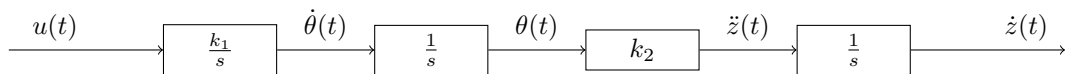
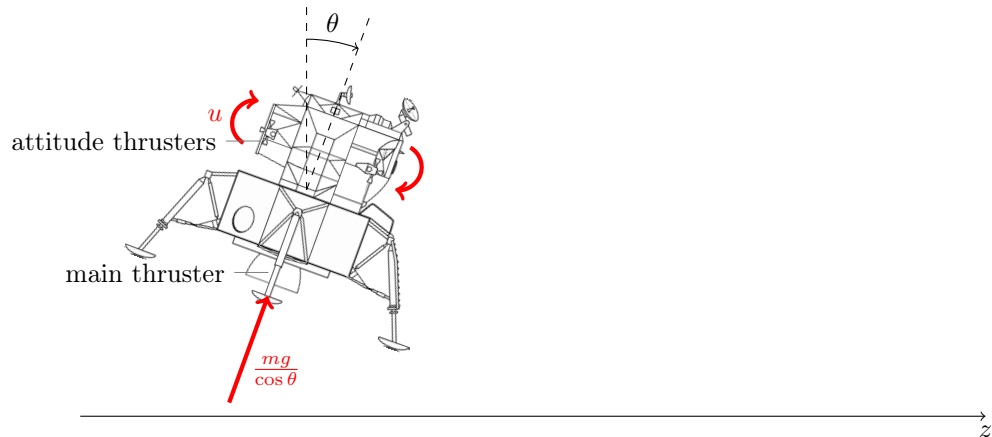


Control computarizado - Retroalimentación con observador

Kjartan Halvorsen

July 25, 2021

Ejemplo - El módulo lunar de Apollo



Ejemplo - El módulo lunar de Apollo

Variables del estado: $x = [x_1 \ x_2 \ x_3]^T = [\dot{\theta} \ \theta \ \dot{z}]^T$. Con dinamica

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

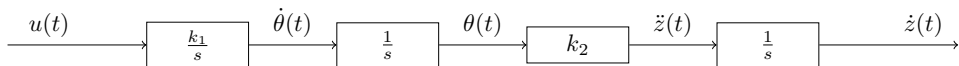
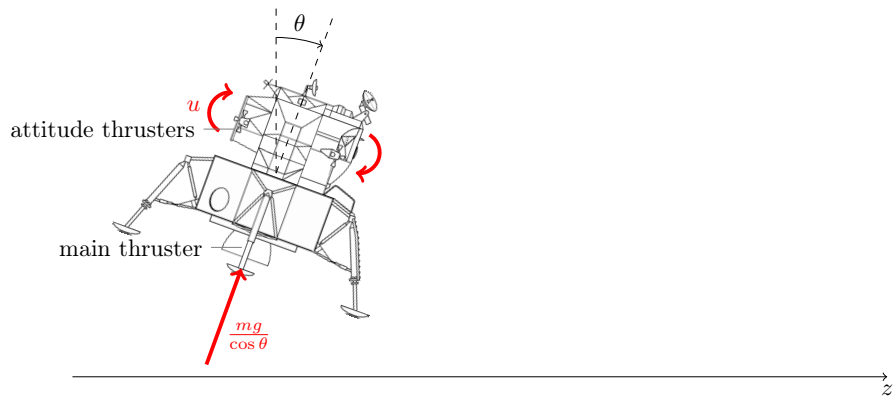
$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & k_2 & 0 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} k_1 \\ 0 \\ 0 \end{bmatrix}}_B u$$

Discretización - Módulo lunar de Apollo

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

Control por retroalimentación de estados reconstruidos

Control por retroalimentación de estados reconstruidos



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

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

dónde

$$L = [l_1 \quad l_2 \quad \cdots \quad l_n].$$

Sustituyendo 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

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

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

El observador sería

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

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, dado el polo discreto p_1 , el polo discreto en

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

corresponde a una respuesta dos veces más rápida.

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

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

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

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

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

$K = (\text{acker}(\Phi', C', p_o))'$

2. Método numericamente estable

$K = (\text{place}(\Phi', C', p_d))'$

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 ocurriendo en $t = 10\text{s}$
2. Determina la desviación máxima en la velocidad en respuesta a la perturbancia.
3. Determina las ganancias $K = [k_1 \quad k_2 \quad k_3]^T$ 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*?