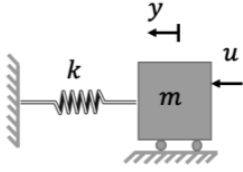


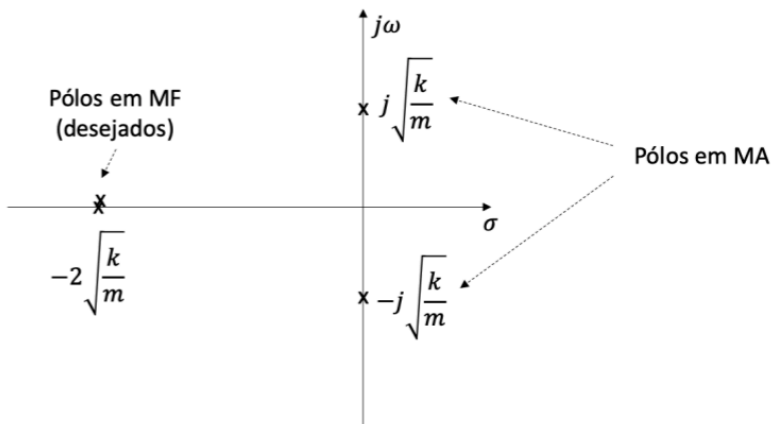
Prova 02 - parte A

Seja o sistema abaixo



utilizado em alguns dos exemplos de sala de aula (ver p. 117, por exemplo). Considere a saída como a posição y e a entrada como a força u . Para seus desenvolvimentos, utilize $m = 2$ kg; $k = 1$ N/m.

Obtenha o conjunto servossistema/observador (com integrador no ramo de malha aberta (MA)) como descrito na Fig. 8.17 das notas de aula, para o requisito abaixo:



Ou seja, deseja-se que os dois pólos dominantes de malha fechada (MF) estejam no eixo real do plano complexo em $-2\sqrt{k/m}$. Para o seu projeto, verifique:

- a estabilidade do conjunto servossistema/observador;
- a aderência ao requisito de projeto, pela simulação para referência do tipo degrau de 10 cm;
- a amplitude de u deve ter módulo inferior a 0,15 N para uma referência de 10 cm para y (altere os requisitos dos pólos dominantes, se necessário);
- a rejeição a distúrbio de carga do tipo degrau de amplitude 0,1 N.

Sugestão de solução: forneça os desenvolvimentos para o projeto usando o livescript do MATLAB (com os devidos comentários/análises diretamente no código) e um arquivo PDF com as simulações requeridas feitas em Simulink.

Resolução:

Inicia-se o projeto pela modelagem do sistema:

$$k = 1, m = 2$$

$$\begin{aligned} k &= 1 \\ m &= 2 \\ x_0 &= 2 \times 1 \\ &0.3000 \end{aligned}$$

0

```
A = [0 1; -k/m 0], B = [0;1/m]
```

```
A = 2x2
      0      1.0000
     -0.5000      0
B = 2x1
      0
      0.5000
```

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

```
C = 1x2
      1      0
D = 0
```

```
sysSS = ss(A,B,C,D)
```

sysSS =

```
A =
      x1  x2
x1      0      1
x2  -0.5      0
```

```
B =
      u1
x1      0
x2  0.5
```

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

```
D =
      u1
y1      0
```

Continuous-time state-space model.

```
% step(sysSS)
damp(sysSS)
```

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
$0.00e+00 + 7.07e-01i$	$0.00e+00$	$7.07e-01$	Inf
$0.00e+00 - 7.07e-01i$	$0.00e+00$	$7.07e-01$	Inf

```
rank(ctrb(A,B))
```

```
ans = 2
```

```
rank(observ(A,C))
```

```
ans = 2
```

```
Mc = [A B; -C 0]
```

```
Mc = 3x3
```

```

    0    1.0000    0
-0.5000    0    0.5000
-1.0000    0    0

```

```
rank(Mc)
```

```
ans = 3
```

```

sig = 1*sqrt(k/m);
P = [- sig*ones(2,1);
     -5*sig] % incluir o requisito

```

```

P = 3x1
   -0.7071
   -0.7071
  -3.5355

```

```
Pe = [-5*sig*ones(2,1)] % incluir o requisito
```

```

Pe = 2x1
  -3.5355
  -3.5355

```

```

Kaux = acker([A zeros(2,1);-C 0],[B;0],P);
K = Kaux(1:end-1)

```

```

K = 1x2
  10.0000    9.8995

```

```
Ki = -Kaux(end)
```

```
Ki = 3.5355
```

```
Ke = acker(A',C',Pe)'
```

```

Ke = 2x1
   7.0711
  12.0000

```

```

At = [A-Ke*C-B*K B*Ki; zeros(1,2) 0];
Bt = [zeros(2,1) Ke;1 -1];
Ct = [-K Ki];
Dt = zeros(1,2);
Gc = zpk(ss(At,Bt,Ct,Dt))

```

```
Gc =
```

```

From input 1 to output:
  3.5355 (s+3.536)^2
-----

```

```
s (s^2 + 12.02s + 52.5)
```

```

From input 2 to output:
-193.04 (s^2 + 0.5698s + 0.2289)
-----

```

```
s (s^2 + 12.02s + 52.5)
```

```
Continuous-time zero/pole/gain model.
```

```
eig(At) % autovalores da matriz de estado
```

```
ans = 3x1 complex
-6.0104 + 4.0466i
-6.0104 - 4.0466i
0.0000 + 0.0000i
```

```
zpk(Gc) % organização em zero/polo/ganho
```

```
ans =

From input 1 to output:
  3.5355 (s+3.536)^2
-----
s (s^2 + 12.02s + 52.5)

From input 2 to output:
-193.04 (s^2 + 0.5698s + 0.2289)
-----
s (s^2 + 12.02s + 52.5)
```

Continuous-time zero/pole/gain model.

```
damp(Gc) % mostra as partes reais dos pólos
```

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
0.00e+00	-1.00e+00	0.00e+00	Inf
-6.01e+00 + 4.05e+00i	8.30e-01	7.25e+00	1.66e-01
-6.01e+00 - 4.05e+00i	8.30e-01	7.25e+00	1.66e-01

```
s=tf('s')
```

```
s =
s
```

Continuous-time transfer function.

```
Greq = 0.5/(s+sig)^2
```

```
Greq =
      0.5
-----
s^2 + 1.414 s + 0.5
```

Continuous-time transfer function.