Pré-relatório Experiência 6
Prof. Marconi Kolm Madrid
EA722 - 2017/2

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

Projeto de realimentação do carro 1

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

% Massa dos carros
mc1 = 0.783; % kg
mc2 = 0.582; % kg

% Massa total dos carros
m1 = mc1 + 4 * 0.500; % kg
m2 = mc2 + 4 * 0.500; % kg

% Coeficiente de atrito dos carros
c1 = 3.90; % N/(m/s)
c2 = 2.36; % N/(m/s)

% Constante de mola
k = 338.6; % N/m

% Ganho de hardware
khw = 14732;

N1 = [m2 c2 k];
N2 = [k];
D = [(m1*m2) (c1*m2 + c2*m1) ((m1+m2)*k + c1*c2) ((c1+c2)*k) 0];
```

1.

```
%% Realimentacao - 1
% Implemente as funcoes de transferencia da planta utilizando os valores
% numericos para definir X1(s)/R*(s)

G1 = khw * tf(N1, D);
```

2.

```
%% Realimentacao - 2
% Determine atraves do lugar das raizes (root locus) o valor de kv que
% forneca o maximo amortecimento

rlocus(s * G1);
grid on;
```

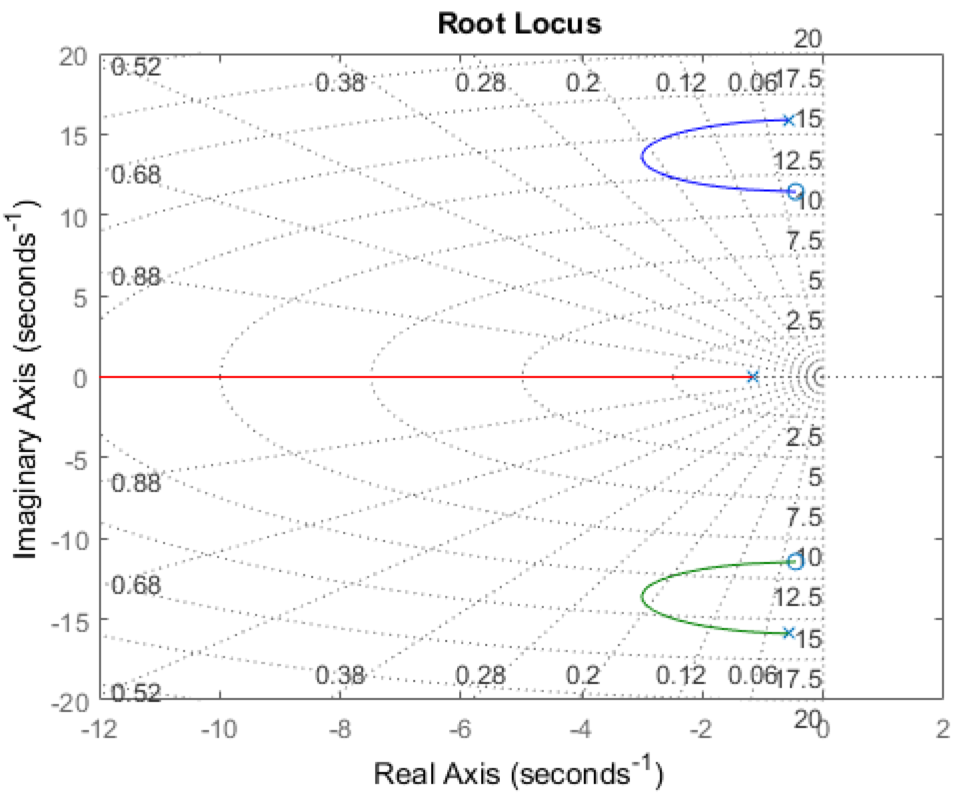


Figura 1: Lugas das raízes de $X_1(s)/R^*(s)$

Ganho do valor de máximo amortecimento: $gain = 0.00359$.

3.

```
%% Realimentacao - 3
% Implemente kv e determine os polos da funcao de transferencia interna
% G*(s). Selecione os polos complexos conjugados desta f.t., denominando-os
% p1 e p2.

kv = 0.00359;

Gstar = tf(N2, N1) * feedback(khw * tf(N1, D), kv * s);
Gstar = minreal(Gstar);

% obtem o denominador de G*(s), acha suas raizes e filtra aquelas cuja
% parte imaginaria eh diferente de 0 (polos complexos conjugados)
[~, den] = tfdata(Gstar);
p = roots(den{1});
p_id = find(imag(p) ~= 0);
p1 = p(p_id(1));
p2 = p(p_id(2));
```

Resultado:

```
p1 = -2.9436 +13.1126i;
p2 = -2.9436 -13.1126i;
```

Projeto do filtro notch

1.

```
%% Notch - 1
% Calculam-se os parametros do filtro Nn(s)/Dn(s) de modo que:
% 1. os dois zeros do filtro cancelem dois polos de G*(s) (tipicamente
% polos pouco amortecidos), isto eh, raizes de D*(s) complexas conjugadas.

Nn = poly([p1 p2]);

% 2. o filtro possua dois pares de polos complexos conjugados de frequencia
% natural fn1 = 5Hz e fn2 = 8Hz respectivamente, e csi = sqrt(2) / 2 para
% ambos os pares.

csi = sqrt(2) / 2;

wn1 = 5 * 2 * pi; % 5 Hz
wn2 = 8 * 2 * pi; % 8 Hz

% 3. o coeficiente do termo de maior grau do polinomio Dn(s) deve
% ser 1 (polinomio monico) e o ganho estatico (DC) da funcao de
% transferencia do filtro deve ser unitario.

Dn = conv([1 (2*csi*wn1) wn1^2], [1 (2*csi*wn2) wn2^2]);

Gnotch = tf(Nn, Dn);
Gnotch = Gnotch * (1 / dcgain(Gnotch));
```

2.

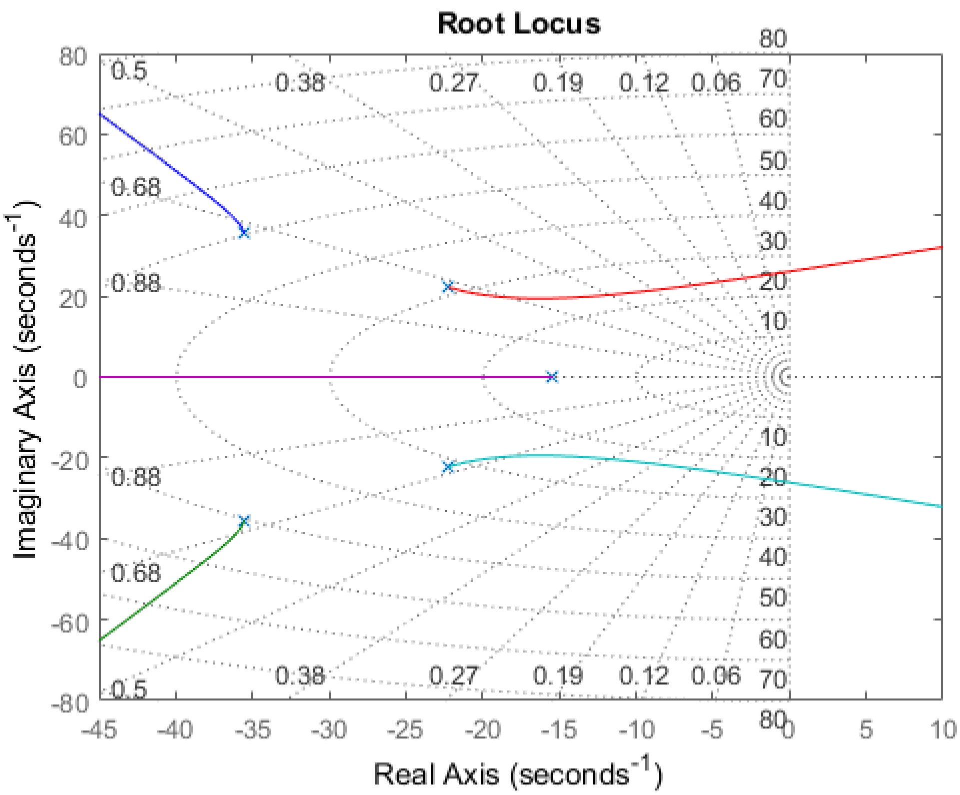
```
%% Notch - 2
% Associe G*(s) ao filtro projetado.
G2 = minreal(Gnotch * Gstar);
```

Projeto do controlador P&D

1.

```
%% P&D - 1
% Determine atraves do lugar das raizes o valor do ganho kd de forma a se
% obter o maximo amortecimento para os polos dominantes da funcao de
% transferencia da saida x2(t)

rlocus(s * G2);
grid on;
```



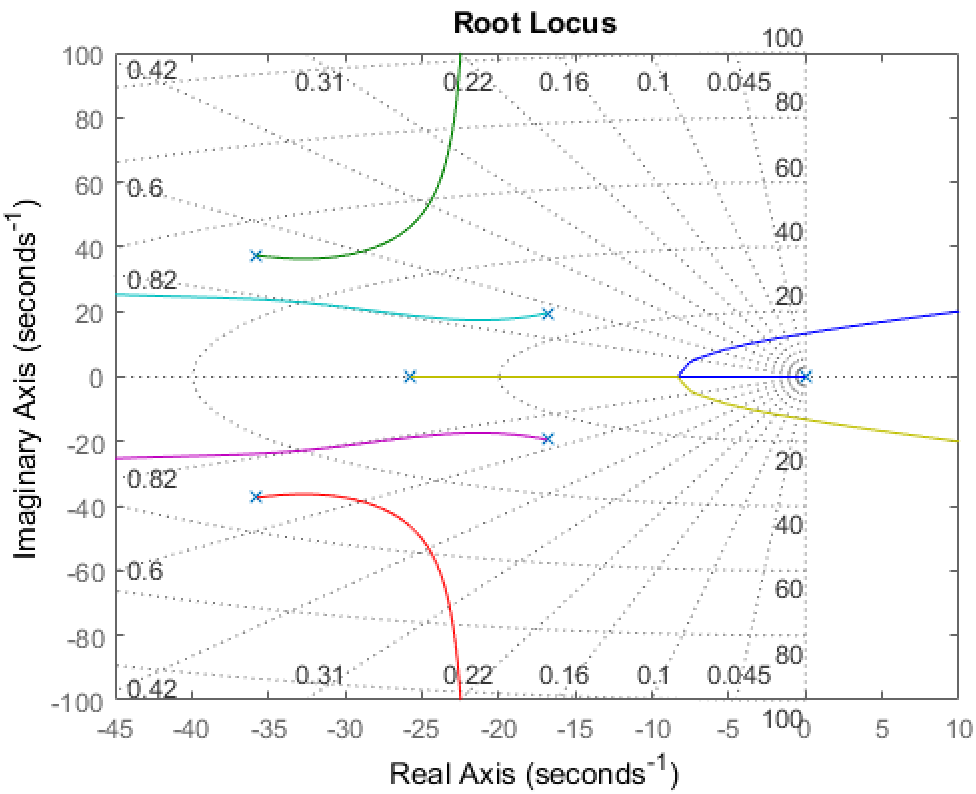
Ganho do valor de máximo amortecimento: $gain = 0.00074$.

2.

```
%% P&D - 2
% Implemente o valor de kd e determine atraves do lugar das raizes o valor
% do ganho kp que tenha o minimo tempo de estabelecimento.

kd = 0.00074;

G3 = feedback(G2, kd * s);
rlocus(G3);
grid on;
```



Ganho do valor com mínimo tempo de estabelecimento: $gain = 0.0147$.

3.

```
%% P&D - 3
% Utilize a resposta ao degrau do sistema em malha fechada com x2(t) como
% saida, como criterio para verificacao da adequacao do ajuste.

kp = 0.0147;

G = feedback(kp * G3, 1);
step(G);
grid on;
```

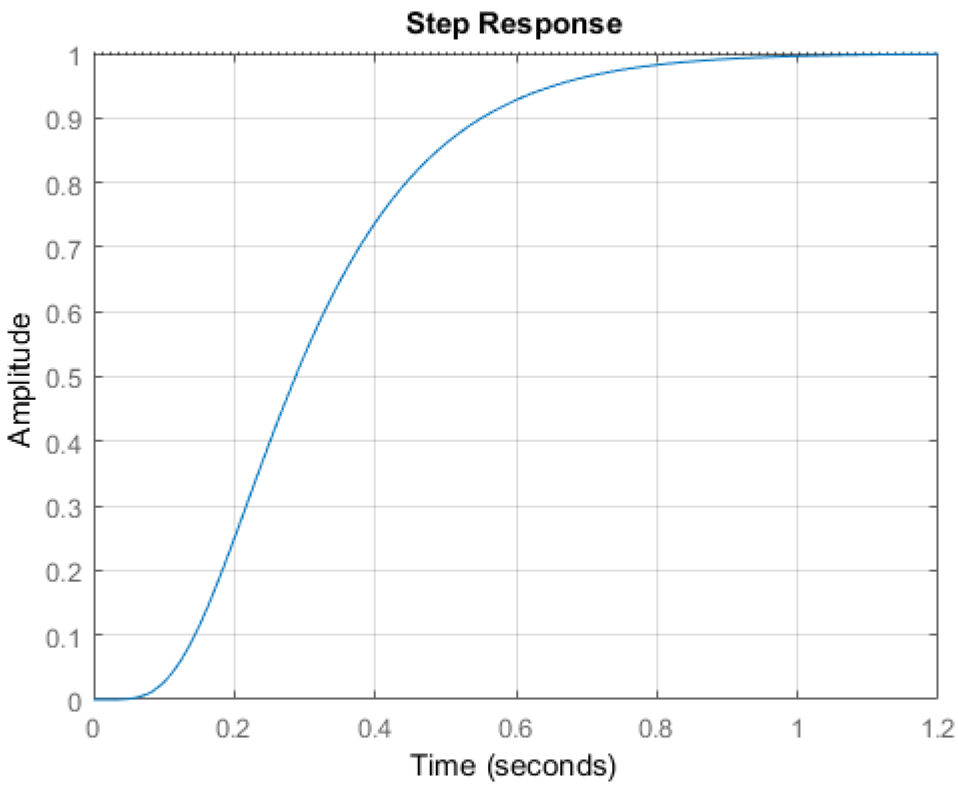


Figura 2: Resposta ao degrau do sistema de malha fechada

Implementação no software ECP

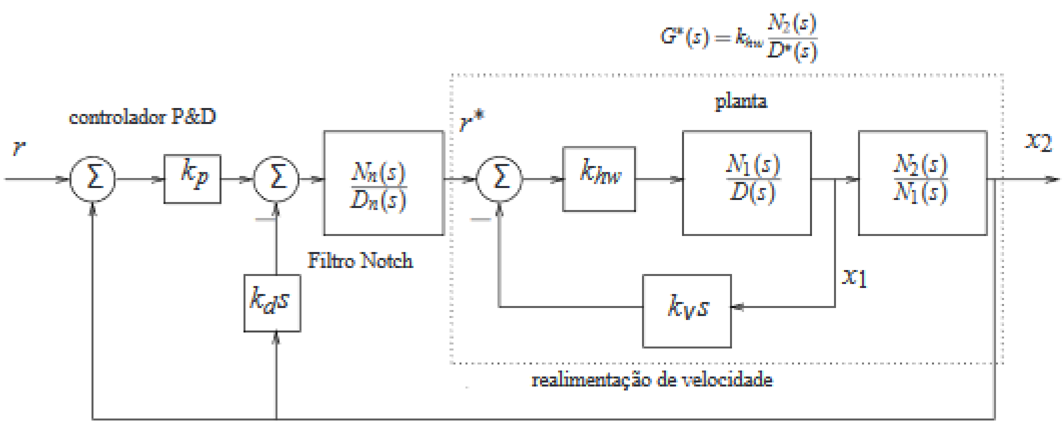


Figura 3: Diagrama para o controle não-co-alocado

Para o diagrama de blocos acima, temos que sua função de transferência se dá por:

$$X_2(s) = G^*(s) \cdot G_{notch}(s) \cdot [k_p \cdot (R(s) - X_2(s)) - k_d s \cdot X_2(s)]$$
$$X_2(s) \cdot [1 + G^*(s) \cdot G_{notch}(s) \cdot (k_p + k_d s)] = G^*(s) \cdot G_{notch} \cdot k_p \cdot R(s)$$

$$\frac{X_2(s)}{R(s)} = \frac{k_p \cdot G^*(s) \cdot G_{notch}(s)}{1 + G^*(s) \cdot G_{notch}(s) \cdot (k_p + k_d s)}$$

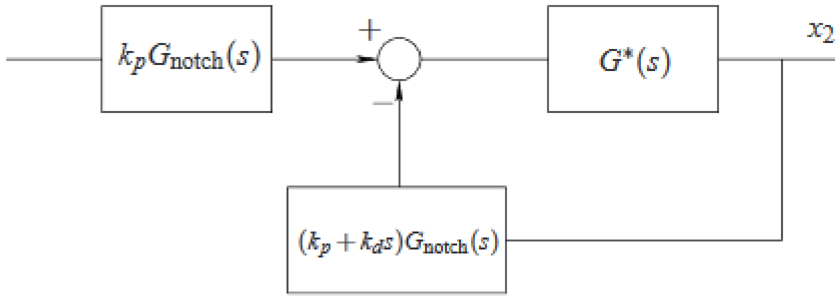


Figura 4: Representação do filtro *notch* + P&D implementado na malha do *loop 1*

Para o diagrama de blocos acima, temos que sua função de transferência se dá por:

$$X_2 = \frac{G^*(s)}{1 + G^*(s)(k_p + k_d s) \cdot G_{notch}(s)} \cdot k_p \cdot G_{notch}(s) \cdot R(s)$$
$$\frac{X_2(s)}{R(s)} = \frac{k_p \cdot G^*(s) \cdot G_{notch}(s)}{1 + G^*(s) \cdot G_{notch}(s) \cdot (k_p + k_d s)}$$

Pode-se ver, então, que a função de transferência do diagrama de blocos da figura 3 é igual à função de transferência do diagrama de blocos da figura 4, como queria-se demonstrar.

O bloco correspondente a $k_p G_{notch}(s)$ se dá na forma $\frac{t(s)}{r(s)}$, enquanto o bloco $(k_p + k_d s) G_{notch}(s)$ se dá como $\frac{s(s)}{r(s)}$. Denotando o numero e o denominador do filtro *notch* por $n_2 s^2 + n_1 s + n_0$ e $s^4 + d_3 s^3 + d_2 s^2 + d_1 s + d_0$, temos as seguintes relações entre os coeficientes dos polinômios:

$$t_0 = n_0 k_p \qquad t_1 = n_1 k_p \qquad t_2 = n_2 k_p$$
$$s_0 = n_0 k_p \qquad s_1 = n_0 k_d + n_1 k_p \qquad s_2 = n_1 k_d + n_2 k_p \qquad s_3 = n_2 k_d$$
$$r_0 = d_0 \qquad r_1 = d_1 \qquad r_2 = d_2 \qquad r_3 = d_3 \qquad r_4 = 1$$

Para o cálculo desses coeficientes t_i, s_i e r_i, o código Matlab adicionado foi:

```
%% Calculo dos coeficientes

% Coeficientes do filtro notch
[num, den] = tfdata(Gnotch);

Nn = num{1};
Dn = den{1};

n0 = Nn(5);
n1 = Nn(4);
n2 = Nn(3);

d0 = Dn(5);
d1 = Dn(4);
d2 = Dn(3);
d3 = Dn(2);
d4 = Dn(1);

% Coeficientes dos blocos
t0 = n0*kp;
t1 = n1*kp;
t2 = n2*kp;

s0 = n0*kp;
s1 = n0*kd + n1*kp;
s2 = n1*kd + n2*kp;
```

```
s3 = n2*kd;

r0 = d0;
r1 = d1;
r2 = d2;
r3 = d3;
r4 = 1;
```