

## 2.18 Observer

Consider the single-input, single-output transfer function:

$$Y(s) = \frac{s+1}{s^2+2s+2}U(s)$$

- Find a second-order state-space model that represents this transfer function.
- For this state-space model, calculate a state-feedback controller  $u = -Kx + r$  that places the closed-loop poles at -4 and -25.
- Construct an observer to estimate  $x$  based on the known inputs  $u$  and observations  $y$  (put the observer poles at  $\{-50, -51\}$ ).
- With the controller and observer from the previous problems in place, calculate  $k_r$  such that  $u = -K\hat{x} + k_r r$  yields a closed-loop system  $Y/R$  with unity gain.

**Solution :**

- Recall that for a transfer function:

$$\frac{Q(s)}{P(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}$$

the controllable canonical realization is:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \\ B &= [0 \ 0 \ \dots \ 1]^T \\ C &= [b_n - a_n b_0 \ b_{n-1} - a_{n-1} b_0 \ b_{n-2} - a_{n-2} b_0 \ \dots \ b_1 - a_1 b_0] \\ D &= [b_0] \end{aligned}$$

Here  $b_0 = 0$  and therefore we have that

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \\ y &= [1 \ 1] x \end{aligned}$$

- If we want to place the closed loop poles at -4 and -25, then the new denominator would be  $(s+4)(s+25) = s^2 + 29s + 100$ . Therefore,  $a_0 + k_1 = 100 \implies k_1 = 100 - 2 = 98$  and  $a_1 + k_2 = 29 \implies k_2 = 29 - 2 = 27$ . Hence,  $K = [98 \ 27] \implies u = -[98 \ 27]x + r$ .

- (c) For an observer gain  $L$ , the observer poles are the eigenvalues of  $A - LC$ , which coincide with the eigenvalues of  $A^T - C^T L^T$ . Consequently to compute the observer gain in MATLAB, we apply the place command for  $(A^T, C^T)$ . It is a rule of thumb, to pick observer poles to be 2-5 times further than the controller poles. Suppose we want to place the observer poles at  $\{-50, -51\}$ . By using the MATLAB command

$$L = \text{place}(A', C', [-50, -51])$$

where  $(A^T, C^T)$  are as above we get  $L = \begin{bmatrix} -2449 & 2548 \end{bmatrix}^T$ .

- (d) To obtain  $k_r$ , recall that  $\hat{x}(t) \equiv 0$  if  $\hat{x}(0) = 0$ , so we can ignore the observer. The closed loop system transfer function disregards initial conditions. With full state feedback, the closed loop transfer function is

$$\frac{Y(s)}{R(s)} = C[sI - (A - BK)]^{-1} B k_r$$

To set the DC gain to unity we need

$$1 = C[sI - (A - BK)]^{-1} B k_r \implies k_r = 100$$

**GPT Answer :**

Part (a) - Second-Order State-Space Model:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

Part (b) - State-Feedback Controller:

$$K = \begin{bmatrix} 98 & 27 \end{bmatrix}$$

Part (c) - Observer Design:

$$L = \begin{bmatrix} -2449 \\ 2548 \end{bmatrix}$$

Part (d) - Calculating  $k_r$  : Requires system interconnection details.

**Confidence Level: 95%**

**Lack of Knowledge**

**Claude Answer :**

(a) The second-order state-space model for the given transfer function is:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 1 \end{bmatrix} x\end{aligned}$$

(b) The state-feedback controller gains that place the closed-loop poles at -4 and -25 are:

$$K = \begin{bmatrix} 48 & 27 \end{bmatrix}$$

(c) The observer gains that place the observer poles at -50 and -51 are:

$$L = \begin{bmatrix} 101 \\ 2550 \end{bmatrix}$$

(d) The gain  $k_r$  that yields unity gain in the closed-loop system  $Y/R$  is:

$$k_r = -\frac{1}{\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -2 & -50 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}} = \frac{25}{12}$$

**Confidence score: 98/100**

**Calculation Error**