UNIVERSIDADE FEDERAL DE MINAS GERAIS

OFICINA DE MODELAGEM E SIMULAÇÃO

Lista de Exercícios 10

Marcone Márcio da Silva Faria

---

1. **1. Dada a equação diferencial de 3ª ordem:**

$$x''' + x'' + 2x' + x = 2f(t)$$

Representa-a no espaço de estados e em função de transferência considerando a saída y = x. Dica: utilize o comando `ss2tf`**.**

```
A = [0 1 0; 0 0 1; -1 -2 -1];
B = [0; 0; 2];
C = [1 0 0];
D = [0];

espaco_estados = ss(A,B,C,D)
[num, den] = ss2tf(A,B,C,D);
G = tf(num, den)

>> lista10
    espaco_estados =

      A =
           x1   x2   x3
       x1   0    1    0
       x2   0    0    1
       x3  -1   -2   -1

      B =
           u1
       x1   0
       x2   0
       x3   2

      C =
           x1   x2   x3
       y1   1    0    0

      D =
           u1
       y1   0

    Continuous-time state-space model.

    G =
               2
      -------------------
      s^3 + s^2 + 2 s + 1
    Continuous-time transfer function.
```

2. **Dado o sistema mecânico rotacional da figura 3, cujo modelo matemático é dado pelo sistema de equações diferenciais**

$$J_1\ddot{\theta}_1(t) + (k_1 + k_2)\theta_1 - k_2\theta_2 = T$$
$$J_2\ddot{\theta}_2(t) - k_2\theta_1 + (k_2 + k_3)\theta_2 = 0$$

onde J1 = 1 kg.m2 , J2 = 2 kg.m2 , k1 = 1, k2 = 2 e k3 = 3. Pedem-se:

a) **a representação no espaço de estados e em função de transferência, sendo T(t) a entrada e θ1(t) a saída;**

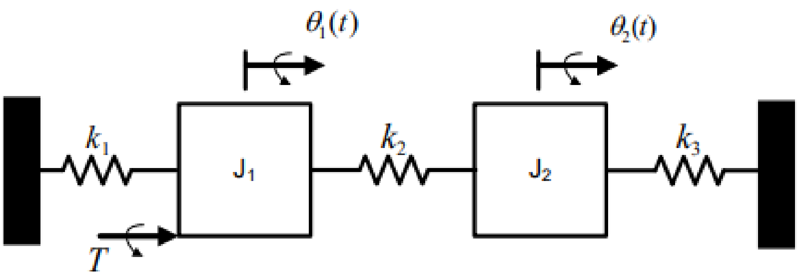


Figura 3: Sistema Massa-Mola.

```
J1 = 1;
J2 = 2;
k1 = 1;
k2 = 2;
k3 = 3;

A = [0 1 0 0; [-(k1+k2) 0 k2 0]./J1; 0 0 0 1; [k2 0 -(k2+k3)
0]./J2];
B = [0; 1/J1; 0; 0];
C = [1 0 0 0];
D = [0];

espaco_estados = ss(A,B,C,D)
[num, den] = ss2tf(A,B,C,D);
G = tf(num,den)

>> lista10
    espaco_estados =

      A =
             x1     x2     x3     x4
       x1     0      1      0      0
       x2    -3      0      2      0
       x3     0      0      0      1
       x4     1      0   -2.5      0

      B =
            u1
       x1    0
       x2    1
       x3    0
       x4    0
```

```
    C =
         x1  x2  x3  x4
     y1   1   0   0   0

    D =
         u1
     y1   0


  G =
                          s^2 + 2.5
    --------------------------------------------------
    s^4 + 3.331e-16 s^3 + 5.5 s^2 + 7.565e-16 s + 5.5
```

b) a representação no espaço de estados e matriz de transferência, sendo T(t) a entrada e θ1(t) e θ2(t) as saídas.

```
C = [1 0 0 0; 0 0 1 0];
D = [0;0];

espaco_estados = ss(A,B,C,D)
[num, den] = ss2tf(A,B,C,D);

G1 = tf(num(1,1:end),den)
G2 = tf(num(2,1:end),den)

>> lista10
   espaco_estados =

     A =
            x1     x2     x3     x4
      x1     0      1      0      0
      x2    -3      0      2      0
      x3     0      0      0      1
      x4     1      0   -2.5      0

     B =
          u1
      x1   0
      x2   1
      x3   0
      x4   0

     C =
          x1  x2  x3  x4
      y1   1   0   0   0
      y2   0   0   1   0

     D =
          u1
      y1   0
      y2   0


   G1 =

                          s^2 + 2.5
     --------------------------------------------------
     s^4 + 3.331e-16 s^3 + 5.5 s^2 + 7.565e-16 s + 5.5
```

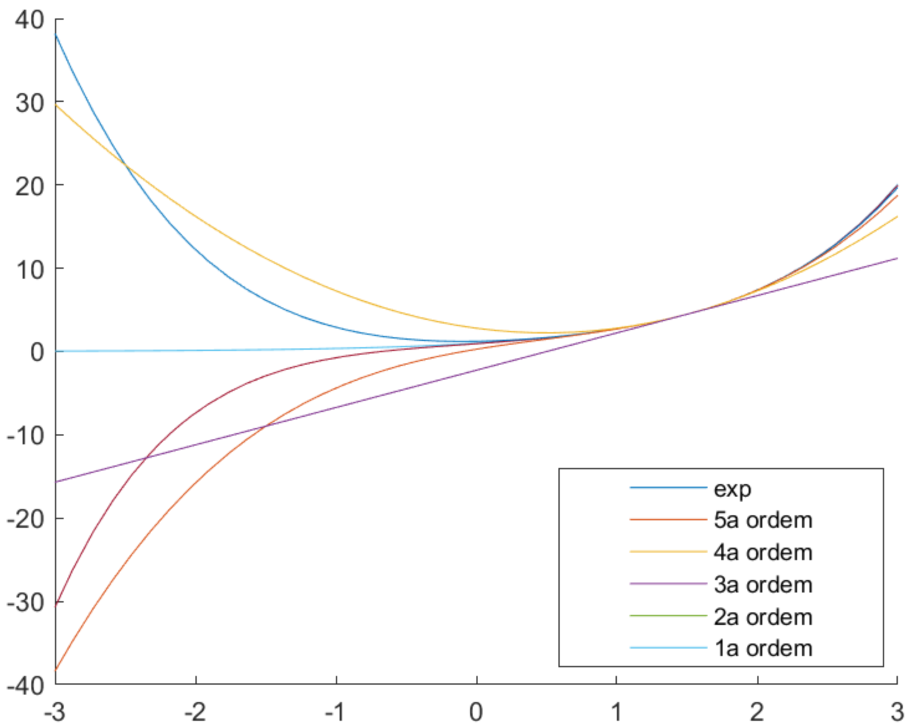
```
G2 =

                         1
  ------------------------------------------------
  s^4 + 3.331e-16 s^3 + 5.5 s^2 + 7.565e-16 s + 5.5
```

3. **Linearize a equação não linear z = xy no intervalo 5 ≤ x ≤ 7 e 10 ≤ y ≤ 12. Encontre o erro de aproximação quando x = 5 e y = 10.**

```
syms x y;
z = x * y;
lin(x, y) = taylor(z, [x,y], [6,11], 'Order',2)

erro = abs(lin(5,10) - 50)/50;
erro = double(erro)

>> lista10

    lin(x, y) =
         11*x + 6*y - 66

    erro =
         0.0200
```

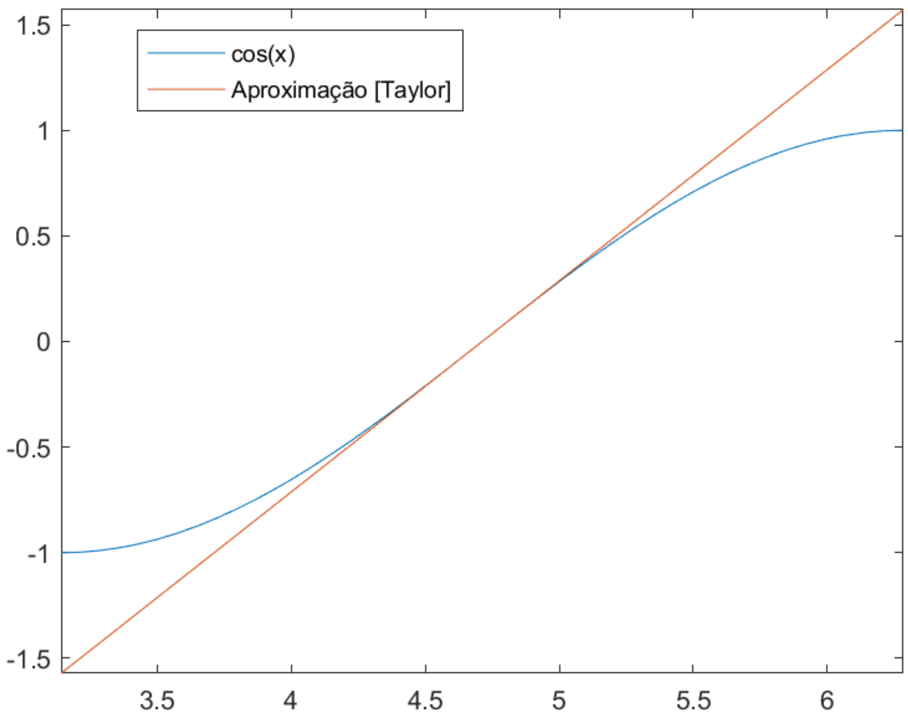
4. **Defina uma variável simbólica x. Encontre a expansão em série de Taylor de f(x) = e x em torno do ponto 1.5. Construa apenas um gráfico contendo seis funções. São elas: a função original f(x) e as séries de Taylor truncadas até os termos de 5ª, 4ª, ..., 1ª ordem. Comente o ocorrido. PS.: Escolha a escala do eixo das abscissas entre −3 e 3 para uma melhor visualização do gráfico.**

```
f1 = exp(x);
y5 = taylor(f1, x, 1.5,'Order', 6);
y4 = taylor(f1, x, 1.5,'Order', 5);
y3 = taylor(f1, x, 1.5,'Order', 4);
y2 = taylor(f1, x, 1.5,'Order', 3);
y1 = taylor(f1, x, 1.5,'Order', 2);
ys = [f1 y5 y4 y3 y2 y1];

hold on
   for a = ys
       fplot(a,[-3 3]);
   end
hold off;

legend('exp','5a ordem','4a ordem', ...
'3a ordem','2a ordem','1a ordem')
legend("Position",[0.5,0.2,0.3,0.2])
```

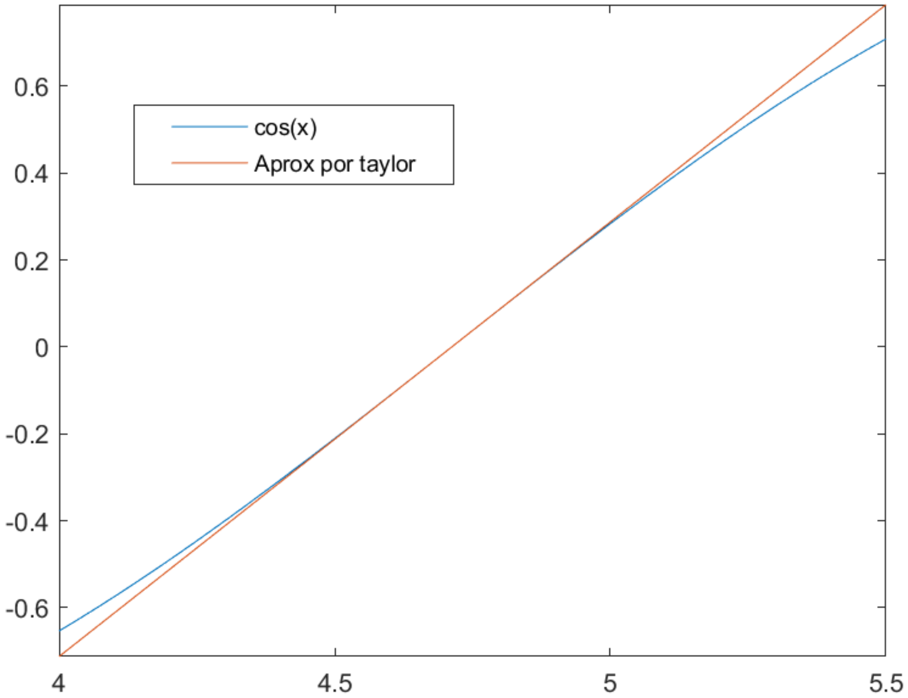


5. **Escolha uma função senoidal. Obtenha uma aproximação linear em torno de um ponto qualquer usando a série de Taylor. Faça as suposições necessárias. Mostre as funções original e aproximada na janela de comando e em gráfico.**

```
f(x) = cos(x)
T(x) = taylor(f, x, 3*pi/2,'Order', 2)

fplot(f, [pi 2*pi])
hold on
fplot(T, [pi 2*pi])
hold off

legend('cos(x)', 'Aproximação [Taylor]');
legend("Position", [0.2,0.8,0.3,0.1]);

erro_lim_superior = (cos(pi) - double(T(pi)))/cos(pi)
erro_lim_inferior = (cos(2*pi) - double(T(2*pi)))/cos(2*pi)

>> lista10

    f(x) =
    cos(x)


    T(x) =
    x - (3*pi)/2


    erro_lim_superior =
       -0.5708


    erro_lim_inferior =
       -0.5708
```




```
fplot(f,[4 5.5])

hold on
fplot(T,[4 5.5])
hold off

legend('cos(x)','Aprox por taylor');
legend("Position",[0.2,0.7,0.3,0.1]);

erro_lim_superior = (cos(5.5) - double(T(5.5)))/cos(5.5)
erro_lim_inferior = (cos(4) - double(T(4)))/cos(4),

>> lista10
    f(x) =
         cos(x)
    T(x) =
         x - (3*pi)/2

    erro_lim_superior =
       -0.1114

    erro_lim_inferior =
       -0.0899
```



6. **Dado o modelo não linear que descreve o comportamento cinemático de um robô móvel diferencial apresentado na Figura 4:**

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$$

$$\dot{x} = v \cdot \cos(\theta)$$
$$\dot{y} = v \cdot \operatorname{sen}(\theta)$$
$$\dot{\theta} = \omega$$

Obtenha o modelo linearizado em torno do ponto de equilíbrio (x0, y0, θ0, v0, ω0). Comandos úteis: `ss`, `ss2tf`, `tf2ss`, `zp2tf`, `zp2ss`, `ss2zp`, `taylor`, `eig`

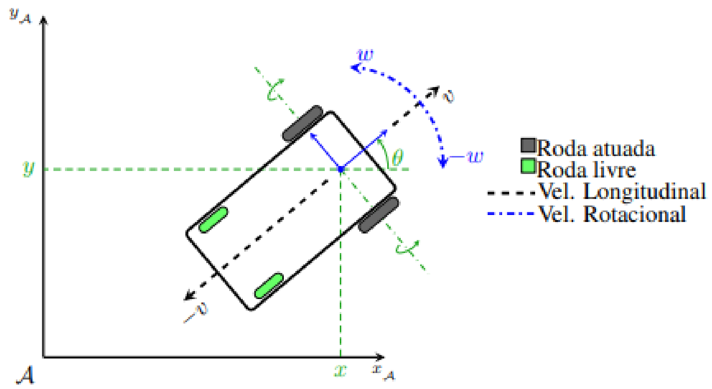


Figura 4: Diagrama de um robô móvel com acionamento diferencial.