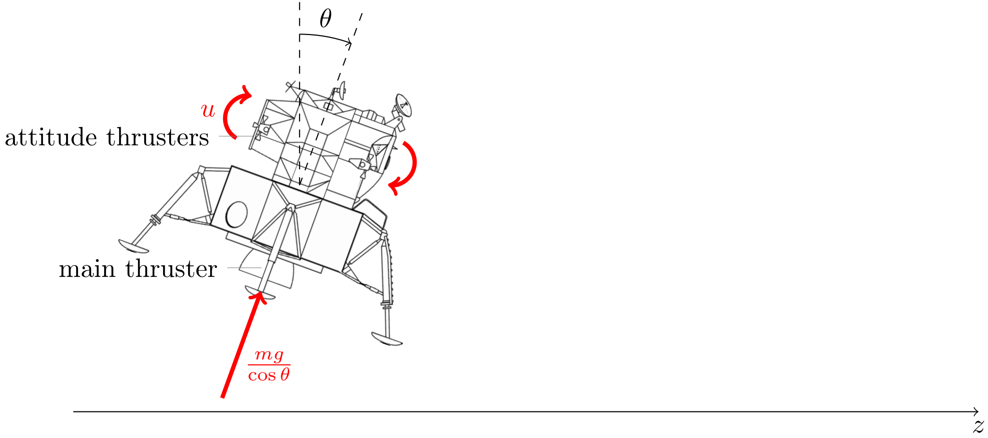
Control computarizado - Retroalimentación con observador

Kjartan Halvorsen

July 28, 2020

Ejemplo - El módulo lunar de Apollo



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

Ejemplo - El módulo lunar de Apollo

Variables del estado: $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} \dot\theta & \theta & \dot z \end{bmatrix}^T$. Con dinamica

$$\begin{cases} \dot x_1 = \ddot\theta = k_1 u \\ \dot x_2 = \dot\theta = x_1 \\ \dot x_3 = \ddot z = k_2\theta = k_2 x_2 \end{cases}$$

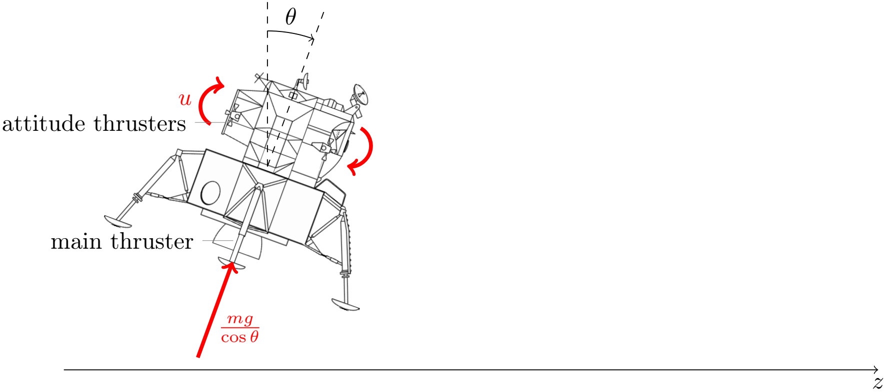
$$\dot x = \begin{bmatrix} \dot x_1 \\ \dot x_2 \\ \dot x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & k_2 & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} k_1 \\ 0 \\ 0 \end{bmatrix}}_{B} u$$

Discretización - Módulo lunar de Apollo

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

Control por retroalimentación de estados reconstruidos

Control por retroalimentación de estados reconstruidos



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

Retroalimentación de estados

Dado

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

y medidas (o valores estimados) del vector de estado $x(k)$.

Retroalimentación lineal de estados es la ley de control

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

dónde

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

Sustituyende la ley de control en el modelo en espacio de estado (8) da

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

Diseño del observador

Dado modelo

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

y medidas de la senal de salida $y(k)$.

El observador sería

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

con polos dados por los eigenvalores de la matriz $\Phi_o = \Phi - KC$

Regla de oro Eliga los polos del observador (eigenvalores de $\Phi - KC$) por lo menos dos veces más rápidos que los polos (eigenvalores) de $\Phi - \Gamma L$.

Diseño del observador

Regla de oro Eliga los polos del observador (eigenvalores de $\Phi - KC$) por lo menos dos veces más rápidos que los polos (eigenvalores) de $\Phi - \Gamma L$.

En tiempo continua (el plano s) hacer un polo dos veces más rápido significa mover el polo a una distancia doble del origen. Entonces, con el polo discreto p_1 el polo discreto en

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

corresponde a una respuesta dos veces más rapida.

Control por retroalimentación de estados reconstruidos

Se puede separar el problema del diseño del controlador en dos

1. Diseñar el vector de ganancias L en la ley de control

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

   para obtener buen seguimiento a referencia.

2. Diseñar el vector de ganancias K en el observador

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

   para obtener una balancia entre rechazo a perturbaciones y sensibilidad a ruido de medida.

Matlab/simulink

Calculo de la ganancia del observador

Una matriz M y su transponada M^{T} tienen los mismos eigenvalores. Entonces, el problema de determinar las ganancias K para obtener eigenvalores deseados de

$$\Phi - KC$$

es equivalente de determinar las ganancias K en

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

Este último problema tiene la misma forma como el problem de determinar L para obtener eigenvalores deseados de

$$\Phi - \Gamma L$$

Entonces, se puede usar el mismo método en matlab para calcular los dos vectores de ganancia, L y K

Calculo de la ganancia del observador

1. Método de Ackerman

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

2. Método numericamente estable

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

Matlab

Actividad

En grupos:

1. Agrega simulación de una perturbancia constante a la salida del sistema (pero antes de la medición). Esa perturbación representa un error de modelación. Simula un escalón positivo unitario occurriendo en $t = 10\text{s}$
2. Determina la desviación máxima en la velocidad en respuesto a la perturbancia.
3. Determina las ganancias $K = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}^{\mathrm{T}}$ para un observador *dead-beat*. Es decir con todos los polos en el origen. Cómo cambia la desviación en la respuesta a la perturbancia?
4. Cambió la respuesta al escalón en la referencia al cambiar a un observador dead-beat?