UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e Tecnologia
Departamento de Computação

Controle e Servomecanismo

Tutorial NR.3

Modelagem de velocidade de um motor CC. [1][2]

1 Instruções Gerais

- Grupo de até no máximo 2 alunos e
- Ler atentamente todo o procedimento desse tutorial antes de realizá-lo.

2 Objetivos do Tutorial

- Modelagem de velocidade de um motor CC.
- Obtenção da resposta do sistema por diferentes métodos.
- Resposta do sistema às entradas degrau, impulso e seno.
- Esboço pólo-zero e análise de pólo dominante.
- E aproximação de primeira ordem.

3 Pré-tutorial

1. Considerando a variável dinâmica $\dot{\theta}(t)$ e a entrada $\nu(t)$, demonstre a obtenção da equação de Entrada/Saída (E/S) (3 - 4).
2. Demonstre a obtenção da função de transferência (5).
3. Como funciona as funções `poly`, `syslin` e `csim` do Scilab.

4 Tutorial

Um atuador comum em sistemas de controle é o motor CC. Ele fornece movimento rotativo diretamente e, acoplado a rodas ou tambores e cabos, pode fornecer movimento translacional. O circuito elétrico equivalente da armadura e o diagrama de corpo livre do rotor são mostrados na Figura 1 a seguir.

[1] Documento adaptado de Control Tutorials for MATLAB & Simulink [1]
[2] Revisão 11/05/2023: Prof. Roberto Santos Inoue e Prof. Artino Quintino

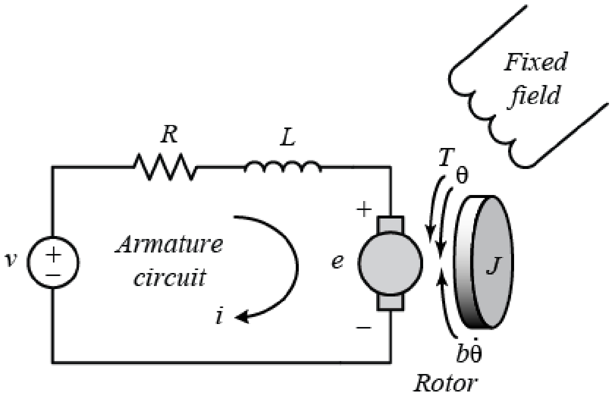

Figura 1: Motor CC.

Para este exemplo, assumiremos que a entrada do sistema é a fonte de tensão (V) aplicada na armadura do motor, enquanto a saída é a velocidade de rotação do eixo $\dot{\theta}$. O rotor e o eixo são considerados rígidos. Assumimos ainda um modelo de atrito viscoso, ou seja, o torque de atrito é proporcional à velocidade angular do eixo.

Os parâmetros físicos para o nosso exemplo são:

- (J) momento de inércia do rotor 0,01 kg.m^2
- (b) constante de atrito viscoso do motor 0,1 N.m.s
- (K_e) constante de força eletromotriz 0,01 V/rad/seg
- (K_t) constante de torque do motor 0,01 N.m/Amp
- (R) resistência elétrica 1 Ohm
- (L) indutância elétrica 0,5 H

4.1 Sistemas de Equações

Em geral, o torque gerado por um motor CC é proporcional à corrente de armadura e à força do campo magnético. Neste exemplo vamos assumir que o campo magnético é constante e, portanto, que o torque do motor é proporcional apenas à corrente de armadura i por um fator constante K_t:

$$T = K_t i. \tag{1}$$

Isso é chamado de motor controlado por armadura.

A fem traseira, e, é proporcional à velocidade angular do eixo por um fator constante K_e.

$$e = K_e \dot{\theta} \tag{2}$$

Em unidades SI, o torque do motor e as constantes de contrafem são iguais, ou seja, $K_t = K_e$; portanto, usaremos K para representar a constante de torque do motor e a constante de fem traseira.

Da Figura 1, podemos derivar as seguintes equações governantes com base na 2ª lei de Newton

$$J\ddot{\theta} + b\dot{\theta} = Ki, \tag{3}$$

e na lei de tensão de Kirchhoff

$$L\frac{di}{dt} + Ri = \nu - K\dot{\theta}. \tag{4}$$

4.2 Resposta do sistema - através de blocos integradores

Deseja-se simular o motor CC através de blocos integradores. Para isso faça:

1. Escreva os parâmetros do motor CC definidos na Seção 4 em um script do Scilab, conforme Código 1. Execute o código para que as variáveis fiquem registrados no "Navegador de Variáveis" do Scilab.

Código 1 Resposta do veículo utilizando a função `ode`.

```
J = 0.01;   // [kg.m^2] (J) moment of inertia of the rotor
b = 0.1;     // [N.m.s] (b)motor viscous friction constant
Ke = 0.01; // [V/rad/sec] (Ke)electromotive force constant
Kt = 0.01; // [N.m/Amp] (Kt)    motor torque constant
R = 1;      // [Ohm] (R) electric resistance
L = 0.5;    // [H] (L) electric inductance
```

2. Realize a simulação do motor CC através das equações (3-4) por blocos integradores conforme diagrama feito no XCOS apresentado na Figura 2. Considere a entrada de tensão do motor como 5V. Para isto, utilize os seguintes blocos:
   (a) Fontes → STEP_FUNCTION,
   (b) Operações matemáticas → BIGSOM_f,
   (c) Operações matemáticas → GAINBLK,
   (d) Sistemas de tempo contínuo → INTEGRAL_f,
   (e) Receptores → CSCOPE e
   (f) Receptores → CLOCK_c.

   No relatório, descrever o função de cada um dos blocos utilizados.

3. Faça as configurações necessárias em Simulação → Configurações e no bloco CSCOPE para que o gráfico de velocidade fique adequado.

4. Envie os dados de saída da simulação do XCOS para o workspace do Scilab. Para isto utilize o bloco "To Workspace" que está na paleta "Receptores". Agora utilize o plot para fazer o gráfico da saída de velocidade.

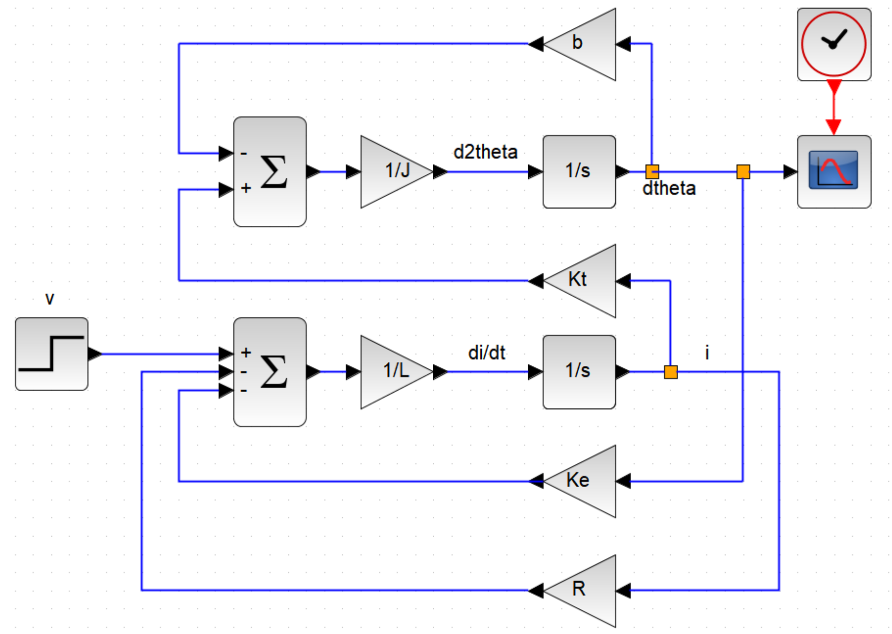

Figura 2: Diagrama de blocos do sistema do veículo por blocos integradores.

4.3 Resposta do sistema - através da função de transferência

A função de transferência $G(s)$ do motor CC para controle velocidade obtida a partir da equações (3-4), é dada por:

$$G(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js+b)(Ls+R)+K^2} \left[\frac{\text{rad/seg}}{\mathbf{V}}\right]. \tag{5}$$

Pede-se:

1. Obtenha a resposta ao degrau unitário. Utilize o Código 2 para isto.
2. Obtenha a resposta ao impulso. Utilize o Código 3 para isto. Mude o degrau para uma amplitude de 5.
3. Obtenha a resposta ao seno. Utilize o Código 4 para isto. Varie a amplitude e a velocidade angular do seno.

Código 2 Resposta ao degrau.

```
clear
clc
xdel(winsid()) // Fecha todas as telas de plot
// Parametros do motor
J = 0.01;  // [kg.m^2] (J) moment of inertia of the rotor
b= 0.1;    // [N.m.s] (b)motor viscous friction constant
Ke = 0.01; // [V/rad/sec] (Ke)electromotive force constant
Kt = 0.01; // [N.m/Amp] (Kt)    motor torque constant
K = 0.01;
R = 1;     // [Ohm] (R) electric resistance
L = 0.5;   // [H] (L) electric inductance

// Funcao de transferencia
s = poly(0,'s')
G = syslin ('c',K/((J*s+b)*(L*s+R)+K^2) )
// Horizonte de tempo da simulacao
t = 0:0.1:8;
// Resposta step
dtheta = csim('step',t,G)
// Graficos
plot2d(t,dtheta,rect = [0,0,8,0.11])
xlabel('Tempo[s]')
ylabel('Velocidade Angular [ rad / s ]')
xgrid(5, 1, 7)
```

Código 3 Resposta ao impulso.

```
// Resposta impulso
dtheta = csim('impulse',t,G)
// Graficos
scf()
plot2d(t,dtheta,rect = [0,0,8,0.15])
xlabel('Tempo[s]')
ylabel('Velocidade Angular [ rad / s ]')
xgrid(5, 1, 7)
```

Código 4 Resposta ao seno.

```
// Resposta seno
w= 1
A = 1.5
deff('u=timefun(t,w,A)','u =A*sin(w*t)')
dtheta = csim(timefun,t,G)
// Graficos
scf()
plot(t,dtheta,rect = [0,-.15,8,0.15])
xlabel('Tempo[s]')
ylabel('Velocidade Angular [ rad / s ]')
xgrid(5, 1, 7)
```

4.4 Resposta do sistema - através de bloco de função de transferência

Também é possível realizar a simulação do sistem utilizando blocos de função de transferência.

Desse modo, pede-se:

1. Realize a simulação da função de transferência dada pela Equação (5) utilizando o XCOS. Para isto,
   (a) Utilize o bloco: Sistemas de tempo contínuo → CLR.
   (b) Obtenha a resposta ao degrau unitário e ao impulso. Compare as duas respostas e a entrada ao degrau no mesmo gráfico.
   (c) Relacione a solução geral e particular de uma EDO com as respostas encontradas. Qual resposta e solução está relacionada com a resposta transitória e com a resposta em regime permanente?

4.5 Esboço do polo-zero das função de transferência

1. Obtenha um esboço de pólo-zero da função $G(s)$ da pela Equação (5). Utilize para isto o comando `plzr(G)`.
2. O pólo mais negativo está quantas vezes mais distante do pólo próximo mais próximo do eixo 0? 5 vezes? Qual é o pólo mais lento? Qual é o pólo dominante?
3. Obtenhas os valores de pólo-zeros e o ganho da função de transferência através do comando `[z, p, k] = tf2zp(G)`.
4. Gere uma nova função de transferência $H(s)$ semelhante a $G(s)$ (5) através pólos e zeros da função de transferência. Utilize para isto o comando `H = zp2tf(z,p,k,"c")`.
5. Explique o funcionamento das funções utilizadas no relatório.

4.6 Aproximação de primeira ordem

Na Seção 4.5 observou-se que o pólo dominante é "-2". Desse modo, pede-se:

1. Demostre que o modelo equivalente de primeira ordem pode ser dado por:

$$G(s) = \frac{0.1}{0.5s + 1}. \tag{6}$$

2. Compare a resposta ao degrau unitário do sistema de primeira ordem com o sistema de segunda ordem.

Referências Bibliográficas

[1] Dawn Tilbury, Bill Messner, Rick Hill, JD Taylor, and Shuvra Das. Control tutorials for MATLAB & Simulink. Technical report, 2021.