Lab 6bc: Luenberger Observer and LQR Controller Design for Inverted Pendulum

1 Objectives

The objectives of this lab are to design a full-state observer to estimate the state and a feedback controller using the Linear Quadratic Refgulator (LQR) design techninque. We will utilize the observer for full state feedback control of the system to get an estimate of the full state while only measuring position of the cart and pendulum. Using the estimated full state, we will apply an LQR controller in which we will vary the penatly matrices P and Q in the cost function to observe performance effects.

2 Theory

2.1 The Luenberger Observer

Pole placement design is performed under the assumption that measurements of all states of the system are available. However, in many physical systems not all states may be easily measurable and thus states need to be estimated based on the limited sensing available. In this case the state feedback becomes $u = -K\hat{x}$, where $\hat{x}$ is the estimated state. We cannot use the controller $u = -Kx$, because the only measurements we have available are y.

Recall from class the dynamics of a Luenberger observer:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \tag{1}$$

where $y = Cx$ and $\hat{y} = C\hat{x}$. The first two terms in the above equation, $A\hat{x} + Bu$, can be called the predictor part and is a replica of the plant dynamics. However, because of uncertainties or errors in the plant model, the estimate of the state using only the predictor ("open-loop") will generally not match the actual state of the system. The corrective term $L(y - \hat{y})$ is thus needed. Together, these form the Luenberger observer.

The $L(y - \hat{y})$ term corrects future estimates of the state based on the present error in estimation. The gain matrix L can be considered a parameter which weighs the relative importance between the predictor and the corrector in state estimation. Intuitively, a "low" value for L is chosen when our confidence in the model (i.e. the predictor) is high and/or confidence in measurement y is low (i.e. when the measurements are noisy) and vice-versa for a "high" value of L.

The objective of this lab is to design the observer gain matrix L and use the state estimator for feedback control of the inverted-pendulum system instead of our previous derivative-based approximation.

2.2 The Linear Quadratic Regulator

Pole placement for controller design relies on specification of the desired closed-loop poles of the system. This is usually difficult to specify, especially for systems with a large number of states. Furthermore, with pole placement design there is hard to take the "amount" of actuation (called actuation or control effort) that gets used during closed-loop operation into account.

Good regulation of the system can usually be achieved by using high amount of actuation (for example in a P-controller, higher K_p, and thus greater actuation effort, gives faster rise time). But in reality, we are often limited by power and energy constraints. Ideally, we would like to achieve good system performance while at the same time minimizing the amount of actuation used in achieving the desired performance. One way of expressing this mathematically is through a cost functional of the form:

$$J = \int_0^\infty x^T Q x + u^T R u \, dt \tag{2}$$

where Q and R are weighting matrices (these are the design parameters).

The LQR design problem is to design a state-feedback controller K (i.e. for $u = -Kx$) such that the cost functional J is minimized[1]. The cost functional (3) consists of two terms, the first of which you can think of as being the cost of regulating the state x (regulatory term) and the second being the cost of actuation u (actuation term). Both of these terms depend on a weighting matrix, Q and R, respectively. These matrices are the design parameters, assumed positive semidefinite. The regulatory term will "penalize" deviations from the desired state (here $x = 0$), while the actuation term will "penalize" you for any actuation effort $u \neq 0$.

For simplicity we assume in this lab that the matrices Q and R are diagonal: $Q = \text{diag}(q_1, \ldots, q_n)$ and $R = \text{diag}(r_1, \ldots, r_m)$. Thus, the objective J reduces to

$$J = \int_0^\infty \left(\sum_{i=1}^{n} q_i x_i^2 + \sum_{j=1}^{m} r_j u_j^2 \right) dt \tag{3}$$

The scalars $q_1, \ldots, q_n$ and $r_1, \ldots, r_m$ can be seen as relative weights between different performance terms in the objective J. For Q and R to be positive semidefinite, we need $q_i \geq 0$ and $r_i \geq 0$ for all i. The key design problem of LQR is to translate performance specifications in terms of the rise time, overshoot, bandwidth, etc. into relative weights of the above form. There is no straightforward way of doing this and it is usually done through an iterative process either in simulations or on an experimental setup. Once the matrices Q and R are completely specified, the controller gain K is found by solving the so-called Algebraic Riccati Equation (ARE), which can be done numerically in MATLAB.

3 Pre-Lab

3.1 Controllability and Observability

Consider the linearized open-loop system from last week's lab, in state-space form. Check whether the system is controllable and / or observable. You can use the Matlab commands `ctrb`, `obsv` and `rank`.

3.2 Observer Design

Recall our state-feedback control from Lab 6a. Although we only measure the position x and angle θ, we assumed that we have access to the full state, and estimated $\dot{x}$ and $\dot{\theta}$ simply by using derivative blocks in Simulink. As we observed during the last lab, this yields a poor-quality estimate of $\dot{x}$ and $\dot{\theta}$ due to the amplification of noise. As a result, the controller output (actuation of the motor) was of poor quality, and this manifested in particular in a loud grinding noise at high frequencies.

In this lab we will solve these problems by implementing a Luenberger observer, which will provide a state estimate $\hat{x}$. We use this estimate for state feedback, i.e. $u(t) = K(r(t) - \hat{x}(t))$.

[1] In fact, one can show that even when optimizing over a larger class of controllers, it turns out that the optimal controller is a linear time-invariant state-feedback controller of the form $u = -Kx$.

Controller gain The model for inverted-pendulum system and the desired closed-loop poles $s_{1,2} = -2.0 \pm 10j$ and $s_{3,4} = -1.6 \pm 1.3j$ are the same as in the previous lab.

Observer gain The gain L is chosen such that the matrix $A - LC$ has eigenvalues in the left half-plane. Further, the exact position of the eigenvalues of $A - LC$ govern the rate at which the state estimate $\hat{x}$ converges to the actual state x of the system. It is desirable that the observer estimate of the state converges to the actual state much faster than the system dynamics. This helps the controller in obtaining a "good" estimate of the actual state of the system in relatively short time and thus it can take appropriate control action. A general rule of thumb is that the error dynamics should be at least an order of magnitude faster than the dynamics of the controlled system.

1. Given that the size of $A - LC$ must be the same as A, what are the dimensions of L?
2. For this lab, we want to place the eigenvalues of the observer at $-10 \pm 15j$ and $-12 \pm 17j$. Note that they have been chosen to be relatively far "away" from the desired closed-loop poles. Using MATLAB, find the matrix L such that this is achieved. How would you use the `place` command to do this? *Hint:* For any real square matrix M, the eigenvalues of M are the eigenvalues of its transpose M^T.

3.3 Simulation

1. Implement the designed observer in MATLAB. As usual, there should be *no* derivative blocks used. Remember that the observer is placed in feedback around the actual system. Use the estimate $\hat{x}$ of the state for state feedback. You can use the feedback gain matrix K designed in the previous lab, since the desired locations of the closed-loop poles have not changed.
2. Simulate the system with a 10 cm position perturbation and 5 degrees angle perturbation of the plant. You can achieve this by using an initial condition x_0 for the plant.
   Plot the observer estimate $\hat{x}$ of the state and the actual state of the plant x, which may be obtained from the plant model in Simulink (again, do *not* use derivative blocks), on 4 separate plots, one for each state variable. Note that in practice, this cannot be done with the physical plant, as we have no measurement of the actual state x (that's the whole point of the observer).
3. Plot the estimation error $e = \hat{x} - x$ and discuss how it varies with time.

3.4 LQR Design

Given the model for the inverted pendulum system has four states $x, \dot{x}, \theta, \dot{\theta}$ and one input, the motor voltage V, what will the dimensions of Q and R be?

This part of the Pre-Lab mainly consists of translating the stated performance specifications into matrices Q and R. For simplicity, we assume Q and R to be diagonal. The subscripts or the weights will denote which element on the diagonal the entry is (i.e. q_2 will be the weight on the diagonal for corresponding to the second state, $\dot{x}$).

Consider the following control objective: Given that the cart and the pendulum are $x_0 = 30$ cm and $\theta_0 = 0.05$ radians (≈ 2.5 deg) displaced from their desired positions $x_{des} = 0$ and $\theta_{des} = 0$ at time $t = 0$, the objective is to get the system to the desired state as soon as possible, but without using, say, more than 6 volts of the input at any point in time. For now, however, we will ignore the constraint on the input. For our problem, we set the scalars q_2 and q_4 to zero, as we have no inherent restriction on how $\dot{x}$ and $\dot{\theta}$ vary with time. Now, in order to use scalars q_1, q_3 and r as relative weights, we will normalize them based on their initial conditions. The modified weights are:

$$\overline{q}_1 = \frac{q_1}{0.3^2} \qquad \overline{q}_3 = \frac{q_3}{0.05^2} \qquad \overline{r} = \frac{r}{6^2}$$

The weights have been normalized with square terms because the integrand of our objective functional J is a quadratic function of x and u (so the matrix Q will use $\overline{q}_1$ and $\overline{q}_3$, and $R = \overline{r}$).

1. For nominal weights $q_1 = 1$, $q_3 = 1$, and $r = 1$ (giving equal weight to each term of the objective function), determine the gain matrix K which minimizes the objective function and its associated closed-loop pole locations. You may use the `lqr` command in MATLAB to do this. Simulate the closed-loop system including the observer from Lab 6b. That is, use the state estimate $\hat{x}$ to control the system – your input is $u = K(r - \hat{x})$. Make sure to use the same initial condition for observer and system. Report the value of your observer gain matrix L. Plot output y and control action u for initial conditions of $x_0 = 30\,\text{cm}$ and $\theta_0 = 0.05\,\text{rad}$.
2. Individually vary the weights from their nominal values and study the influence of the weights on how the system outputs and control effort varies with time. The weights are relative, so you may assume $q_1 = 1$ in all cases, and vary only the other two. Choose your weights such that you can clearly see the effect in the system behavior (you can restrict your weights to the range $0 - 100$). Consider the following five cases: (nominal, $q_3 \ll 1$, $q_3 \gg 1$, $r \ll 1$, and $r \gg 1$). For each case:
   (a) report the value of K and the closed-loop pole locations
   (b) plot the output y and the control action u
   (c) report the maximum deviations in x and θ as well as u_{max}, the maximal (absolute) value of u
   (d) describe briefly the effect of changing the weights on the closed-loop system behavior
3. You will observe that the position x will first increase before converging to zero. What is the physical reason for this behavior?

4 Lab

For the entire lab please use a fixed step solver with a time step of 0.001 seconds.

1. Implement the state feedback controller operating on the state estimate $\hat{x}$ provided by the Luenberger observer on the "hardware". For a zero reference signal, observe and record the output $\hat{y}$ of the observer and the actual measurement y when manually applying small perturbations. That is, plot both the estimated and actual signals on the same graph for the position of the cart and the pendulum. The difference between these two signals indicates how well the observer estimates the state of the system.
2. We will now compare the controller performance from Lab 6a with and without an observer. Remember that the closed loop poles of both systems are the same.
   For each of the following reference signals, *qualitatively* describe any noticeable differences in performance, and plot the cart position and the angular position of the rod for both controllers on top of each other and compare their tracking abilities.

- zero reference
- zero reference with small perturbations (try to be consistent in how you apply the perturbations)
- sinusoidal reference position with amplitude 5 cm and frequency 1 rad/s, i.e. $r_1(t) = .05\sin(t)$. The reference velocity, angle, and angular velocity should be set to 0.

3. Now we will look at the differences in performances a little more closely. Compare the estimates of the cart and pendulum velocities from this lab with the measurements obtained by taking the derivatives of the position and angle signals from the previous lab. How do these two schemes differ when a noise is present in the actual measurement of the positions?

4. Which scheme do you think gives the "better" performance, and more importantly, *why*? There is no definite answer here. Just form your own opinion and defend it.

5. Now replace the controller from lab 6a and implement the LQR controllers you designed in the Pre-Lab on the hardware (with the observer). Use a step input of the form $r = [0.3\ 0\ 0\ 0]^T$. Make sure to set the observer initial state to zero. For the weight matrices Q and R, consider:
   - nominal weights
   - a higher relative weight q_1, other weights nominal
   - a higher relative weight q_3, other weights nominal
   - a higher relative weight of r, other weights nominal

   In each case, observe the output response of the system, note the variation of the position of the cart and the pendulum with time and the control input. In addition, plot output y and control u and discuss the effect of the weights on the system behavior. Make sure that the differences are noticeable on your plots. For each case,
   - discuss how the closed-loop system behavior changes w.r.t. the nominal case
   - discuss if and how your results differ from the ones obtained in the simulations in the Pre-Lab

6. Run the sinusoidal reference from step 2 ($r_1(t) = .05\sin(t)$) using the weights that you think are best. How does the performance compare to step 2?