fed back into the control law, while the tracking error $e(t)$ is fed into the adaptation law.

(e) Apply MRAC to the unstable plant

$$\dot{y}(t) - y(t) = 3u(t).$$

That is, simulate an adaptive controller for this system that does not have access to the true model parameters $\alpha = -1$ and $\beta = 3$. The desired reference model is

$$\dot{y}_m(t) + 4y_m(t) = 4r(t),$$

with $\alpha_m = 4$ and $\beta_m = 4$. Use an adaptation gain of $\gamma = 2$, and zero initial conditions for y, y_m, k_r, and k_y. For $t \in [0, 10]$, plot both $y(t)$ and $y_m(t)$ in one figure, and $k_r(t)$, k_r^*, $k_y(t)$, and k_y^* in another figure for $r(t) \equiv 4$. Then repeat this for $r(t) = 4\sin(3t)$. Overall, you should have four figures in total. What do you notice about the trends for different reference signals? Why do you think this occurs? In your explanation, try to link your observations with the statements about $e(t)$, $\delta_r(t)$, and $\delta_y(t)$ we were able and *unable* to prove in parts (c,d).

Problem 3: Extremal curves

Given the functional

$$J(x) = \int_0^1 \left(\frac{1}{2}\dot{x}(t)^2 + 5x(t)\dot{x}(t) + x(t)^2 + 5x(t) \right) dt,$$

find an extremal curve $x^* : [0, 1] \to \mathbb{R}$ that satisfies $x^*(0) = 1$ and $x^*(1) = 3$.

Problem 4: Dubins car

The kinematics of the Dubins car are described by

$$\begin{aligned} \dot{x} &= v \cos\theta \\ \dot{y} &= v \sin\theta \\ \dot{\theta} &= u \end{aligned} \quad ,$$

where $(x, y) \in \mathbb{R}^2$ is the car's position, $\theta \in \mathbb{R}$ is the car's heading, $v > 0$ is the car's constant known speed, and u is the controlled turn rate. The turn rate is bounded, i.e., $u \in [-\bar{\omega}, \bar{\omega}]$, where $\bar{\omega} > 0$ is a known constant.

The car starts at $(x, y) = (0, 0)$ with a heading of $\theta = 0$ at $t = 0$. We want the car to drive to $(x, y) = (0, c)$ in the least amount of time possible, where $c > 0$ is a given constant.

(a) Use Pontryagin's minimum principle to express the optimal control input $u^*(t)$ as a function of the optimal co-state $p^*(t) := (p_x^*(t), p_y^*(t), p_\theta^*(t)) \in \mathbb{R}^3$.

Hint: You should discover that the minimum condition for $u^*(t)$ is not informative whenever $p_\theta^*(t) \equiv \bar{p}_\theta$ for a particular fixed value $\bar{p}_\theta \in \mathbb{R}$. When such a lack of information persists over a non-trivial time interval, i.e., any time interval $[t_1, t_2]$ with $t_2 > t_1 \geq 0$, this is known as a *singular arc*. To compute $u^*(t)$ in this case, use the fact that $p_\theta^*(t) \equiv \bar{p}_\theta$ is constant in time along such an arc for this particular problem.

(b) Argue why $p^*(t)$ might end in a singular arc depending on the boundary conditions. Suppose we know $p^*(t)$ begins on a non-singular arc, then switches once to and ends on a singular arc. For this particular case, argue why $u^*(0) = \bar{\omega}$ and describe the optimal state trajectory $(x^*(t), y^*(t), \theta^*(t))$ and control trajectory $u^*(t)$ in words without explicitly deriving them.

Problem 5: Single shooting for a unicycle

Consider the kinematic model of a unicycle

$$\begin{aligned}
\dot{x}(t) &= v(t)\cos(\theta(t)), \\
\dot{y}(t) &= v(t)\sin(\theta(t)), \\
\dot{\theta}(t) &= \omega(t).
\end{aligned}$$

with state $\mathbf{x} = (x, y, \theta)$ and control $\mathbf{u} = (v, \omega)$. Suppose the objective is to drive from a starting configuration to a target configuration with minimum time and control effort; specifically we want to minimize the functional

$$J = \int_0^{t_f} \left(\lambda + v(t)^2 + \omega(t)^2\right) dt,$$

where $\lambda > 0$ is a weighting factor and t_f is free, subject to the initial and terminal conditions

$$\begin{bmatrix} x(0) \\ y(0) \\ \theta(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \pi/2 \end{bmatrix}, \quad \begin{bmatrix} x(t_f) \\ y(t_f) \\ \theta(t_f) \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ \pi/2 \end{bmatrix}.$$

(a) Derive the Hamiltonian and necessary conditions for optimality, specifically (i) the ODE the state and costate must satisfy, (ii) the optimal control as a function of the state and costate, and (iii) the boundary conditions including the relevant transversality condition for free t_f.

In practice, you might use a boundary value problem solver (e.g., `scipy.integrate.solve_bvp`), but in this problem we'll use a bit of nonlinear optimization theory to write our own!

(a) In the file `p5_unicycle_single_shooting.py`, complete the implementations of `dynamics`, `hamiltonian`, `optimal_control`, and `shooting_ode`.

Recall from class that the simplest shooting method[1] basically boils down to guessing the correct initial costate and propagating the dynamics and costate equations until the appropriate final time (also a "guess" in free-final-time problems) at which the terminal boundary conditions are satisfied.

(a) Use the ODE integration performed by `state_and_costate_trajectories` to implement `shooting_residual`, a measure of how far off each of your terminal boundary conditions is from satisfaction, given guesses for the initial costate and final time.

[1]Single shooting as opposed to, e.g., multiple shooting.

(b) Finally, in `newton_step` and `shooting_method`, implement Newton's method for finding roots of `shooting_residual`. Now, if you provide an appropriate guess for the initial costate and final time, you can run `python3 p5_unicycle_single_shooting.py` and see a plot of the optimal solution. You may find that whether or not your BVP solver converges to a solution is highly dependent on the quality of your initial guess – indeed, initialization is a major challenge when applying indirect methods for optimal control!

Hint: Recall that for finding zeros of a function $f : \mathbb{R}^n \to \mathbb{R}^n$, each iteration of Newton's method entails improving a current best guess x_k using the formula

$$x_{k+1} = x_k - \nabla f(x_k)^{-1} f(x_k),$$

where ∇f denotes the Jacobian of f.

Learning goals for this problem set:

Problem 1: Learn how to construct stabilizing controllers for complicated systems by exploiting structure in the dynamics.

Problem 2: Explore the theoretical underpinnings of MRAC, and observe its behaviour on an example system in simulation.

Problem 3: Become familiar with the process of solving calculus of variations problems.

Problem 4: Learn how singular arcs can complicate analyses that use Pontryagin's minimum principle.

Problem 5: Implement an indirect method for optimal control and gain familiarity with JAX.