Pontifícia Universidade Católica de Goiás
Escola de Engenharia
ENG 3503 – Sistemas de Controle
Prof: Marcos Lajovic Carneiro
Aluno (a):____

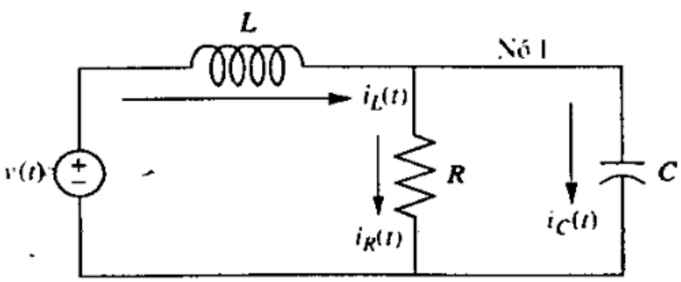
Espaço dos Estados

Laboratório Resumo
Experimentos da Modelagem no Espaço dos Estados

1 – Modelagem no Espaço dos Estados no MATLAB (resumo cap.3)

1.1 – Representação no espaço dos estados

Considere o circuito abaixo:



Passo 1 Nomear as correntes de todos os ramos do circuito i_L, i_R e i_C.

Passo 2 Selecionar as variáveis de estado escrevendo a equação da derivada relativa a todos os elementos armazenadores de energia.

$$C\frac{dv_C}{dt} = i_C \qquad L\frac{di_L}{dt} = v_L$$

Variáveis de Estado: v_C e i_L

Passo 3 Aplicar a teoria de circuitos, como as leis de Kirchhoff das tensões e das correntes, para obter i_c e v_L em termos das variáveis de estado, v_c e i_L

Equação no nó 1:

$$i_C = -i_R + i_L = -\frac{1}{R}v_C + i_L$$

Equação na malha externa:

$$-v(t) + v_L + v_C = 0$$
$$v_L = -v_C + v(t)$$

Passo 4 Obter as equações de estado:

$$C\frac{dv_C}{dt} = i_C = -\frac{1}{R}v_C + i_L$$

$$L\frac{di_L}{dt} = v_L = -v_C + v(t)$$

Equações de estado

$$\frac{dv_C}{dt} = -\frac{1}{RC}v_C - \frac{1}{C}i_L$$

$$\frac{di_L}{dt} = -\frac{1}{L}v_C \qquad + \frac{1}{L}v(t)$$

Passo 5 Obter a equação de saída:

$$i_R = \frac{1}{R}v_C$$

Passo 6 Obter representação no espaço dos estados de forma matricial:

$$\begin{cases} \dfrac{dv_C}{dt} = -\dfrac{1}{RC}v_C - \dfrac{1}{C}i_L \\ \dfrac{di_L}{dt} = -\dfrac{1}{L}v_C \qquad + \dfrac{1}{L}v(t) \\ i_R = \dfrac{1}{R}v_C \end{cases}$$

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -1/(RC) & -1/C \\ -1/L & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} v(t)$$

$$i_R = [1/R \quad 0] \begin{bmatrix} v_C \\ i_L \end{bmatrix}$$

O modelo do circuito acima obedece a representação geral de um sistema com uma entrada v(t), saída única y=i_R e duas variáveis de estado $x_1 = v_C$ e $x_2 = i_L$:

Equações de estado:

$$\frac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + b_1v(t)$$

$$\frac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2 + b_2v(t)$$

Equação da saída:

$$y = c_1x_1 + c_2x_2 + d_1v(t)$$

Organização na forma de matrizes:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

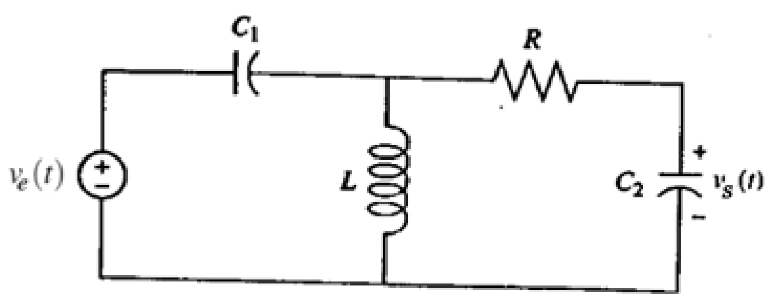
$$C = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \qquad D = d_1$$

1.2 – Representação no espaço dos estados no MATLAB

A representação no espaço de estados consiste na especificação das matrizes A, B, C e D seguida da criação de um objeto espaço de estados LTI utilizando o comando MATLAB, SS(A, B, C, D).

```
A=[0 1 0;0 0 1;-9 -8 -7];           % Representa A.
B=[7;8;9];                          % Representa o vetor coluna B.
C=[2 3 4];                          % Representa o vetor linha C.
D=0;                                % Representa D.
F=ss(A,B,C,D)                       % Cria um objeto LTI e exibe.
pause
```

Questão 1) Considere o circuito abaixo juntamente com sua representação no Espaço dos Estados:



Componentes:
$C_1 = 1F$
$C_2 = 2F$
$L = 5H$
$R = 3\Omega$

Equações de estado

$$\frac{dv_{C_1}}{dt} = -\frac{1}{RC_1}v_{C_1} + \frac{1}{C_1}i_L - \frac{1}{RC_1}v_{C_2} + \frac{1}{RC_1}v_e$$

$$\frac{di_L}{dt} = -\frac{1}{L}v_{C_1} + \frac{1}{L}v_e$$

$$\frac{dv_{C_2}}{dt} = -\frac{1}{RC_2}v_{C_1} - \frac{1}{RC_2}v_{C_2} - \frac{1}{RC_2}v_e$$

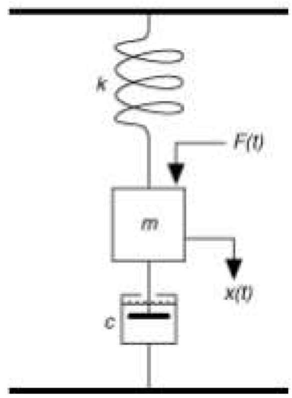
Equação da saída:

$$v_s = v_{C_2}$$

Converta o sistema de equações em uma representação matricial com as matrizes A, B, C e D conforme apresentado na teoria acima. Crie um objeto LTI com a representação no espaço dos estados no MATLAB utilizando os comandos apresentados.

2 – Plotando a saída de um sistema modelado no Espaço dos Estados no MATLAB (resumo cap.4)

Considere o sistema massa, mola e amortecedor abaixo:



Definimos o sistema no seguinte formato de espaço dos estados:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Assumindo que o sinal de controle é igual a força e que apenas medimos a posição. O modelo de espaço de estados para o sistema torna-se então:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

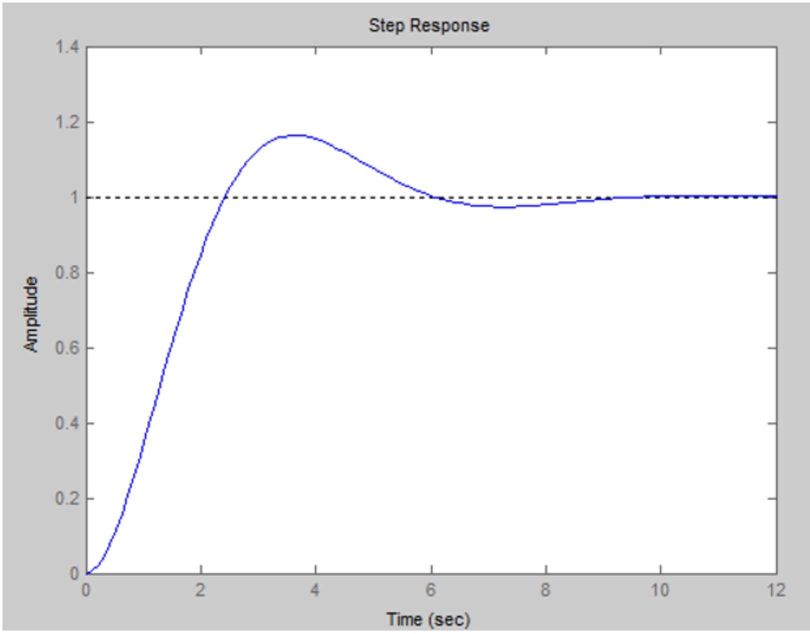
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Observa-se na representação acima as matrizes A, B e C (matriz D=0). Após criar a representação no espaço dos estados internamente no MATLAB através do comando:

```
ss_model = ss(A,B,C,D)      %Equação Espaço do Estado
```

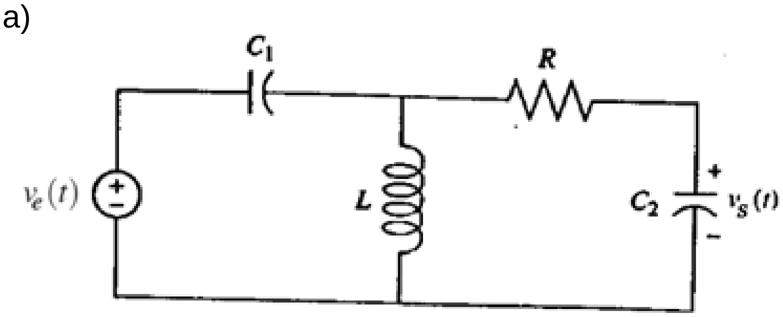
para visualizarmos a resposta no tempo utilizamos o comando abaixo:

```
ss_model = ss(A,B,C,D)      %Equação Espaço do Estado
step(ss_model)              %Entrada a degrau
```



Questão 2) Plote a resposta no tempo para os sistemas abaixo:

a)



<u>Componentes:</u>

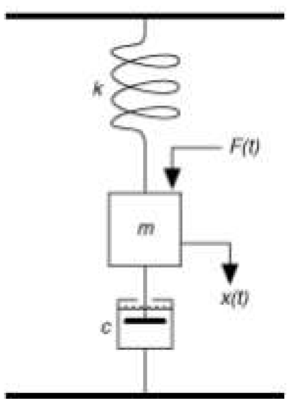
$C_1 = 1F$

$C_2 = 2F$

$L = 5H$

$R = 3\Omega$

b)



<u>Componentes:</u>

C=2 N.s/m

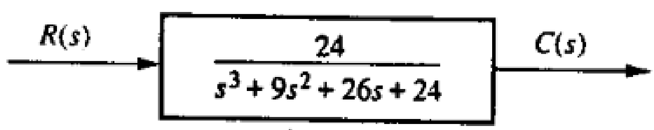
K=1 N/m

M=3 kg

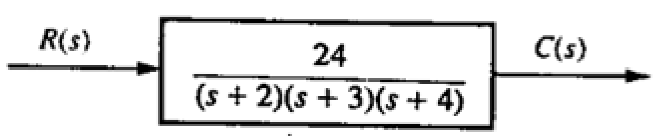
3 – Convertendo funções de transferência para diferentes formas de representação no estados (resumo cap.5)

Uma mesma função de transferência pode ser representada por matrizes no espaço dos estados de diferentes formas sem que o resultado se altere. As diferentes formas de representação podem facilitar a modelagem e a solução de problemas de aplicações específicas.

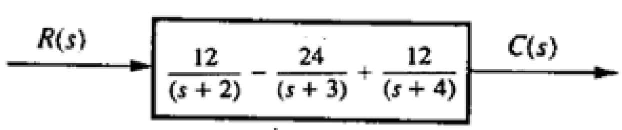
Observe a função de transferência do sistema abaixo e suas matrizes correspondentes no espaço dos estados:

Representação por função de transferência	Representação no Espaço dos Estados por **variáveis de fase**
	$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -24 & -26 & -9\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 24\end{bmatrix} r$ $y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$

Fatorando o denominador da equação acima obtemos uma equação equivalente, ou seja, a apresentação muda mas seus resultados são exatamente os mesmos:

Representação por função de transferência	Representação no Espaço dos Estados em **cascata**
	$\dot{\mathbf{x}} = \begin{bmatrix}-4 & 1 & 0\\ 0 & -3 & 1\\ 0 & 0 & -2\end{bmatrix}\mathbf{x} + \begin{bmatrix}0\\ 0\\ 24\end{bmatrix} r$ $y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\mathbf{x}$

Aplicando frações parciais nesta mesma equação obtemos outra representação equivalente que pode ser vista de forma evidente no formato de equação, contudo de forma não evidente matricialmente:

Representação por função de transferência	Representação no Espaço dos Estados em **paralelo**
	$\dot{\mathbf{x}} = \begin{bmatrix}-2 & 0 & 0\\ 0 & -3 & 0\\ 0 & 0 & -4\end{bmatrix}\mathbf{x} + \begin{bmatrix}12\\ -24\\ 12\end{bmatrix} r$ $y = \begin{bmatrix}1 & 1 & 1\end{bmatrix}\mathbf{x}$

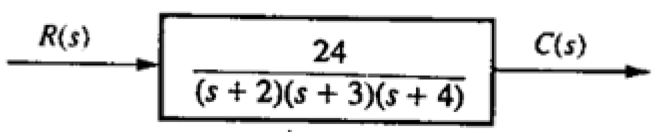
A seguinte função de transferência é representada com outros dois formatos importantes para sistemas de controle:

$$G(s) = \frac{C(s)}{R(s)} = \frac{s^2 + 7s + 2}{s^3 + 9s^2 + 26s + 24}$$

Forma Canônica do Controlador	Forma Canônica do Observador
$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -9 & -26 & -24 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} r$ $y = \begin{bmatrix} 1 & 7 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$	$\dot{\mathbf{x}} = \begin{bmatrix} -9 & 1 & 0 \\ -26 & 0 & 1 \\ -24 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 7 \\ 2 \end{bmatrix} r$ $y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}$

É possível converter uma função de transferência para a representação no espaço dos estados através de comandos no MATLAB e em seguida transformar um tipo de representação matricial em outra através dos comandos a seguir:

Representando a função de transferência a seguir:



<u>Comandos:</u>

```
numt=24;                           % Define o numerador de T(s).
dent=poly([-2 -3 -4]);             % Define o denominador de T(s).
'T(s)'                             % Exibe o título.
T=tf(numt,dent)                    % Cria e exibe T(s).
```

Convertendo para matrizes de estado e em seguida criando o objeto LTI que representa internamente no MATLAB o sistema de controle na forma canônica do controlador:

```
[Acc Bcc Ccc Dcc]=tf2ss(numt,dent); % Converte T(s) para
                                    % a forma canônica controlável.
Scc=ss(Acc,Bcc,Ccc,Dcc);            % Cria objeto espaço de estados
                                    % LIT canônico controlável.
```

Convertendo a forma canônica do controlador para a forma paralela:

```
Sp=canon(Scc,'modal');              % Converte a forma canônica
                                    % controlável para a forma paralela.
```

Apresenta as matrizes da forma canônica do controlador e da forma paralela:

```
'Forma Canônica Controlável'          % Exibe o título.
[Acc,Bcc,Ccc,Dcc]=ssdata(Scc)         % Extrai e exibe as matrizes
                                      % da forma canônica controlável.
'Forma paralela'                      % Exibe o título.
[Ap,Bp,Cp,Dp]=ssdata(Sp)              % Extrai e exibe as matrizes da
                                      % forma paralela.
```

Questão 3) Converta a função de transferência a seguir para a representação no espaço dos estados na forma canônica do controlador e na forma paralela:

$$G(s) = \frac{C(s)}{R(s)} = \frac{s^2 + 7s + 2}{s^3 + 9s^2 + 26s + 24}$$

4 – Teste de estabilidade de um sistema no Espaço dos Estados (resumo cap.6)

Podemos verificar se um sistema é instável pela existência de pólos no semi plano s da direta. Os pólos de um sistema representado no espaço dos estados pode ser obtido através da matriz A do sistema através dos comandos a seguir:

```
A=[0 3 1;2 8 1;-10 -5 -2]             % Define a matriz de sistema, A.
autovalores=eig(A)                    % Obtém os autovalores.
```

Obtemos como resultado do sistema acima a resposta:

autovalores =

-0.8821 + 2.4330i
-0.8821 - 2.4330i
7.7642 + 0.0000i

Os autovalores da matriz são equivalentes aos pólos do sistema. Observa-se um pólo no semi plano da direita em 7.76, portanto esse sistema é instável.

Questão 4) Verifique a estabilidade do sistema abaixo:

$$\dot{\mathbf{x}} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 7 & 1 \\ -3 & 4 & -5 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

$$y = [0 \quad 1 \quad 0]\mathbf{x}$$

5 – Compensando um sistema modelado através do Espaço dos Estados (resumo cap.12)

Podemos usar o MATLAB para projetar ganhos do controlador usando alocação de polos. Você irá entrar a ultrapassagem percentual e o tempo de acomodação desejados. Introduzimos os seguintes comandos: [num, den]=ord2 (wn,z), que produz um sistema de segunda ordem, dados a frequência natural (wn) e o fator de amortecimento (z). Em seguida, utilizamos o denominador (den) para especificar os polos dominantes; e K=acker(A,B, –polos), que calcula os ganhos do controlador a partir da matriz de sistema (A), da matriz de entrada (B) e dos polos desejados (polos).

O código abaixo projeta os ganhos de retroação das variáveis de fase para a função de transferência dada. O programa compensa o sistema levando a uma ultrapassagem percentual e um tempo de assentamento desejado. Execute o bloco de código abaixo para uma ultrapassagem percentual de 9,5% e um tempo de assentamento de 0,74s.

$$G(s) = \frac{20(s+5)}{s(s+1)(s+4)}$$

```
numg=20*[1 5];                        % Define o numerador de G(s).
deng=poly([0 -1 -4]);                 % Define o denominador de G(s).
'G(s) sem compensação'                % Exibe o título.
G=tf(numg,deng)                       % Cria e exibe G(s).
pup=input('Digite %UP desejada ');    % Entra a ultrapassagem
                                      % percentual desejada.
Ts=input('Digite o tempo de acomodação desejado ');
                                      % Entra o tempo de acomodação desejado.
z=(-log(pup/100))/(sqrt(pi^2+log(pup/100)^2));
                                      % Calcula o fator de amortecimento
                                      % requerido.
wn=4/(z*Ts);                          % Calcula a frequência natural
                                      % requerida.
[num,den]=ord2(wn,z);                 % Produz um sistema de
                                      % segunda ordem que atende aos
                                      % requisitos de resposta transitória.
r=roots(den);                         % Usa o denominador para
                                      % especificar os polos dominantes.
polos=[r(1) r(2) -5.1];               % Especifica a posição de
                                      % todos os polos.
eq_caracteristica_desejada=poly(polos)
                                      % Cria o polinômio característico
                                      % desejado para exibir na tela.
[Ac Bc Cc Dc]=tf2ss(numg,deng);       % Obtém a forma canônica
                                      % controlável da representação
                                      % no espaço de estados de G(s).
P=[0 0 1;0 1 0;1 0 0];                % Matriz de transformação
                                      % da forma canônica controlável
                                      % para a forma de variáveis de fase.
Ap=inv(P)*Ac*P;                       % Transforma Ac para a
                                      % forma de variáveis de fase.
Bp=inv(P)*Bc;                         % Transforma Bc para a
```

```
                                          % forma de variáveis de fase.
Cp=Cc*P;                                  % Transforma Cc para a
                                          % forma de variáveis de fase.
Dp=Dc;                                    % Transforma Dc para a
                                          % forma de variáveis de fase.
Kp=acker(Ap,Bp,polos)                     % Calcula os ganhos do controlador
                                          % na forma de variáveis de fase.
Apnew=Ap-Bp*Kp;                           % Cria a matriz A compensada.
Bpnew=Bp;                                 % Cria a matriz B compensada.
Cpnew=Cp;                                 % Cria a matriz C compensada.
Dpnew=Dp;                                 % Cria a matriz D compensada.
[numt,dent]=ss2tf(Apnew,Bpnew,Cpnew,Dpnew);
                                          % Cria o numerador e o denominador
                                          % de T(s).
'T(s)'                                    % Exibe o título.
T=tf(numt,dent)                           % Cria e exibe T(s).
polos=roots(dent)                         % Exibe os polos de T(s).
Tee=ss(Apnew,Bpnew,Cpnew,Dpnew)           % Cria e exibe Tee, um
                                          % objeto espaço de estados LTI.
step(Tee)                                 % Produz a resposta ao degrau
                                          % compensada.
title('Resposta ao Degrau Compensada')
                                          % Adiciona um título à resposta
                                          % ao degrau compensada.
```

Bibliografia:

NISE, N. S. **Engenharia de sistemas de controle**. 5.ed. Rio de Janeiro: LTC, 2010.