Spring 2019

Stanford
AA 203: Introduction to Optimal Control and Dynamic Optimization
Problem set 8, due on June 5

Please remember to attach your code to your problem set.

Problem 1: Consider the discrete-time system

$$x(t+1) = ax(t) + bx(t-1) + w(t)$$

This *autoregressive (AR) model* represents a stochastic process where the next state depends linearly on previous states and some process noise $w(t)$.

a) Suppose we have measurements $\{x(t)\}_{t=0}^{N}$, but we do not know the true parameter values (a, b). Formulate the linear least-squares problem for determining an estimate of (a, b); i.e., identify the regressors and all vectors that appear in the least-squares objective for this data set.

b) Generate such data sets for $N \in \{10, 100, 1000\}$ by simulating the system above with $(a, b) = (1, -0.1)$, $x(0) = 1$, $x(1) = 0.5$, and $w(t) \sim \mathcal{N}(0, 1)$. For each different data set size N, compute the least-squares estimate of (a, b) averaged over 100 trials. Report the mean and standard deviation of your estimates for each N, and briefly comment on any variation.

c) Repeat part (b) with $w(0) \sim \mathcal{N}(0, 1)$ and $w(t) \sim \mathcal{N}(0, 2\,|w(t-1)|)$ for $t > 0$. How do the mean and standard deviation of your parameter estimates compare to part (b)? Why do you think this is the case?

d) Now consider the nonlinear system

$$x(t+1) = cx(t)x(t-1) + w(t)$$

with the unknown parameter c. The term $x(t)x(t-1)$ is nonlinear, yet why can you still apply the linear least-squares procedure to this system? Repeat parts (a) and (b) for this system. Use $c = 0.1$, $x(0) = 1$, $x(1) = 0.5$, and $w(t) \sim \mathcal{N}(0, 1)$.

Problem 2: Consider the continuous-time system

$$\dot{y}(t) + ay(t) = bu(t) \tag{1}$$

We want to control this system, but we do not know the true plant parameters (a, b). You will use *model-reference adaptive control (MRAC)* to match the behavior of the true plant with that of the reference model

$$\dot{y}_m(t) + a_m y_m(t) = b_m r(t)$$

where (a_m, b_m) are *known* constant parameters, and $r(t)$ is a chosen bounded external 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 we knew (a, b), the following constant control gains

$$k_r^* \coloneqq \frac{b_m}{b}$$
$$k_y^* \coloneqq \frac{a - a_m}{b}$$

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

b) When we do not know (a, b), we need to adaptively update our controller over time. 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 $e(t) \coloneqq y(t) - y_m(t)$ and the parameter errors

$$\delta_r(t) \coloneqq k_r(t) - k_r^*$$
$$\delta_y(t) \coloneqq k_y(t) - k_y^*$$

Determine the differential equation governing the dynamics of $e(t)$, in terms of e, any of its derivatives, y, r, δ_y, δ_r, and suitable constants.

We will consider the adaptation law for k_r and k_y described by

$$\dot{k}_r(t) = -\operatorname{sign}(b)\gamma e(t)r(t)$$
$$\dot{k}_y(t) = -\operatorname{sign}(b)\gamma e(t)y(t)$$

where $\gamma \in \mathbb{R}_{>0}$ is a chosen constant *adaptation gain*. For this adaptation law, we must at least know the sign of b, which indicates in what direction the input $u(t)$ "pushes" the output $y(t)$ in (1). For example, when modeling a car, you could reasonably assume that an increased braking force slows down the car.

To show that tracking error and parameter errors are stabilized by our chosen control law and adaptation law, we will use Lyapunov theory. While previously we applied Lyapunov theory to discrete-time systems in the context of MPC, we are now dealing with continuous-time systems.

Theorem 1 (Lyapunov): *Consider the continuous-time system $\dot{x} = f(x,t)$, where $x = 0$ is an equilibrium point, i.e., $f(0,t) \equiv 0$. If there exists a continuously differentiable scalar function $V(x,t)$ such that*

- *V is positive definite in x, and*
- *$\dot{V}$ is negative semi-definite in x,*

then $x = 0$ is a stable point in the sense of Lyapunov, i.e., $\|x(t)\|$ remains bounded as long as $\|x(0)\|$ is bounded.

c) Now, consider the state $x \coloneqq (e, \delta_r, \delta_y)$ and the Lyapunov function candidate

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

Show that $\dot{V} = -a_m e^2$. Based on Lyapunov theory, what can you say about $e(t)$, $\delta_r(t)$, and $\delta_y(t)$ for all $t \in [0, \infty)$ if $a_m > 0$? Furthermore, apply Barbalat's lemma to $\dot{V}$ to make a stronger statement about $e(t)$ than you originally did with just Lyapunov theory.

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

Hint: To prove that a function is uniformly continuous, it suffices to show that its derivative is bounded. Lipschitz continuity and thus uniform continuity follow from this.

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

d) 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 $(a, b) = (-1, 3)$. The desired reference model is

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

i.e., $(a_m, b_m) = (4, 4)$. Use an adaptation gain of $\gamma = 2$, and zero initial conditions for y, y_m, k_r, and k_y. Plot both $y(t)$ and $y_m(t)$ over time in one figure, and $k_r(t)$, k_r^*, $k_y(t)$, and k_y^* over time in another figure for $r(t) = 4$. Then repeat this for $r(t) = 4\sin(3t)$. That is, you should have four figures in total. What do you notice about the trends for different reference signals? Why do you think this occurs?

Hint: To do the simulation in MATLAB, form a system of ODEs for either (y, y_m, k_r, k_y) or $(y, e, \delta_r, \delta_y)$, then use `ode45()`.

Problem 3: In this problem we will consider the application of adaptive LQR to control the cooling of a data center. This problem has seen recent attention in reinforcement learning [1, 2], and a simple linear version of it was used as a benchmark of adaptive control/RL algorithms in [3].
We assume dynamics of the form

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + \boldsymbol{w}_k$$

in which $\boldsymbol{x}_k \in \mathbb{R}^3$ denotes the temperature of a collection of servers at timestep k, $\boldsymbol{u}_k \in \mathbb{R}^3$ denotes how much cooling effort is applied to each server, and $\boldsymbol{w}_k \in \mathbb{R}^3$ is a stochastic disturbance induced by randomly varying server utilization. Each server has a nominal cooling system that is designed to slowly drop the temperature level of the server toward a nominal operating point; as such, our initial estimate of the system dynamics are

$$\hat{A}_0 = \begin{bmatrix} 0.99 & 0 & 0 \\ 0 & 0.99 & 0 \\ 0 & 0 & 0.99 \end{bmatrix}, \quad \hat{B}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In reality, the close positioning of the servers results in the true dynamics

$$A = \begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix}, \quad B = \hat{B}_0,$$

with noise $\boldsymbol{w}_k \sim \mathcal{N}(0, \sigma^2 I)$, where $\sigma = 0.01$. We will specify cost function $Q = I$ and $R = 1000I$. Use $x_0 = [1.0, 1.5, 2.0]^T$ for all problems below.

a) First, design an LQR controller based on $\hat{A}_0, \hat{B}_0$, and simulate this controller with the dynamics A, B (including the stochastic disturbance). Simulate this system for 100 time steps, 1000 time steps, and 10000 time steps. Plot the state vs. time and control actions vs. time, and report the total cost achieved for each of the three simulation lengths.

b) Next, implement a certainty-equivalent adaptive LQR controller: at each timestep, compute via recursive least squares an updated system estimate of the system, $\hat{A}_k, \hat{B}_k$ (starting with our initial estimate $\hat{A}_0, \hat{B}_0$). A convenient form for this update is as follows. Let $L_0 = I_{6\times 6}$ and $Q_0 = [\hat{A}_0^T, \hat{B}_0^T]^T$. Furthermore, let $\hat{\boldsymbol{x}}_k = [\boldsymbol{x}_k^T, \boldsymbol{u}_k^T]^T$. Then, the update can be recursively computed as

$$L_{k+1} = L_k - \left(\frac{1}{1 + \hat{\boldsymbol{x}}_k^T L_k \hat{\boldsymbol{x}}_k}\right)(L_k\hat{\boldsymbol{x}}_k)(L_k\hat{\boldsymbol{x}}_k)^T$$
$$Q_{k+1} = \hat{\boldsymbol{x}}_k \boldsymbol{x}_{k+1}^T + Q_k$$

and

$$[\hat{A}_k, \hat{B}_k] = (L_k Q_k)^T.$$

Then, re-optimize the controller with this new system model. For problem lengths of 100, 1000, and 10000, plot the state vs. time, actions vs. time, as well as $\|\hat{A}_k - A\|_F$ and $\|\hat{B}_k - B\|_F$ (where $\|\cdot\|_F$ denotes the Frobenius norm) vs. time. Report the total cost achieved for each problem length. *Hint: warm-start the Riccati recursion by initializing with the value matrix of the previous iteration to improve numerical efficiency.*

c) Try adding white noise to the control action to improve system identification. In particular, replace your previous control actions $\boldsymbol{u}_k$ with $\boldsymbol{u}_k + \epsilon_k$, where $\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2 I)$. Try a few values of σ_ϵ, and report the cost for each. For problem lengths of 100, 1000, and 10000, plot the state vs. time, actions vs. time, $\|\hat{A}_k - A\|_F$ vs. time, and $\|\hat{B}_k - B\|_F$ vs. time. What do you notice about the estimate of the system dynamics, relative to the certainty-equivalent approach? What do you notice about the cost?

d) Look at the literature on the adaptive LQR/RL for LQR problem. Implement an exploration scheme from a paper in this body of work. Write a paragraph on how the exploration scheme works, and your observations on the performance of the method. Plot the same quantities as in b) and c), and report the achieved cost. *Hint: a good literature review of learning for the LQ problem is available in [4].*

Learning goals for this problem set:

Problem 1: To understand the utility and limitations of linear least-squares regression for system identification.

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

Problem 3: To gain practical experience with adaptive LQR, understand the trade-offs that exist in different exploration methods, and become familiar with the literature on the problem.

References

[1] N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu and G. Imwalle. Data center cooling using model-predictive control. Neural Information Processing Systems (NeurIPS), 2018.

[2] J. Gao and R. Jamidar. Machine learning applications for data center optimization. Technical report, Google White Paper, 2014

[3] S. Dean, H. Mania, N. Matni, B. Recht and S. Tu. On the sample complexity of the linear quadratic regulator. arXiv:1710.01688, 2017.

[4] Y. Abbasi-Yadkori and C. Szepesvari, . Regret bounds for the adaptive control of linear quadratic systems. Conference on Learning Theory, 2011.