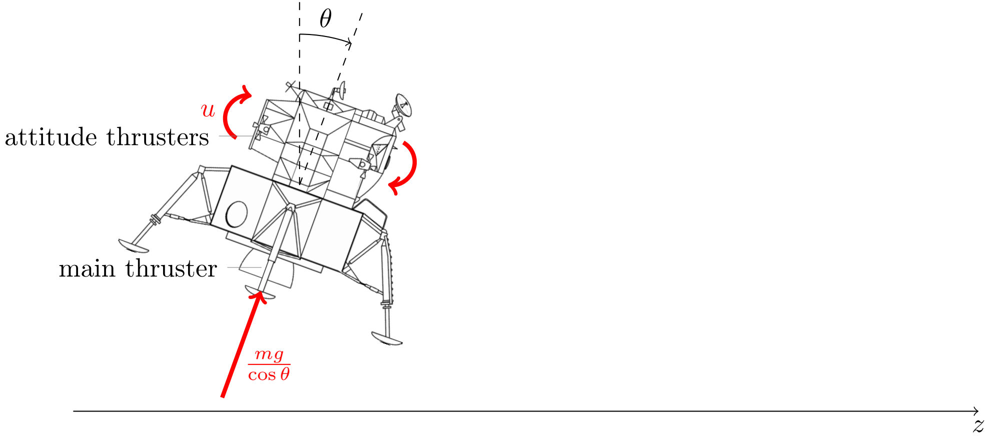
Output feedback (observer)

Kjartan Halvorsen

July 26, 2021

Example - The Apollo lunar module



$$u(t) \longrightarrow \boxed{\frac{k_1}{s}} \xrightarrow{\dot{\theta}(t)} \boxed{\frac{1}{s}} \xrightarrow{\theta(t)} \boxed{k_2} \xrightarrow{\ddot{z}(t)} \boxed{\frac{1}{s}} \xrightarrow{\dot{z}(t)}$$

Example - The Apollo lunar module

State variables: $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} \dot{\theta} & \theta & \dot{z} \end{bmatrix}^T$. With dynamics

$$\begin{cases} \dot{x}_1 = \ddot{\theta} = k_1 u \\ \dot{x}_2 = \dot{\theta} = x_1 \\ \dot{x}_3 = \ddot{z} = k_2 \theta = k_2 x_2 \end{cases}$$

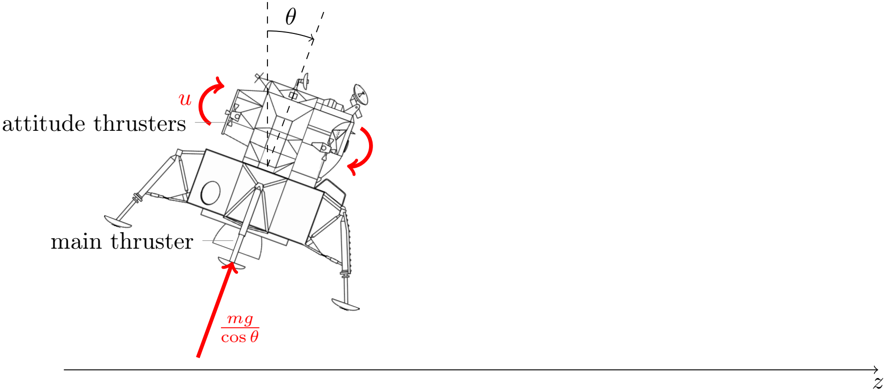
$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & k_2 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} k_1 \\ 0 \\ 0 \end{bmatrix}}_{B} u$$

Example - The Apollo lunar module

$$
\begin{aligned}
x(kh+h) &= \mathrm{e}^{Ah}x(kh) + \int_0^h \mathrm{e}^{As}Bu(kh+h-s)ds \\
&= \underbrace{\mathrm{e}^{Ah}}_{\Phi(h)}x(kh) + \underbrace{\left(\int_0^h \mathrm{e}^{As}Bds\right)}_{\Gamma(h)}u(kh) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ \frac{h^2k_2}{2} & hk_2 & 1 \end{bmatrix} x(kh) + k_1 \begin{bmatrix} h \\ \frac{h^2}{2} \\ \frac{k_2h^3}{6} \end{bmatrix} u(kh)
\end{aligned}
$$

State feedback with reconstructed states

State feedback with reconstructed states



$$u(t) \longrightarrow \boxed{\frac{k_1}{s}} \xrightarrow{\dot{\theta}(t)} \boxed{\frac{1}{s}} \xrightarrow{\theta(t)} \boxed{k_2} \xrightarrow{\ddot{z}(t)} \boxed{\frac{1}{s}} \xrightarrow{\dot{z}(t)}$$

State feedback

Given

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

and measurements (or estimates) 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 the control law in the state space model (8) gives

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

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 obserser is given by

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

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

Rule-of-thumb Choose the poles of the observer (eigenvalues of $\Phi - KC$) at least twice as fast as the poles (eigenvalues) of $\Phi - \Gamma L$.

Observer design

Rule-of-thumb Choose the poles of the observer (eigenvalues of $\Phi - KC$) at least twice as fast as the poles (eigenvalues) of $\Phi - \Gamma L$.

In continuous time (the s-plane), choosing a pole to be twice as fast, means moving the pole to twice the disance from the origin. Given a discrete pole p_1, the discrete pole in

$$p_2 = \exp\left(2\frac{\ln p_1}{h}h\right) = \exp\bigl(2\ln p_1\bigr) = p_1^2$$

corresponds to a response that is twice as fast.

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)'
```