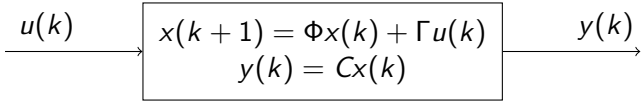
Discrete-time Ouput feedback (state feedback with observer)

Kjartan Halvorsen

November 26, 2021

The discrete-time state-space model

The discrete-time state-space model



Stability

Eigenvalues and eigenvectors

Definition The eigenvalues $\lambda_i \in \mathbb{R}$ and eigenvectors $v_i \in \mathbb{R}^n$ of a matrix $\Phi \in \mathbb{R}^{n \times n}$ are the n pairs $(\lambda_i, v_i \neq 0),\ i = 1, 2, \ldots, n$ that satisfy

$$\Phi v_i = \lambda_i v_i$$

Stability

The system

$$x(k+1) = \Phi x(k), \quad x(0) = x_0$$

is stable if $\lim_{t\to\infty} x(kh) = 0, \quad \forall\, x_0 \in \mathbb{R}^n$.

A necessary and sufficient requirement for stability is that all the eigenvalues of Φ are inside the unit circle.

The eigenvalues of Φ are the poles of the system.

State feedback control

State feedback control

Given

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

and measurements (or an estimate) of the state vector $x(k)$.

Linear state feedback is the control law

$$\begin{aligned} u(k) &= f\big((x(k), u_c(k)\big) = -l_1 x_1(k) - l_2 x_2(k) - \cdots - l_n x_n(k) + l_0 u_c(k) \\ &= -Lx(k) + l_0 u_c(k), \end{aligned}$$

where

$$L = \begin{bmatrix} l_1 & l_2 & \cdots & l_n \end{bmatrix}.$$

Substituting this in the state-space model (14) gives

$$\begin{aligned} x(k+1) &= (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \\ y(k) &= Cx(k) \end{aligned} \tag{2}$$

Pole placement by state feedback

Given (or choosing) a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2)\cdots(z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n. \tag{3}$$

Pole placement by state feedback

Given (or choosing) a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2)\cdots(z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n. \tag{3}$$

Linear state feedback gives the system

$$x(k+1) = (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \tag{4}$$

with characteristic polynomial

$$\det\left(zI - (\Phi - \Gamma L)\right) = z^n + \beta_1(l_1, \ldots, l_n) z^{n-1} + \cdots \beta_n(l_1, \ldots, l_n). \tag{5}$$

Pole placement by state feedback

Given (or choosing) a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2)\cdots(z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n. \tag{3}$$

Linear state feedback gives the system

$$x(k+1) = (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \tag{4}$$

with characteristic polynomial

$$\det\left(zI - (\Phi - \Gamma L)\right) = z^n + \beta_1(l_1, \ldots, l_n) z^{n-1} + \cdots \beta_n(l_1, \ldots, l_n). \tag{5}$$

Set the coefficients of the desired characteristic polynomial (11) equal to the coefficients of (5) to obtain the system of equations

$$\begin{aligned} \beta_1(l_1, \ldots, l_n) &= \alpha_1 \\ \beta_2(l_1, \ldots, l_n) &= \alpha_2 \\ &\vdots \\ \beta_n(l_1, \ldots, l_n) &= \alpha_n \end{aligned}$$

Pole placement by state feedback

The system of equations

$$\begin{aligned} \beta_1(l_1, \ldots, l_n) &= \alpha_1 \\ \beta_2(l_1, \ldots, l_n) &= \alpha_2 \\ &\vdots \\ \beta_n(l_1, \ldots, l_n) &= \alpha_n \end{aligned}$$

is always linear in the parameters of the controller, hence

$$M L^{\mathrm{T}} = \alpha,$$

where $\alpha^{\mathrm{T}} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{bmatrix}$.

Pole placement by state feedback

Given a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2)\cdots(z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n.$$

and closed-loop system

$$\begin{aligned} x(k+1) &= (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \\ y(k) &= Cx(k) \end{aligned}$$

The Matlab (*control systems toolbox*) has methods for computing the gain vector L

1. Ackerman's method

   ```
   L = acker(Phi, Gamma, pd)
   ```

2. Numerically more stable method

   ```
   L = place(Phi, Gamma, pd)
   ```

The reference input gain l_0

The closed-loop state space system

$$x(k+1) = \underbrace{(\Phi - \Gamma L)}_{\Phi_c} x(k) + l_0 \Gamma u_c(k)$$
$$y(k) = Cx(k)$$

with constant reference signal $u_c(k) = u_{c,f}$ has the steady-state solution $(x(k+1) = x(k))$

The reference input gain l_0

The closed-loop state space system

$$x(k+1) = \underbrace{(\Phi - \Gamma L)}_{\Phi_c} x(k) + l_0 \Gamma u_c(k)$$
$$y(k) = Cx(k)$$

with constant reference signal $u_c(k) = u_{c,f}$ has the steady-state solution $(x(k+1) = x(k))$

$$x_f = l_0 (I - \Phi_c)^{-1} \Gamma u_{c,f}$$
$$y_f = C x_f = l_0 C (I - \Phi_c)^{-1} \Gamma u_{c,f}.$$

We want $y_f = u_{c,f}$,

$$\Rightarrow \qquad l_0 = \frac{1}{C(I - \Phi_c)^{-1}\Gamma}$$

State feedback with reconstructed states

Observer design

Given model

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = Cx(k)$$

and measurements of the output signal $y(k)$.

The observer is given by

$$\hat{x}(k+1) = \underbrace{\Phi\hat{x}(k) + \Gamma u(k)}_{\text{simulation}} + \underbrace{K\big(y(k) - C\hat{x}(k)\big)}_{\text{correction}} = (\Phi - KC)\,\hat{x}(k) + \Gamma u(k) + Ky(k)$$

with poles given by the eigenvalues of the matrix $\Phi_o = \Phi - KC$

Rule-of-thumb (From continous-time state-space theory) Choose the poles of the observer (eigenvalues of $\Phi - KC$) at least twice as fast as the poles (eigenvalues) of $\Phi - \Gamma L$. In discrete-time place the observer-poles closer to the origin.

Control by feedback from reconstructed states

The design problem can be separates into two problems

1. Determine the gain vector L and the gain l_0 of the control law

$$u(k) = -L\hat{x}(k) + l_0 u_c(k)$$

so that the closed-loop system has good reference tracking.

2. Determine the gain vector K of the observer

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K\big(y(k) - C\hat{x}(k)\big)$$

to get a good balance between disturbance rejection and noise attenuation.

Computing the observer gain

A matrix M and its transpose M^{T} have the same eigenvalues. Hence, the problem of determining the gain K to obtain desired eigenvalues of

$$\Phi - KC$$

is equivalent to determining the gain K in

$$(\Phi - KC)^{\mathrm{T}} = \Phi^{\mathrm{T}} - C^{\mathrm{T}}K^{\mathrm{T}}.$$

The last problem has the exact same form as the problem of determining L to obtain desired eigenvalues of

$$\Phi - \Gamma L$$

So, the same matlab function can be used for both problems.

Computing the observer gain

1. Ackerman's method

   ```
   K = acker(Phi', C', po)'
   ```

2. More numerically stable method

   ```
   K = place(Phi', C', pd)'
   ```

Where to place the closed-loop poles?

Where to place the closed-loop poles?

If the system is controllable we can place the closed-loop poles freely.

Where to place the closed-loop poles?

If the system is controllable we can place the closed-loop poles freely.

But not all placements are good choices

Where to place the closed-loop poles?

If the system is controllable we can place the closed-loop poles freely.

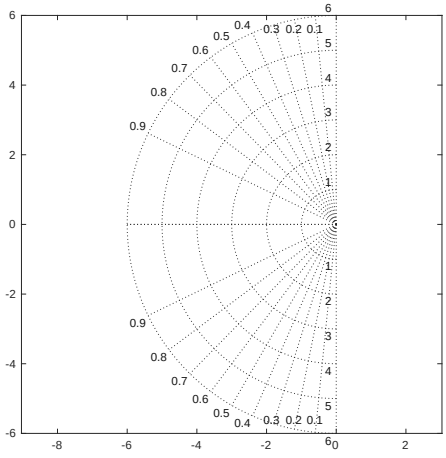
But not all placements are good choices

Take into account

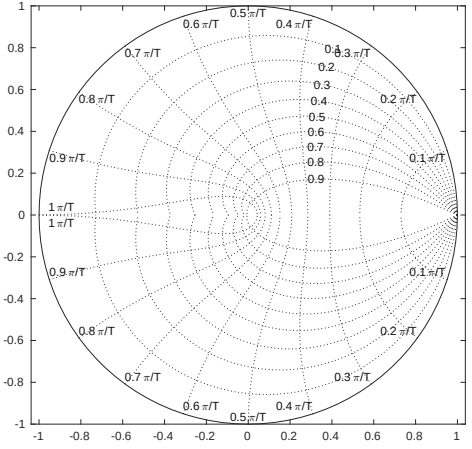
- Desired speed
- Desired damping
- Poles and zeros of the plant

Placing for desired speed and damping

s-plane



z-plane



The effect of plant zeros and poles

Feedback Systems

An Introduction for Scientists and Engineers

Karl Johan Åström
Richard M. Murray

The effect of plant zeros and poles

Feedback Systems

An Introduction for Scientists and Engineers

Karl Johan Åström
Richard M. Murray

12.4 Robust Pole Placement

In Chapters 6 and 7 we saw how to design controllers by setting the locations of the eigenvalues of the closed loop system. If we analyze the resulting system in the frequency domain, the closed loop eigenvalues correspond to the poles of the closed loop transfer function and hence these methods are often referred to as design by *pole placement*.

State space design methods, like many methods developed for control system design, do not explicitly take robustness into account. In such cases it is essen-

Placing w.r.t plant zeros and poles

- Cancel slow plant zeros
- Place fast closed-loop poles near fast plant poles