Avaliação Espaço de Estados 2

Erick N. M. Alves e Robson R. T. Júnior

02 de setembro de 2024

Dado um sistema descrito por:

$$G(s) = \frac{2s+1}{s^3+3} \quad , \tag{1}$$

cujos estados estão disponíveis para realimentação.

Deseja-se projetar um controlador por realimentação de estados que assegure um tempo de acomodação igual a 2 s e máximo sobressinal de 8%. O sistema, em malha fechada, deve também ter erro nulo para entrada em degrau e ser tolerante a erros nos valores dos coeficientes do sistema.

Projeto da Topologia de Controle por Realimentação de Estados:

Uma vez que deseja-se erro de regime permanente nulo para entrada em degrau e tolerância a erros dos parâmetros do sistema (modelo), é adotada uma topologia de realimentação de estados com integrador atuando sobre o erro de seguimento, conforme o diagrama mostrado na Figura 1.

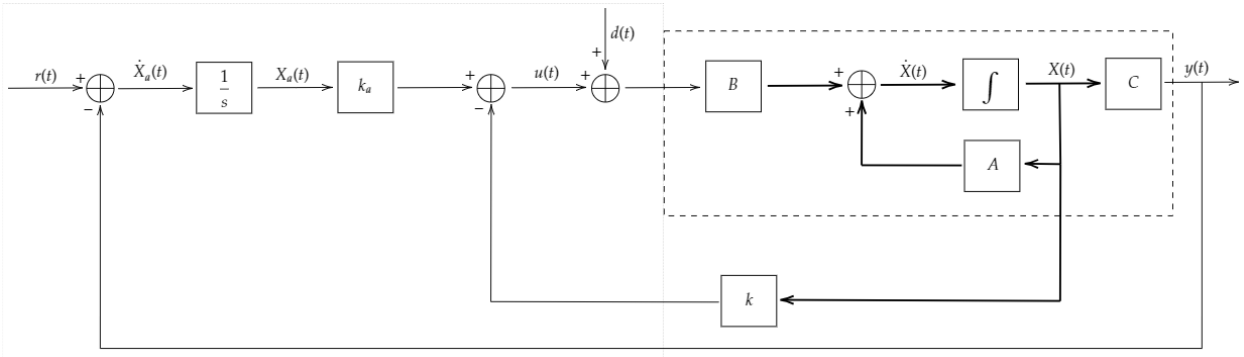

Figure 1: Topologia para realimentação de estados com integrador.

Inicialmente, descreve-se o sistema em espaço de estados:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 2 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix} \quad . \tag{2}$$

Avaliando-se os polos do sistema, têm-se que: $p = [-1,4422 \quad 0,7211+1,429j \quad 0,7211-1,429j]$ e os zeros: $z = [-0,5]$.

Para o projeto, a partir do *overshoot* desejado, obtem-se o coeficiente de amortecimento desejado:

$$\zeta = -\frac{\ln(0.08)}{\sqrt{\pi^2 + \ln^2(0.08)}} = 0,6266 \quad . \tag{3}$$

A partir do valor obtido para o coeficiente de amortecimento e o tempo de acomodação desejado, calcula-se a frequência natural desejada:

$$\omega_n = \frac{4}{0.6266 \cdot 2} = 3,1919 \ rad/s \quad . \tag{4}$$

O par de polos complexos que deseja-se ser dominante será:

$$pd = -\zeta\omega_n \pm \omega_n\sqrt{1-\zeta^2} = -2 \pm 2,4877 \quad . \tag{5}$$

A malha fechada terá quatro polos, sendo os três do sistema original, mais um adicionado pelo integrador. Assim, é necessário alocar outros dois polos de forma que eles não interfiram (ou sua interferência seja desprezável). Adota-se portanto, a estratégia de alocar um desses polos em $-0,5$ a fim de cancelar o zero do sistema, e outro uma década a baixo da dinâmica dominante, em -20, visando uma interferência mínima na resposta.

A partir dos polos que deseja-se alocar, são construídas as matrizes aumentadas para os estados, a fim de englobar o estado x_a do integrador:

$$\tilde{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \qquad \tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad . \tag{6}$$

É montada então, uma matriz F com os autovalores (polos) desejados em sua diagonal, sendo, neste caso:

$$F = \begin{bmatrix} -0,5 & 0 & 0 & 0 \\ 0 & -20 & 0 & 0 \\ 0 & 0 & -2 & -2,4877 \\ 0 & 0 & 2,4877 & -2 \end{bmatrix} \quad . \tag{7}$$

Assim, solucionando a equação de Sylvester, para um vetor $\tilde{K}$ unitário, com o auxílio da função de *Lyapunov*: $T = lyap\left(\tilde{A}, -F, -\tilde{B}\tilde{K}\right)$, obtem-se a matriz de transformação de base:

$$T = \begin{bmatrix} -0,3478 & 0,0001 & -0,0142 & -0,0375 \\ 0,1739 & -0,0025 & -0,0649 & 0,1105 \\ -0,087 & 0,05 & 0,4046 & -0,0595 \\ 0 & 0 & 0,0165 & 0,0712 \end{bmatrix} \quad . \tag{8}$$

Essa matriz de transformação de base é tal que:

$$\bar{K} = \tilde{K}T^{-1} \quad , \tag{9}$$

sendo que o vetor $\bar{K}$ contém os ganhos para realimentação dos estados do sistema e também da ação integral:

$$\bar{K} = \begin{bmatrix} K & -k_a \end{bmatrix} \quad . \tag{10}$$

Após cálculos utilizando as bibliotecas do *Python*, obtem-se o vetor:

$$\bar{K} = \begin{bmatrix} 42,0943 & 102,1885 & 24,5 & -101,8851 \end{bmatrix} \quad . \tag{11}$$

Portanto, os ganhos projetados para o controlador *PI-Like* (realimentação de estados mais um integrador) serão:

$$K = \begin{bmatrix} 42,0943 & 102,1885 & 24,5 \end{bmatrix} \qquad \& \qquad k_a = 101,8851 \quad , \tag{12}$$

resultando em uma lei de controle por realimentação de estados mais integrador:

$$u(t) = r(t) - KX(t) + k_a x_a(t) \quad , \tag{13}$$

em que o estado aumentado x_a trata-se da integral do erro no tempo.

Ao conferir os polos de malha fechada do sistema compensado, observou-se que o projeto alocou os polos conforme o desejado: $p_{mf} = [-0,5 \quad -20 \quad -2+2,4877j \quad -2-2,4877j]$.

Análise dos Resultados:

Ao simular o sistema, caso os estados iniciais não sejam nulos, haverá uma acomodação desses estados e, consequentemente, da saída, para o sistema em um ponto de equilíbrio, como no caso da Figura 2, em que $r(t) = 0$ e o vetor de estados iniciais foi arbitrado como $X0 = [1 \quad 2 \quad 3 \quad 4]$.

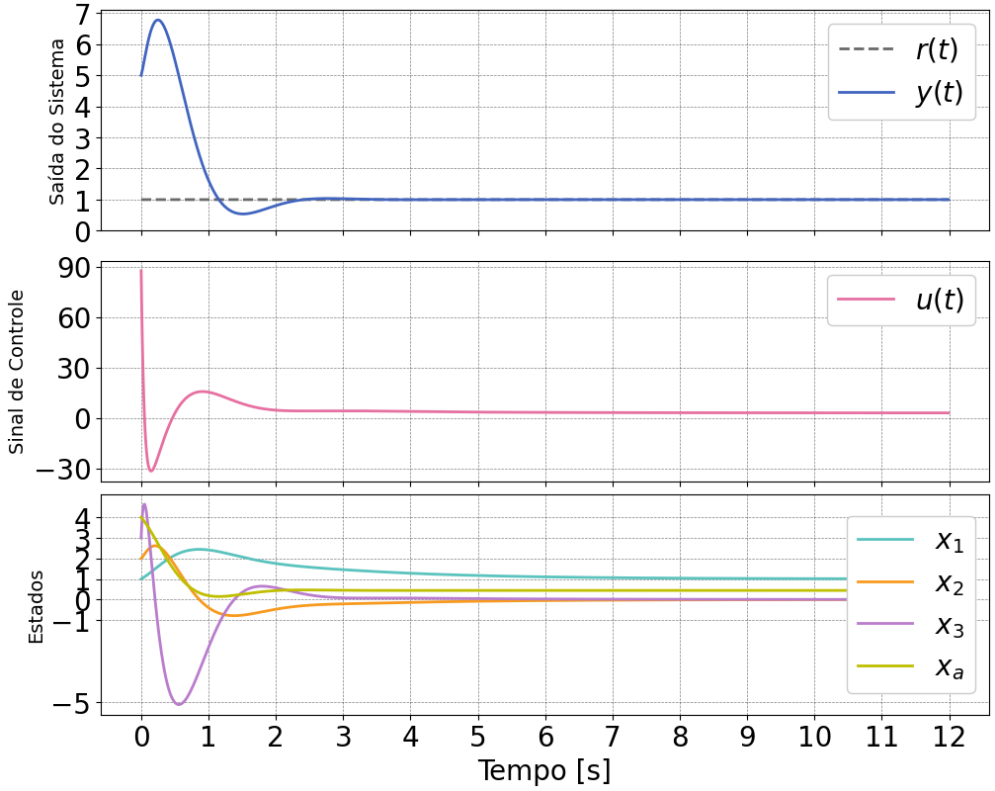

Figure 2: Acomodação do sistema para condições iniciais não nulas.

Outrossim, a Figura 3 mostra a aplicação de um degrau unitário ao sistema em um instante $t = 0$ s após a acomodação dos estados (regime permanente / equilíbrio).

Como pode ser visto, o tempo de acomodação é de aproximadamente 2 s e o *overshoot* foi de $7,88\%$. As especificações de projeto iniciais foram atendidas sem a necessidade de ajustes extras.

Em uma situação real, a incerteza nos parâmetros do projeto causaria uma má anulação do zero do sistema, o que poderia interferir na resposta do sistema. Ademais, se outra estratégia fosse adotada para escolha dos polos da malha fechada, o zero do sistema de fato alteraria a resposta do sistema, possivelmente acelerando-a e causando *overshoot*.

Para os casos citados, seriam necessários reajustes na alocação dos polos, buscando valores menores de *overshoot* e tempo de acomodação, a partir da alocação de polos com maior coeficiente de amortecimento e maior frequência natural.

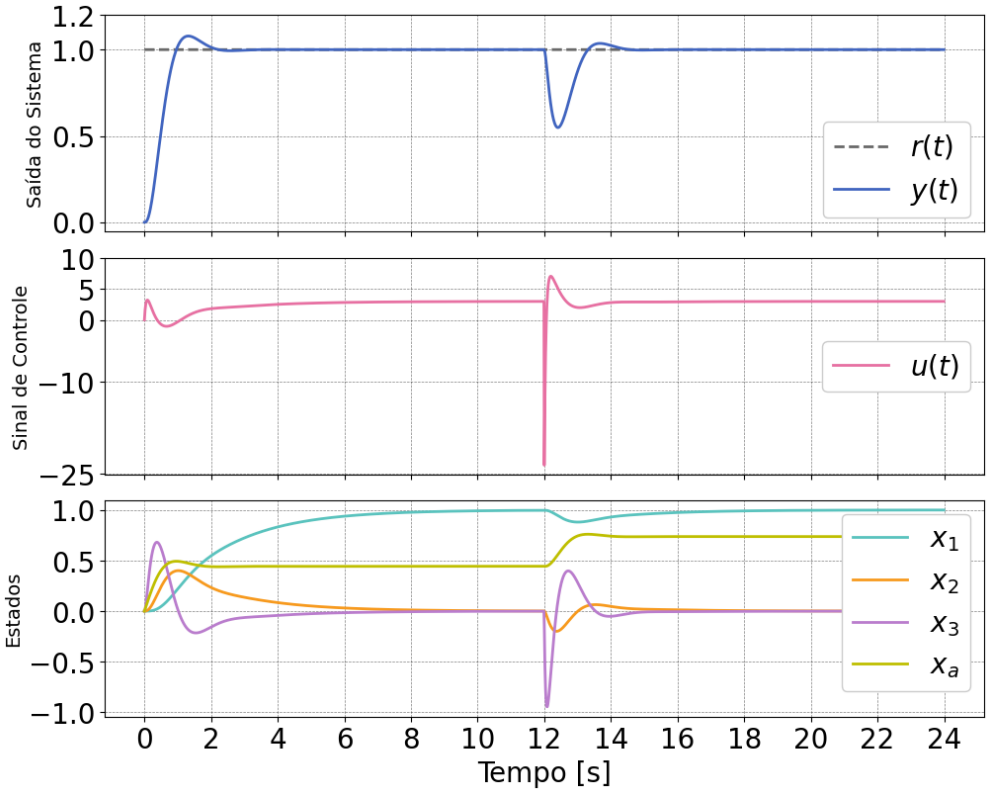

Figure 3: Aplicação de degrau unitário e distúrbio ao sistema.

Por fim, retomando a resposta mostrada na Figura 3, foi aplicada uma perturbação na entrada do sistema, no instante $t = 12\ s$, a fim de verificar sua robustez à perturbações externas. Graças ao integrador, a perturbação foi rejeitada e a malha fechada garantiu o erro nulo em regime permanente. Esse comportamento é semelhante à variação de parâmetros do sistema, sendo um teste que comprova a eficiência da topologia adotada, também para a compensação de imprecisões nos modelos.

Cálculos e Simulação:

`https://github.com/ErickNMA/TC/blob/main/av_SS_2/SS-2.ipynb`