UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Controle e Servomecanismo

Prática 3 - Modelagem de velocidade de um motor CC

Professor: Prof. Roberto Santos Inoue

Alexandre Strabello, 770076, Engenharia Física

São Carlos, 27 de junho de 2023

1 Pré Tutorial

Todos os códigos, imagens e documentos utilizados para a escrita do presente relatório estão disponíveis no repositório `https://github.com/alestrab/Controle-e-Servo`, na pasta 'prática 3'.

Com base na segunda lei de Newton, a equação regente para o movimento de um rotor é dado por:

$$J\ddot{\theta} + b\dot{\theta} = K\,i \tag{1}$$

Da mesma forma, considerando um circuito elétrico, os enrolamentos no motor podem ser descritos pela equação:

$$L\frac{\mathrm{d}i}{\mathrm{d}t} + R\,i = v - K\dot{\theta} \tag{2}$$

Considerando como sinal de entrada v e sinal de saída $\dot{\theta}$, pode-se obter a função de transferência do sistema. Primeiramente realizando a transformada de Laplace para (1) e (2), obtém-se:

$$\begin{gathered} J\,s\,\dot{\Theta} + b\,\dot{\Theta} = K\,I \\ L\,s\,I + R\,I = V - K\,\dot{\Theta} \end{gathered}$$

Isolando I como função de $\dot{\Theta}$, obtém-se:

$$I = \frac{V - K\dot{\Theta}}{L\,s + R}$$

Substituindo na expressão para $\dot{\Theta}$:

$$\begin{gathered} J\,s\,\dot{\Theta} + b\,\dot{\Theta} = K\,I \\ J\,s\,\dot{\Theta} + b\,\dot{\Theta} = K\,\frac{(V - K\dot{\Theta})}{L\,s + R} \\ \dot{\Theta}\left[(J\,s + b)\,(L\,s + R) + K^2\right] = K\,V \\ \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(J\,s + b)\,(L\,s + R) + K^2} \end{gathered}$$

Esse sistema pode ser solucionado das mais diferentes formas, utilizando tanto o Simulink quanto o Matlab. Pode-se ainda utilizar diretamente (1) e (2) para fazer a solução com auxílio de blocos integradores.

2 Execução e Avaliação de Resultados

Como forma geral tomou-se $\dot{\Theta}(0) = 0$, bem como $L = 0.5$; $R = 1$; $J = 0.01$; $B = 0.1$; $Ke = 0.01$; $Kt = 0.01$.

2.1 Solução utilizando Simulink

A solução do sistema utilizando Simulink pode ser obtida tanto com blocos integradores quanto por funções de transferência. Ambas opções foram adotadas, com a função de transferência sendo separada em duas componentes, apenas para ilustrar a semelhança entre as duas formas de resolução. O sistema montado está apresentado na Figura 1.

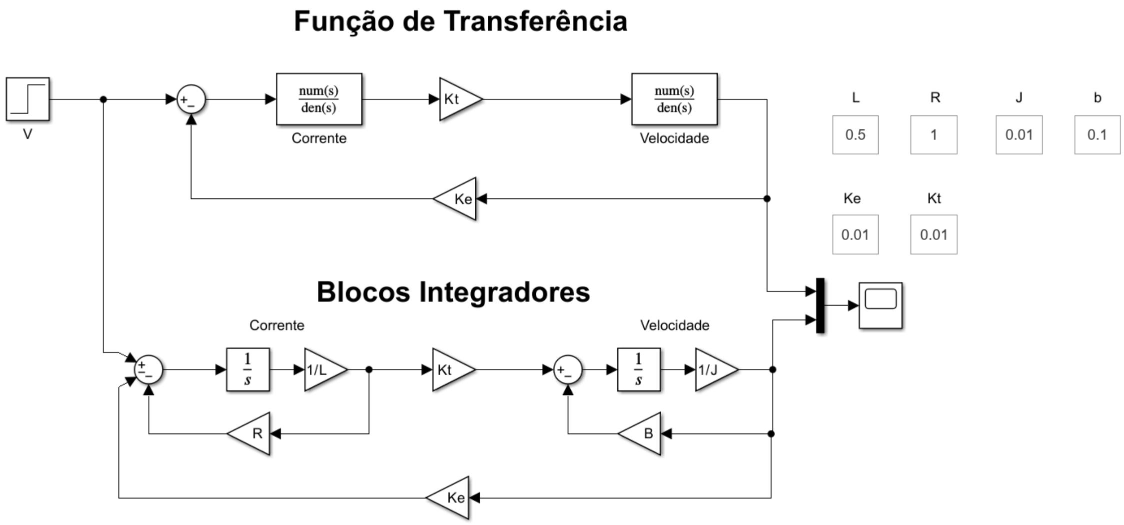


Figura 1: Simulação da dinâmica de um motor CC

Foram realizados testes utilizando tanto uma entrada com degrau unitário de amplitude igual a 1 quanto também um impulso, ambos iniciando em $t = 1\,s$. Os resultados obtidos para as simulações são apresentadas na Figura 2.

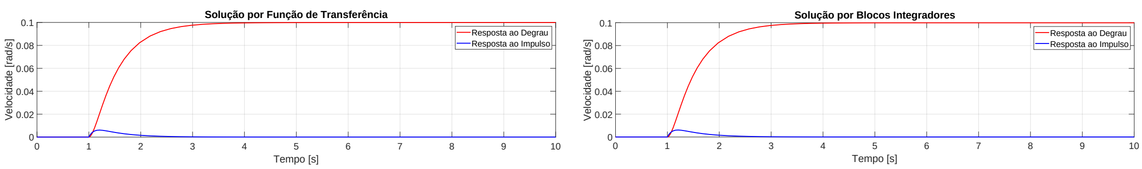


Figura 2: Resposta do motor CC para entradas do tipo degrau e impulso segundo simulação no Simulink

Pode-se observar que ambos os métodos funcionaram de maneira idêntica, aumentando a confiabilidade das simulações realizadas. Com relação ao tipo de entrada utilizada, observa-se que a resposta do tipo degrau corresponde à solução da equação diferencial para um regime permanente, enquanto o impulso analisa a resposta transitória desse sistema.

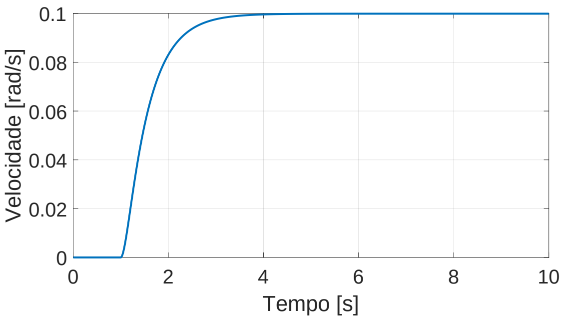
2.2 Solução por Função de Transferência no MATLAB

Outro método para solução consiste em aplicar a função de transferência diretamente no MATLAB. Assim, desenvolveu-se o código abaixo para simular a dinâmica do motor CC, dessa vez tomando uma única função de transferência, considerando como única saída a velocidade do motor.

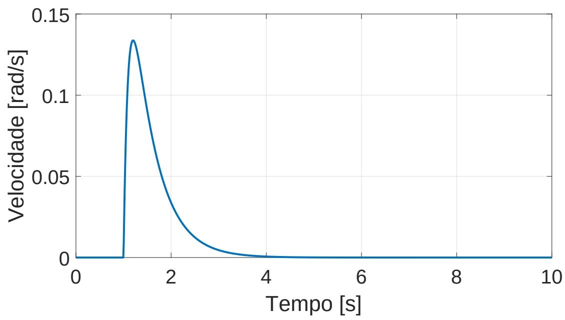
```
clear
clc

%Parametros simulacao
t_i=0;
t_f=10;

%Parametros do motor
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

%Fun  o de transferencia
tf=tf(K,[L*J,(R*J)+(L*b),b*R+K^2],'InputDelay',1.0);

%Apresentacao dos resultados
%Resposta ao degrau
[y_d,t] = step(tf,t_f);
%Resposta ao impulso
[y_i,t] = impulse(tf,t_f);
%Resposta ao seno
y_s = lsim(tf,sin(t),t);
subplot(2,2,1)
plot(t,y_d,'LineWidth',2)
ylabel('Velocidade [rad/s]')
xlabel('Tempo [s]')
grid()
ax = gca;
ax.FontSize = 20;

subplot(2,2,2)
plot(t,y_i,'LineWidth',2)
ylabel('Velocidade [rad/s]')
xlabel('Tempo [s]')
grid()
ax = gca;
ax.FontSize = 20;
```

```
subplot(2,2,3)
plot(t,y_s,'LineWidth',2)
ylabel('Velocidade [rad/s]')
xlabel('Tempo [s]')
grid()
ax = gca;
ax.FontSize = 20;
```

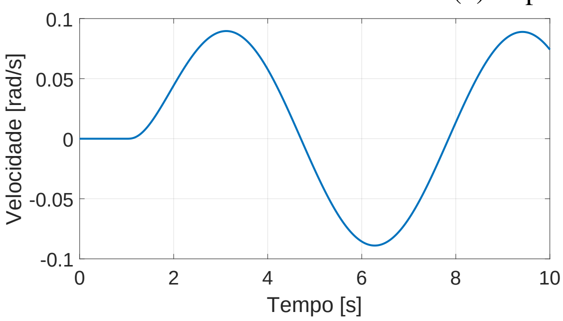
Adicionou-se um atraso de $1\,s$ para iniciar a simulação apenas para se adequar aos gráficos realizados no Simulink. Utilizou-se a função $lsim$ para execução da resposta para uma entrada senoidal, enquanto a resposta ao degrau e ao impulso foram obtidos de acordo com as funções próprias do matlab, $step$ e $impulse$, respectivamente. A evolução da resposta do sistema é apresentado na Figura 3.



(a) Degrau unitário



(b) Impulso Unitário



(c) Função Senoidal

Figura 3: Resposta do motor CC para entradas do tipo degrau, impulso e senoidal

2.3 Esboço do polo-zero da função de transferência

Utilizando a função $pzplot$ presente no matlab foram verificados os polos da função de transferência utilizada para o sistema. Essa função analisa os valores para os quais a função explode, tendendo ao infinito, ou seja, os valores que fazem o denominador da função ser nulo. O gráfico obtido é apresentado na Figura 4.

Conforme pode ser observado na Figura 4, o sistema apresenta dois polos, um para $s = -2$ e outro para $s = -10$, sendo 5 vezes inferior ao polo menos negativo, que também é o polo dominante do sistema.

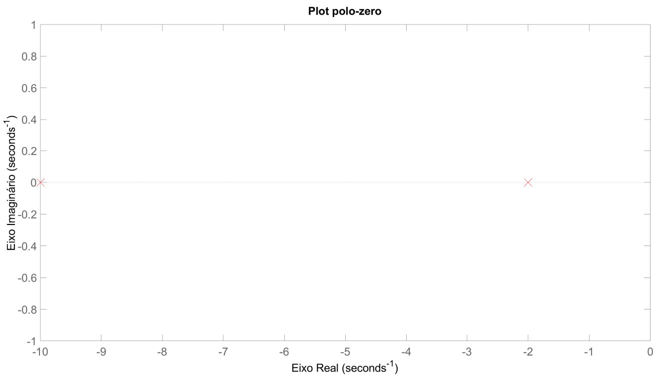


Figura 4: Representação dos polos da função de transferência do sistema

Com intuito de representar de outra forma a função de transferência foi utilizada a função $tf2zp$, que seleciona a função de transferência e analisa tanto o numerador quanto o denominador da função, além de apresentar um ganho a ser multiplicado. De forma geral, o comando $[z, p, k] = tf2zp(b, a)$ gera o resultado dado por:

$$G(s) = \frac{b}{a} = k\,\frac{(s - z_1)(s - z_2)(s - z_3)\dots(s - z_n)}{(s - p_1)(s - p_2)(s - p_3)\dots(s - p_n)}, \tag{3}$$

em que z_n e p_n são os elementos dos vetores z e p, respectivamente. Para de fato montar a função de transferência no formato apresentado acima, se faz necessário utilizar a função $zp2tf$, que tem como entrada as variáveis $[z, p, k]$.

2.4 Aproximação de primeira ordem

Considerando a função de transferência do sistema original e o fato de que os dois polos obtidos são -2 e -10, temos:

$$\frac{K}{(J\,s + b)\,(L\,s + R) + K^2} = \frac{a}{(s + 2)\,(s + 10)}$$
$$K\,(s^2 + 12\,s + 20) = a\,(J\,L\,s^2 + (J\,R + b\,L)\,s + b\,R + K^2)$$

Substituindo os valores, obtemos:

$$s^2 + 12\,s + 20 = a\,\left(\frac{s^2}{2} + 6\,s + 10\right)$$

Assim, chega-se em $a = 1/2$, de tal forma que:

$$\frac{K}{(J\,s + b)\,(L\,s + R) + K^2} = \frac{1}{2\,(s + 2)\,(s + 10)}$$

Como estamos considerando o polo dominante, deseja-se então escrever o sistema por:

$$\frac{1}{2\,(s+2)\,(s+10)} = \frac{\alpha}{(s+2)}$$

Tomando o limite para baixas frequências, temos:

$$\frac{1}{40} = \frac{\alpha}{4},$$

logo temos $\alpha = 0.1$. Assim, a função de transferência de primeira ordem é dada por:

$$G(s) = \frac{0.1}{0.5\,s+1}$$

Com essa função é possível realizar a simulação do comportamento do sistema, como apresentado na Figura 5, onde é comparado também com a resposta original, de segunda ordem.

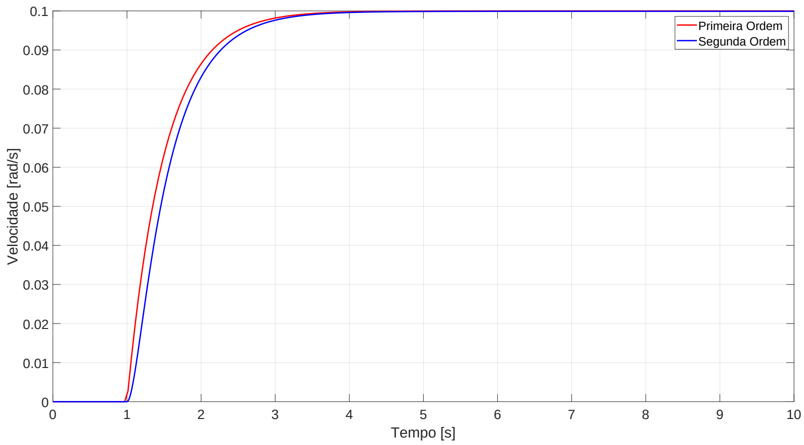


Figura 5: Resposta do sistema para modelos de primeira e segunda ordem

Observa-se uma proximidade entre as funções, com o sistema em primeira ordem sendo mais rápido para chegar até o valor de referência, o que poderia ser esperado, já que parte da elaboração do sistema deixa de ser considerado nesse modelo.