Experiência 1
Prof. Marconi Kolm Madrid
EA772 - 2017/2

Danilo Pereira Titato - RA 122541
Giovani Granzotto Oliani - RA 146253
Pedro Gabriel Calixto Mendonça - RA 118363

1. Problema do servo

$$G'_{p_s} := \frac{X_1(s)}{F_a(s)}$$

$$X_1(s) = \frac{1}{m_1 s^2 + c_1 s + k_1} \cdot k_{hw} \cdot (F_a(s) - k_v s \cdot X_1(s))$$

$$X_1(s) \cdot \left(1 + \frac{k_{hw} \cdot k_v \cdot s}{m_1 s^2 + c_1 s + k_1}\right) = F_a(s) \cdot \frac{k_{hw}}{m_1 s^2 + c_1 s + k_1}$$

$$X_1(s) \cdot \left[m_1 s^2 + (c_1 k_{hw} k_v) s + k_1\right] = F_a(s) \cdot k_{hw}$$

$$G'_{p_s} = \frac{X_1(s)}{F_a(s)} = \frac{k_{hw}}{m_1 s^2 + (c_1 k_{hw} k_v) s + k_1}, C.Q.D. \tag{1}$$

2. Problema da regulação

$$G'_{p_r} := \frac{X_1(s)}{F_p(s)}$$

$$X_1(s) = \frac{1}{m_1 s^2 + c_1 s + k_1} \cdot (F_p(s) - k_{hw} \cdot k_v s \cdot X_1(s))$$

$$X_1(s) \cdot \left(1 + \frac{k_{hw} \cdot k_v \cdot s}{m_1 s^2 + c_1 s + k_1}\right) = F_p(s) \cdot \frac{1}{m_1 s^2 + c_1 s + k_1}$$

$$X_1(s) \cdot \left[m_1 s^2 + (c_1 k_{hw} k_v) s + k_1\right] = F_p(s)$$

$$G'_{p_r} = \frac{X_1(s)}{F_p(s)} = \frac{1}{m_1 s^2 + (c_1 k_{hw} k_v) s + k_1}, C.Q.D. \tag{2}$$

Substituindo k_1 por k_1^*, como deve ser feito para a simulação de perturbação:

$$G'_{p_r^*} = \frac{1}{m_1 s^2 + (c_1 k_{hw} k_v) s + k_1^*} = \frac{1}{m_1 s^2 + (c_1 k_{hw} k_v) s + (k_1 + \Delta k_1)}.$$

3. (a) Os valores de ganho de baixa frequência calculados para o G'_{p_s} são:

$$G'_{p_s}(0) = \frac{14372}{2.778 \cdot s^2 + 76.6 \cdot s + 338.6} \approx 43.5086$$

$$G'_{p_s^*}(0) = \frac{14372}{2.778 \cdot s^2 + 76.6 \cdot s + 700} \approx 21.0457$$

O *script* desenvolvido em Matlab para gerar a função de transferência e obter os valores de ganho de baixa frequência foi:

```
% parametros iniciais
s = tf('s');

mc1 = 0.778;
mw1 = 4*0.500;
m1 = mc1 + mw1;

c1 = 2.94;
kv = 0.005;
khw = 14732;

k1 = 338.6;
deltak1 = 361.4;

% Gps eh a funcao de transferencia da "planta compensada" para o problema servo
% GpsDelta eh Gps*, a funcao de transferencia para o valor perturbado
Gps = khw / (m1*s^2 + (c1+khw*kv)*s + k1);
GpsDelta = khw / (m1*s^2 + (c1+khw*kv)*s + (k1 + deltak1));

display(tf(Gps));
display(tf(GpsDelta));

% ganhos de baixa frequencia
Gps0 = dcgain(Gps);
Gps0Delta = dcgain(GpsDelta);

display(Gps0);
display(Gps0Delta);
```

Os resultados obtidos no Matlab são:

```
ans =
              14732
  --------------------------
  2.778 s^2 + 76.6 s + 338.6

Continuous-time transfer function.

ans =
             14732
  ------------------------
  2.778 s^2 + 76.6 s + 700

Continuous-time transfer function.

Gps0 =
   43.5086

Gps0Delta =
   21.0457
```

Como pode-se ver, os resultados são exatamente iguais aos calculados.

(b) Os valores de ganho de baixa frequência calculados para o G'_{p_r} são:

$$G'_{p_r}(0) = \frac{1}{2.778 \cdot s^2 + 76.6 \cdot s + 338.6} \approx 0.00295334$$

$$G'_{p_r^*}(0) = \frac{1}{2.778 \cdot s^2 + 76.6 \cdot s + 700} \approx 0.00142857$$

O *script* desenvolvido em Matlab para obter os valores de ganho de baixa frequência, considerando os já implementados parâmetros inicias, foi:

```
% funcoes de transferencia
Gpr = 1 / (m1*s^2 + (c1+khw*kv)*s + k1);
GprDelta = 1 / (m1*s^2 + (c1+khw*kv)*s + (k1 + deltak1));

display(tf(Gpr));
display(tf(GprDelta));

% ganhos de baixa frequencia
Gpr0 = dcgain(Gpr);
Gpr0Delta = dcgain(GprDelta);

display(Gpr0);
display(Gpr0Delta);
```

Os resultados obtidos no Matlab são:

```
ans =
                  1
  ---------------------------
  2.778 s^2 + 76.6 s + 338.6

Continuous-time transfer function.

ans =
                 1
  ------------------------
  2.778 s^2 + 76.6 s + 700

Continuous-time transfer function.

Gpr0 =
    0.0030

Gpr0Delta =
    0.0014
```

Como pode-se ver, os resultados são exatamente iguais aos calculados.

(c)

```
display(Gps0Delta - Gps0);
display(Gpr0Delta - Gpr0);
```

```
  -22.4629
   -0.0015
```

(d)

```
% parametros iniciais
Gpf = 1;
kp = 0.12;

% funcoes de transferencia
Ga = Gpf * Gps;
GaDelta = Gpf * GpsDelta;

Gf = Gpf * feedback(Gps * kp, 1);
GfDelta = Gpf * feedback(GpsDelta * kp, 1);

display(tf(Ga));
display(tf(GaDelta));
display(tf(Gf));
display(tf(GfDelta));
```

```
ans =
              14732
  --------------------------
  2.778 s^2 + 76.6 s + 338.6

Continuous-time transfer function.


ans =
             14732
  ------------------------
  2.778 s^2 + 76.6 s + 700

Continuous-time transfer function.

ans =
              1768
  -------------------------
  2.778 s^2 + 76.6 s + 2106

Continuous-time transfer function.


ans =
              1768
  -------------------------
  2.778 s^2 + 76.6 s + 2468

Continuous-time transfer function.
```

(e)

$$G\left(s\right)=\frac{X_1\left(s\right)}{X_{1r}\left(s\right)} \implies X_1\left(s\right)=G\left(s\right)\cdot X_{1r}\left(s\right)$$

$$\begin{aligned} e_r=\lim_{s\to 0} sE\left(s\right) &= \lim_{s\to 0} s\cdot\left(X_{1r}\left(s\right)-X_1\left(s\right)\right)\\ &= \lim_{s\to 0} s\cdot\left(X_{1r}\left(s\right)-G\left(s\right)\cdot X_{1r}\left(s\right)\right)\\ &= \lim_{s\to 0} s\cdot X_{1r}\left(s\right)\cdot\left(1-G\left(s\right)\right)\end{aligned}$$

Dado que a entrada x_{1r} é o degrau unitário; X_{1r}, que é a sua transformada, é igual a $1/s$. Logo:

$$e_r = \lim_{s \to 0} s \cdot \frac{1}{s} \cdot (1 - G(s))$$

$$e_r = \lim_{s \to 0} 1 - G(s) \tag{3}$$

O *script* para obter os valores de erro de regime dos sistemas em malha aberta e em malha fechada foi, então:

```
% calculo dos erros de regime
eRegA = dcgain(1 - Ga);
eRegADelta = dcgain(1 - GaDelta);

eRegF = dcgain(1 - Gf);
eRegFDelta = dcgain(1 - Gfdelta);

display(eRegA);
display(eRegF);
display(eRegADelta);
display(eRegFDelta);
```

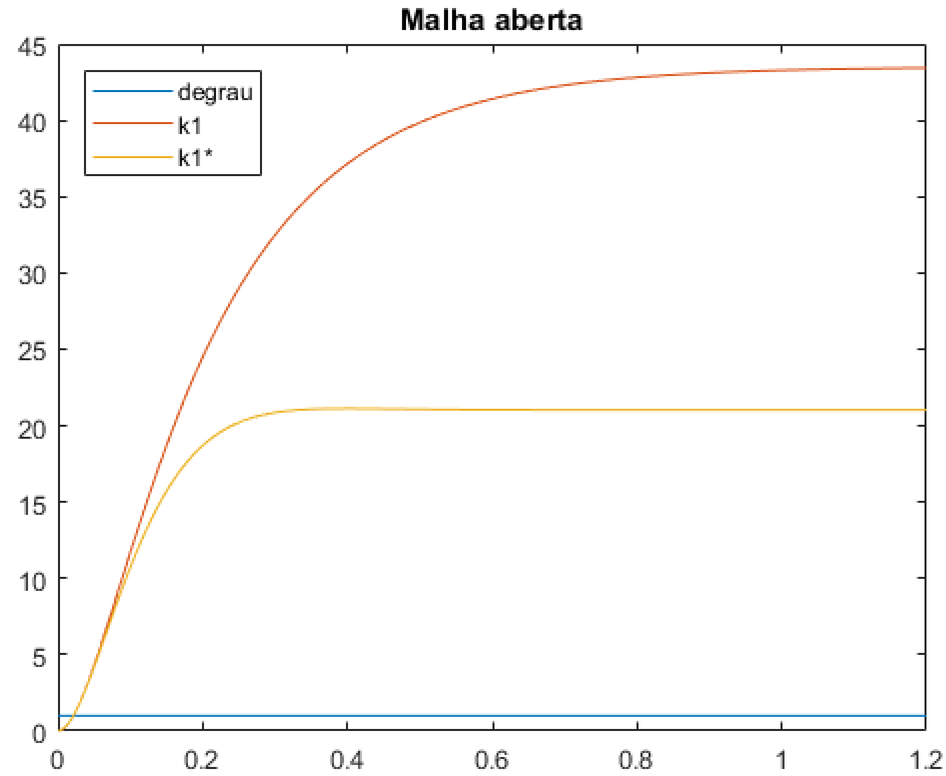
```
eRegA =
   -42.5086

eRegF =
     0.1607

eRegADelta =
   -20.0457

eRegFDelta =
     0.2836
```

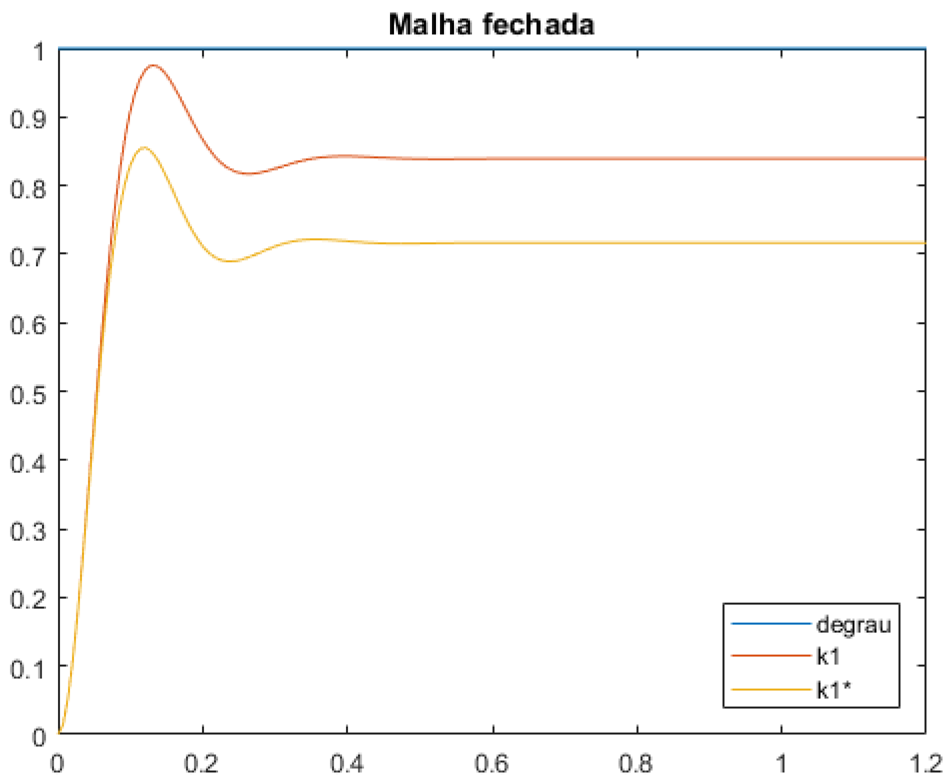
Como podemos ver, para k_1 e k_1^*, os valores de erro de regime são bem menores em malha fechada do que em malha aberta, respectivamente. Isso se dá por conta da regulação obtida pela realimentação do sinal de saída x_1.

(f) O *script* desenvolvido em Matlab para gerar as respostas ao degrau dos sistemas (considerando k_1 e k_{1r} em malha aberta e em malha fechada foi:

```
% respostas ao degrau
[stepGa, tA] = step(Ga, 1.2);
[stepGaDelta, tADelta] = step(GaDelta, 1.2);
[stepGf, tF] = step(Gf, 1.2);
[stepGfDelta, tFDelta] = step(GfDelta, 1.2);

% funcoes degrau para malha aberta e fechada
degrauA = ones(length(stepGa), 1);
degrauF = ones(length(stepGf), 1);

% graficando malha aberta para k1 e k1*
figure, plot(tA, degrauA, tA, stepGa, tADelta, stepGaDelta);
title('Malha aberta');
legend('degrau', 'k1', 'k1*', 'Location', 'northwest');

% graficando malha fechada para k1 e k1*
figure, plot(tF, degrauF, tF, stepGf, tFDelta, stepGfDelta);
title('Malha fechada');
legend('degrau', 'k1', 'k1*', 'Location', 'southeast');
```

Os gráficos gerados são:





4. Para que o erro de regime de malha aberta seja mínimo (nulo), usando a equações 1 e 3:

$$\lim_{s \to 0} 1 - G_a\left(s\right) = 0 \implies$$

$$\lim_{s \to 0} 1 - G_{pf}\left(s\right) \cdot G'_{p_s}\left(s\right) = \lim_{s \to 0} 1 - k_{pf} \cdot \frac{k_{hw}}{m_1 s^2 + \left(c_1 k_{hw} k_v\right) s + k_1} = 1 - k_{pf} \cdot \frac{k_{hw}}{k_1} = 0 \implies$$

$$k_{pf} = \frac{k_1}{k_{hw}}$$