April 22, 2024

# 6.3100 Lecture 20 Notes – Spring 2024

## Integral linear quadratic regulator (LQR) control

Dennis Freeman, Elfar Adalsteinsson, and Kevin Chen

Outline:

1. Integral LQR control
2. External disturbance
3. Example: propellor arm
4. Motivation: sensor noise

### 1. **Integral LQR control**

From the last two lectures, we know the state-space representation has the following form:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

$$u(t) = K_r y_d - Kx$$

We can choose the control matrix K using pole placement or LQR methods. After we design K, we can use Kr to remove the steady state error:

$$K_r = \frac{-1}{C(A-BK)^{-1}B}$$

However, this design has a problem: if A, B, C, and D matrices have some modeling error, then there will exist a steady-state error. In some sense, modeling error is unavoidable. How do we remove the steady state error in the presence of modeling error?

In the first part of this course, we introduced PID control, where the integral term can remove the steady-state error even in presence of a modeling error. We will try to incorporate integral control into the state-space framework.

To start, we define a new state w. Note this w is a generic convention, it is not to be confused with the angular velocity state w. Here we have:

$$w(t) = \int_0^t (y(\tau) - y_d(\tau))d\tau$$

Then the derivative of w is given by:

$$\frac{d}{dt}w(t) = y(t) - y_d(t) = Cx(t) - y_d(t)$$

Note that if this state converges, that is if $\frac{d}{dt}w(t) = 0$, then the system error becomes 0. Let's try to modify our state-space problem by adding in this new state w. We can write:

$$\begin{aligned}\frac{d}{dt}x &= Ax + Bu\\ \frac{d}{dt}w &= Cx + 0u - y_d\end{aligned}$$

We write both equations into the matrix form:

$$\begin{bmatrix} I & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{w}\end{bmatrix} = \begin{bmatrix} A & 0\\ C & 0\end{bmatrix}\begin{bmatrix} x\\ w\end{bmatrix} + \begin{bmatrix} B\\ 0\end{bmatrix}u + \begin{bmatrix} 0\\ -1\end{bmatrix}y_d$$

$$y = \begin{bmatrix} C & 0\end{bmatrix}\begin{bmatrix} x\\ w\end{bmatrix}$$

Here we can define new matrices and states:

$$I_+ = \begin{bmatrix} I & 0\\ 0 & 1\end{bmatrix}, A_+ = \begin{bmatrix} A & 0\\ C & 0\end{bmatrix}, B_+ = \begin{bmatrix} B\\ 0\end{bmatrix}, C_+ = \begin{bmatrix} C & 0\end{bmatrix}, x_+ = \begin{bmatrix} x\\ w\end{bmatrix}$$

In the new notation, the equations become:

$$\begin{aligned} I_+\dot{x}_+ &= A_+x_+ + B_+u + \begin{bmatrix} 0\\ -1\end{bmatrix}y_d\\ y &= C_+x_+\end{aligned}$$

Given this new equation, how should we design the control input u? We simply let:

$$u = -K_+x_+$$

Note that in this design, $x_+ = \begin{bmatrix} x\\ w\end{bmatrix} = \begin{bmatrix} x\\ \int_0^t (Cx - y_d)d\tau\end{bmatrix}$. Since it has 1 more state, the feedback matrix $K_+$ should also have 1 more state. We have:

$$K_+ = [k_1, k_2, \dots k_{n+1}],$$

where $K_+$ has 1 more entry than the typical feedback matrix K.

The new control problem becomes:

$$\begin{aligned}\dot{x}_+ &= (A_+ - B_+K_+)x_+ + \begin{bmatrix} 0\\ -1\end{bmatrix}y_d\\ y &= C_+x_+\end{aligned}$$

We can define some closed loop matrices:

$$A_{clp} = A_+ - B_+K_+; B_{clp} = [0; -1]; and\ C_{clp} = C_+$$

To find the optimal matrix $K_+$, we can either use pole placement and lqr methods.

**Pole placement:**

$$K_+ = place(A_+, B_+, [desired\ poles])$$

**LQR:**

$$K_+ = lqr(A_+, B_+, Q, R)$$

2. **External disturbance**

We have discussed disturbance rejection in the early part of this course. When we implement PD or PID controllers, we can write a disturbance transfer function. Here, we are going to learn how state space controllers handle noise.

An unmodelled disturbance will affect the state variables x(t). The noise d(t) is mapped to changes of x(t) through a matrix F. The state space system is given by:

$$\dot{x}(t) = Ax(t) + Bu(t) + Fd(t)$$

$$y(t) = Cx(t) + Du(t)$$

Here F is a $n \times 1$ vector. Equivalently, the new system has two input variables: $u(t)$ and $d(t)$. Suppose we define a new input vector:

$$u'(t) = \begin{bmatrix} u(t) \\ d(t) \end{bmatrix}$$

The next control system will become:

$$\dot{x}(t) = Ax(t) + [B\ F]\, u'(t)$$

$$y(t) = Cx(t) + [D\ 0]\, u'(t)$$

To design our feedback matrix K', we let

$$u'(t) = \begin{bmatrix} K_r r \\ d \end{bmatrix} - K'x$$

Note that we cannot set the disturbance d(t). However, in this new formulation, the dimension of K' grows to $n \times 2$, which means we have more flexibility to choose the values of K' and reject noise.

You are going to experiment with different choices of K' in post lab 5. After you design a K' matrix, you can find the noise to output transfer function through using the transform techniques. We have:

$$\dot{x}(t) = Ax(t) + [B\ F]u'(t)$$

$$\dot{x}(t) = Ax(t) + [B\ F]\left(\begin{bmatrix} K_r r \\ d \end{bmatrix} - K'x\right)$$

$$sX = AX + [B\ F]\left(\begin{bmatrix} K_r R \\ D_{dist} \end{bmatrix} - K'X\right)$$

$$sI - (A - [B\ F] \times K')X = [B\ F]\begin{bmatrix} K_r R \\ D_{dist} \end{bmatrix}$$

$$X = \left( sI - (A' - [B\ F] \times K')\right)^{-1} FD_{dist}$$

Here we assume R(s) = 0 because when writing the disturbance transfer function, we can set the input r(t) to 0. Finally, we can relate disturbance to output:

$$y = Cx + [D\ 0]u' = C\left( sI - \left(A - [B\ F] \times K'\right)\right)^{-1} FD_{dist}$$

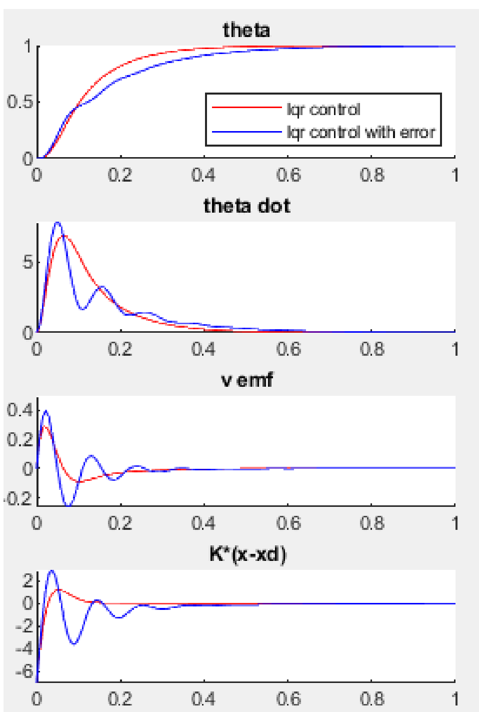
The closed loop transfer function is:

$$H_{close} = C\left( sI - \left(A - [B\ F] \times K'\right)\right)^{-1} F$$

3. **Example: propellor arm**

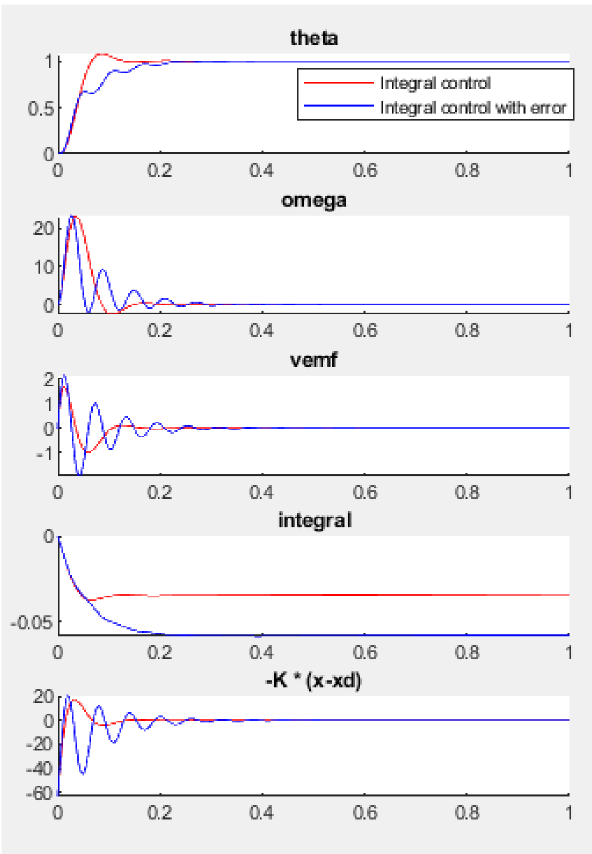
We are going to take a quick look of lab 6 – controlling the propellor arm using LQR control. We have derived the system state-space model last week. You will practice implementing part of the model and controller in the lab. Here, we show some sample results.

First, we show the simulation result of LQR control. The system has 3 states, and we want the propellor arm to reach 1 radian following a step input. The top three panels show the 3 state values, and the lower panel is the weighted sum of error. The red curve shows the case of controlling the system with an ideal feedback matrix K. The blue curve shows another controller where the control matrix K' is generated from noisy A and B matrices. If we generate the controller matrix K with an inaccurate system model, then we see more oscillations. There is also a tiny steady state error (about 0.5%) caused by the model inaccuracy. In addition, there may be additional error if there is external disturbance.



Next, we implement the integral LQR controller. The results are shown on the next page. Note that there is an added state, which is the integral of the error. As long as this state reaches a steady state value, then the derivative is 0, which implies that there is no steady state error. The comparison of a perfect integral LQR controller and another controller generated with a modeling error is shown below. Note that there is no steady state error.

In practice, this controller is much better at handling external disturbance. You will implement it in lab 5B.




4. **Motivation: sensor noise**

Thus far, we have shown that state-space control is more general than PID control because it gives us access to all the states. The LQR formulation is also more intuitive for us to set the control matrix K. However, the current formulation has a problem: it assumes we have perfect knowledge of all the state values x. In reality, this is difficult. It is difficult to directly measure all the state variables, and if we take the derivatives numerically, then the state variables can be noisy. How do we measure or estimate the state variables x?

We will introduce a new topic next lecture, which is called an observer. The core idea is that based on a limited set of sensor-measured data, we will estimate the state values. The control input will be generated based on the estimated state values, and you will see a significant performance improvement.