# COE-835 Controle adaptativo

## Trabalho 12

**Grupo:** Guilherme Pires Sales de Carvalho
Matheus Ferreira dos Reis
Renan Salles de Freitas

**Algoritmo:** MIMO MRAC Direto ($n^* = 2$)

**Caso**: $n = 2$ (ordem da planta)
$n^* = 2$ (grau relativo)
$n_p = 5, 17$ (# de parâmetros)

## Conteúdo

# 1 MIMO MRAC direto ($n^* = 2$)

O controle adaptativo de modelo de referência (MRAC) é um método de controle adaptativo com uma fundação teórica rigorosa e sistemática, possui promissora e atraente perspectiva de aplicação, e seu projeto é simples e conciso. O objetivo do MRAC é garantir que a resposta de saída de um sistema controlado (planta) convirja para a resposta de um modelo de referência, dado que a planta possui parâmetros desconhecidos. Isto é feito através de uma lei de adaptação dos parâmetros do controlador que garante a estabilidade do sistema em malha fechada e a *limitação* dos parâmetros estimados.

Assim como no caso SISO, considere um sistema linear invariante no tempo descrito pela equação diferencial:

$$y(t) = P(s)\,[u(t)], \tag{1}$$

No caso MIMO, $y(t) \in \mathbb{R}^n$ e $u(t) \in \mathbb{R}^n$ são os sinais medidos de saída e entrada do sistema, sendo $P(s) \in \mathbb{R}^{n\times n}$ a matriz função de transferência. Neste trabalho, vamos considerar o problema em que a planta possui grau relativo $n^* = 2$, para o caso MIMO de duas entradas e duas saídas:

$$P(s) = \begin{bmatrix} \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s^2} \end{bmatrix} K_p \tag{2}$$
$$K_p = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -h\sin(\phi) & h\cos(\phi) \end{bmatrix}$$

Como no caso de grau relativo $n^* = 1$, a matriz *interactor* de $P(s)$ é tal que:

$$\lim_{s\to\infty} \xi(s)P(s) = K_p \tag{3}$$

é não singular. No caso de grau relativo 2 ($n^* = 2$), a matriz *interactor* é $\xi(s) = s^2 I$.

Define-se índice de observabilidade $\nu$ como os graus dos polinômios da matriz $P(s)$. E *observability index* como o maior grau.

Dado um modelo de referência descrito como:

$$y_M(t) = M(s)r(t), \tag{4}$$

onde $M(s) = \text{diag}\left\{\frac{1}{(s+a_i)(s+\lambda)}\right\}$ é uma matriz diagonal com polinômios estáveis e mônicos, e $r(t) \in \mathbb{R}^n$ é a entrada do sistema MIMO, sinal limitado. O objetivo é encontrar uma lei de controle $u(t)$ tal que todo o sistema de malha fechada produza sinais limitados e a saída da planta $y(t)$ rastreie assintoticamente o modelo de referência $y_M(t)$. A estrutura exige algumas premissas:

1. Os zeros de $P(s)$ tem parte real negativa;
2. $P(s)$ tem posto completo e grau relativo 2;
3. O *Observability index* de $P(s)$ é conhecido;
4. Os sinais dos menores principais líderes da matriz $K_p$ são conhecidos.
5. O modelo de referencia é dado por uma função de transferência $P_M(s)$. E $M(s)L(s)$ é SPR, onde $L(s) = (s+\lambda)I$.

É possível provar que existe uma fatoração SDU não única de $K_p$ (notas de aula). Assim como caso SISO, a estrutura do controlador é 2DOF. Escolhe-se então um polinômio estável $\Lambda(s) = \text{diag}\left\{(s+a_i)^{\nu-1}\right\}$ para filtrar a entrada e a saída da planta com o filtro $\frac{1}{\Lambda(s)}$. O filtro é descrito pela realização de estados:

$$\dot\omega_1(t) = A_\lambda\,\omega_1(t) + B_\lambda u(t)\,, \tag{5}$$
$$\dot\omega_2(t) = A_\lambda\,\omega_2(t) + B_\lambda y(t)\,, \tag{6}$$

Observe que agora $B_\lambda$ é matriz para o caso MIMO. Podemos, então, formular a equação do erro:

$$\begin{aligned} e &= M(s)K_p\left[u - \theta^{*\intercal}\omega\right] \\ &= M(s)SD\left[Uu - U\theta_1^{*\intercal}\omega_1 - U\theta_2^{*\intercal}\omega_2 - U\theta_3^{*\intercal}y - U\theta_4^{*\intercal}r\right] \\ &= M(s)SD\left[u - K_1\omega_1 - K_2\omega_2 - K_3y - K_4r - K_5u\right] \\ &= M(s)SD\left[u - u^*\right] \end{aligned} \tag{7}$$

Notar que $K_5$ é matriz estritamente superior. E $u^*$ pode ser descrito como:

$$u^* = \begin{bmatrix} \Theta_1^{*\intercal}\Omega_1 \\ \Theta_2^{*\intercal}\Omega_2 \\ \vdots \\ \Theta_m^{*\intercal}\Omega_m \end{bmatrix} \tag{8}$$

onde $\Omega_i$ são os novos regressores para o caso MIMO:

$$\begin{aligned} \Omega_1^\intercal &= \begin{bmatrix}\omega^\intercal & u_2 & u_3 & \dots & u_m\end{bmatrix} \\ \Omega_2^\intercal &= \begin{bmatrix}\omega^\intercal & u_3 & \dots & u_m\end{bmatrix} \\ &\vdots \\ \Omega_m^\intercal &= \begin{bmatrix}\omega^\intercal\end{bmatrix} \\ \Omega^\intercal &= \begin{bmatrix} \Omega_1^\intercal & 0 & \dots & 0 \\ 0 & \Omega_2^\intercal & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Omega_m^\intercal \end{bmatrix} \end{aligned} \tag{9}$$

e $\Theta$ é o vetor de parâmetros:

$$\Theta^\intercal = \begin{bmatrix} \Theta_1^* \\ \Theta_2^* \\ \vdots \\ \Theta_m^* \end{bmatrix} \tag{10}$$

A nova estrutura MIMO MRAC pode ser vista na figura 1.

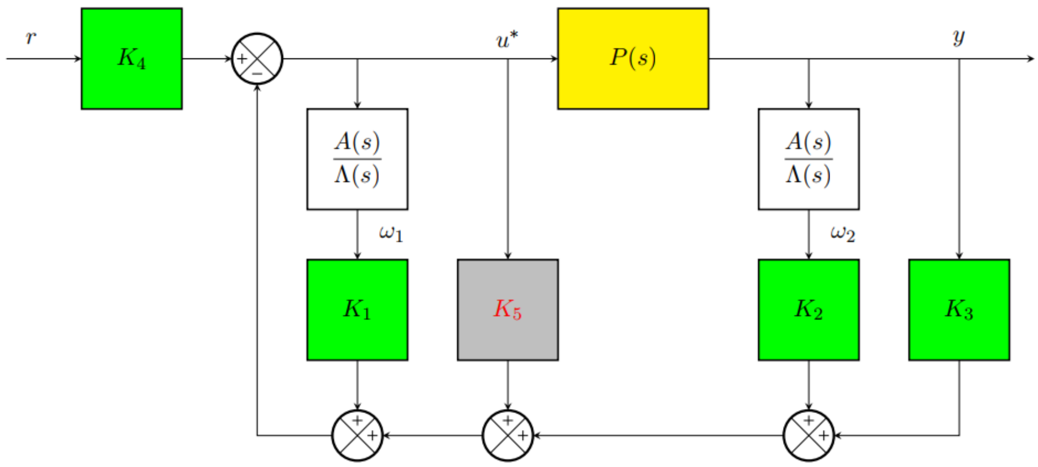



Figura 1: Estrutura MIMO MRAC.

Utilizando o algoritmo de Monopoli, temos:

$$e = [M(s)S]\, DL(s)L^{-1}(s)\, [u - \Omega^{\intercal}\Theta^*] \tag{11}$$
$$= [M(s)L(s)S]\, D\left[L^{-1}(s)u - L^{-1}(s)\Omega^{\intercal}\Theta^*\right] \tag{12}$$
$$u = L(s)\chi \tag{13}$$

Escolhmos:

$$\chi = \zeta^{\intercal}\Omega \tag{14}$$
$$u = \Omega^{\intercal}\Omega + \zeta^{\intercal}\dot{\Omega} \tag{15}$$

Como no caso $n^* = 1$, é possível provar que para todo modelo de referência $M(s)$ existe $D_+$ tal que $M(s)L(s)S = M(s)L(s)L_1D_+L_1^{\intercal}$ é SPR. Escolhendo a função de Lyapunov:

$$2V = z^{\intercal}P_M z + \tilde{\Theta}^{\intercal}|D|\Gamma^{-1}\tilde{\Theta} \tag{16}$$

podemos chegar na lei de adaptação:

$$\dot{\Theta}(t) = -\mathrm{sign}[D]\,\Gamma\,\zeta\,e(t), \tag{17}$$

onde $D = \mathrm{diag}\,\{d_1I_1, d)2I_2, \ldots, d_mI_m\}$

o erro converge assintoticamente para zero, em que $\Gamma = \Gamma^{\intercal} > 0$ é uma matriz de ganhos positiva definida.

Observe que ainda falta analisar a estabilidade de $L(s)$. Nas notas de aula, pode-se verificar a prova completa com a segunda lei de Lyapunov:

$$V_L(z, \epsilon, \tilde{\Theta}) = V + \alpha\epsilon^{\intercal}P_1\epsilon \tag{18}$$

Neste trabalho, serão consideradas plantas de primeira e segunda ordens com grau relativo 2 ($n^* = 2$). Iremos simular e discutir o comportamento do erro e das saídas para variações das condições iniciais dos parâmetros estimados ($\theta(0)$) e da planta ($y(0)$), do ganho de adaptação $\Gamma$ e para diferentes parâmetros da planta e modelo.

# 2 Implementação

Foram considerados dois casos: caso em que $K_p$ é desconehcido; e caso em que todos os parâmetros da planta são desconhecidos.

### 2.1 Caso em que apenas $K_p$ é desconhecido

Consideremos a planta descrita pela eq. 2. A equação ideal para o controlador 2DOF poode ser computada pela equação do erro eq. 7:

$$\begin{aligned}
u^* &= \theta_1^{*\intercal}\omega_1 + \theta_2^{*\intercal}\omega_2 + \theta_3 y + \theta_4 r\\
\omega_1 &= \frac{u^*}{\Lambda}\\
\omega_2 &= \frac{y}{\Lambda}\\
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) u^* &= \theta_2^{*\intercal}\Lambda^{-1} y + \theta_3 y + \theta_4 r
\end{aligned}$$

Temos que:

$$\frac{1}{\Lambda(s)} = \begin{bmatrix} \frac{1}{(s+\lambda_f)} & 0 \\ 0 & \frac{1}{(s+\lambda_f)} \end{bmatrix}$$

e

$$M(s) = \begin{bmatrix} \frac{\lambda^2}{(s+\lambda)^2} & 0 \\ 0 & \frac{\lambda^2}{(s+\lambda)^2} \end{bmatrix}$$

Voltando à equação da planta:

$$\begin{aligned}
y &= PK_p u\\
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) y &= \left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) P\bar{u}
\end{aligned}$$

Como $P(s)$ é diagonal, temos $AP = PA$ (comuta). E $y = y_m$, logo:

$$\begin{aligned}
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) y_m &= P\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right)\bar{u}\\
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) M(s) r &= P(\theta_2^{*\intercal}\Lambda^{-1} y_m + \theta_3 y_m + \theta_4 r)\\
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) M(s) r &= P(\theta_2^{*\intercal}\Lambda^{-1} M(s) r + \theta_3 M(s) r + \theta_4 r)\\
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) M(s) r &= P(\theta_2^{*\intercal}\Lambda^{-1} M(s) + \theta_3 M(s) + \theta_4) r\\
\left(I - \theta_1^{*\intercal}\Lambda^{-1}\right) M(s) &= P(\theta_2^{*\intercal}\Lambda^{-1} M(s) + \theta_3 M(s) + \theta_4)
\end{aligned}$$

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \theta_1^{*\intercal}\begin{bmatrix} \frac{1}{(s+\lambda_f)} & 0 \\ 0 & \frac{1}{(s+\lambda_f)} \end{bmatrix}\right)\begin{bmatrix} \frac{\lambda^2}{(s+\lambda)^2} & 0 \\ 0 & \frac{\lambda^2}{(s+\lambda)^2} \end{bmatrix} =$$
$$\begin{bmatrix} \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s^2} \end{bmatrix}\left(\theta_2^{*\intercal}\begin{bmatrix} \frac{1}{(s+\lambda_f)} & 0 \\ 0 & \frac{1}{(s+\lambda_f)} \end{bmatrix}\begin{bmatrix} \frac{\lambda^2}{(s+\lambda)^2} & 0 \\ 0 & \frac{\lambda^2}{(s+\lambda)^2} \end{bmatrix} + \theta_3\begin{bmatrix} \frac{\lambda^2}{(s+\lambda)^2} & 0 \\ 0 & \frac{\lambda^2}{(s+\lambda)^2} \end{bmatrix} + \theta_4\right)$$

Como todas as matrizes são diagonais, vamos resolver para o primeiro termo:

$$\left(1 - \frac{\theta_1}{(s+\lambda_f)}\right)\frac{\lambda^2}{(s+\lambda)^2} = \frac{1}{s^2}\left(\frac{\theta_2}{(s+\lambda_f)}\frac{\lambda^2}{(s+\lambda)^2} + \theta_3\frac{\lambda^2}{(s+\lambda)^2} + \theta_4\right)$$

$$(\lambda^2 - \theta_4)s^3 + (\lambda^2\lambda_f - \theta_1\lambda^2 - \theta_4(\lambda_f + 2\lambda))s^2 + (-\theta_3\lambda^2 - \theta_4(2\lambda\lambda_f + \lambda^2))s + (-\theta_2\lambda^2 - \theta_3\lambda^2\lambda_f - \theta_4\lambda^2\lambda_f) = 0$$

Resolvendo para $\theta$, temos:

$$\begin{aligned}
\theta_4 &= \lambda^2 \\
\theta_1 &= -2\lambda \\
\theta_3 &= -(2\lambda\lambda_f + \lambda^2) \\
\theta_2 &= 2\lambda\lambda_f^2
\end{aligned}$$

Desssa forma, a lei de controle é descrita como:

$$u^* = K_p^{-1}\left[2\lambda\lambda_f^2\Lambda^{-1}y - (\lambda^2 + 2\lambda\lambda_f)y + \lambda^2 r\right] - 2\lambda\Lambda^{-1}u \tag{19}$$

Utilizando o algoritmo de Monopoli 11, temos:

$$\begin{aligned}
u &= \Omega^\intercal\Theta + \zeta^\intercal\dot{\Theta} - 2\lambda\omega_1 && (20)\\
\Omega_1^\intercal &= [\omega^\intercal \quad u_2 + 2\lambda\omega_{1_2}] \\
\Omega_2^\intercal &= [\omega^\intercal] && (21)\\
\omega &= 2\lambda\lambda_f^2\omega_2 - (\lambda^2 + 2\lambda\lambda_f)y + \lambda^2 r && (22)\\
\omega_f &= L^{-1}(s)\omega && (23)\\
\zeta_1^\intercal &= \omega_f^\intercal\,[I \quad \Theta_2] \\
\zeta_2^\intercal &= \omega_f^\intercal && (24)\\
\dot{\Theta} &= -\text{sign}(D)\Gamma\zeta e && (25)
\end{aligned}$$

A implementação segue abaixo:

```
%=====================================================================
%
%   COE-835   Controle adaptativo
%
%   Script para simular o exemplo
%
%   MRAC  : n   = 2      First order plant
%           n* = 2      Relative degree
%           np = 5      Adaptive parameters
%
%=====================================================================
function dx=mrac(t,x)

global sysP sysM sysL gamma a1 a2 w1 w2 lambda lambda_f;

xp      = x(1:4);
```

```
xm      = x(5:8);
uf      = x(9:10);
yf      = x(11:12);
omega_f    = x(13:14);
theta   = x(15:end);

y = sysP.C*xp;
ym = sysM.C*xm;
%-------------------------
r1 = 0;
r2 = 0;
for i=1:length(w1)
    r1 = r1 + a1(i)*sin(w1(i)*t);
    r2 = r2 + a2(i)*sin(w2(i)*t);
end

r = [r1 r2]';

theta_2_omega_2 = 2*lambda*lambda_f^2*yf;
theta_3_y = - (lambda^2 + 2*lambda*lambda_f)*y;
theta_4_r = lambda^2*r;

omega =  theta_2_omega_2 + theta_3_y + theta_4_r;
zeta1 = (omega_f'*[eye(2) theta(4:5)])';
zeta2 = omega_f;

e  = y - ym;

%% ------ Adaptacao -------
dtheta = -gamma*[zeta1'*e(1), zeta2'*e(2)]';

%% ----- u ------
Omega2 = omega;
u2 = Omega2'*theta(4:5) + zeta2'*dtheta(4:5) - 2*lambda*uf(2);

Omega1 = [omega' u2+2*lambda*uf(2)]';
u1 = Omega1'*theta(1:3) + zeta1'*dtheta(1:3) - 2*lambda*uf(1);

u = [u1 u2]';

%% ------- Planta --------
dxp = sysP.A*xp + sysP.B*u;

%% ------ Modelo --------
dxm = sysM.A*xm + sysM.B*r;

%% ------ omega_1 --------
duf = sysL.A*uf + sysL.B*u;

%% ------ omega_2 --------
dyf = sysL.A*yf + sysL.B*y;

%% ------ omega_f --------
domega_f = sysL.A*omega_f + sysL.B*omega;

%-------------------------
dx = [dxp' dxm' duf' dyf' domega_f' dtheta']';    %Translation

%---------------------------
```

## 2.2 Caso onde todos os parâmetros são desconhecidos

```
%==================================================================
%
%   COE-835   Controle adaptativo
%
%   Script para simular o exemplo
%
%   MRAC   : n   = 2      First order plant
%            n*  = 2      Relative degree
%            np  = 17     Adaptive parameters
%
%==================================================================
function dx=mrac(t,x)

global sysP sysM sysL gamma w1 w2 a1 a2;

xp          = x(1:4);
xm          = x(5:8);
uf          = x(9:10);
yf          = x(11:12);
omega_f     = x(13:20);
theta       = x(21:end);

y = sysP.C*xp;
ym = sysM.C*xm;
%-------------------------
r1 = 0;
r2 = 0;
for i=1:length(w1)
    r1 = r1 + a1(i)*sin(w1(i)*t);
    r2 = r2 + a2(i)*sin(w2(i)*t);
end

r = [r1 r2]';

omega = [uf' yf' y' r']';
zeta1 = (omega_f'*[eye(8) theta(10:end)])';
zeta2 = omega_f;

e  = y - ym;

%% ------ Adaptacao -------
dtheta = -gamma*[zeta1'*e(1), zeta2'*e(2)]';

%% ----- u ------
Omega2 = omega;
u2 = Omega2'*theta(10:end) + zeta2'*dtheta(10:end);

Omega1 = [omega' u2]';
u1 = Omega1'*theta(1:9) + zeta1'*dtheta(1:9);

u = [u1 u2]';

%% ------- Planta --------
dxp = sysP.A*xp + sysP.B*u;

%% ------- Modelo --------
dxm = sysM.A*xm + sysM.B*r;
```

```

%% ------ omega_1 --------
duf = sysL.A*uf + sysL.B*u;

%% ------ omega_2 --------
dyf = sysL.A*yf + sysL.B*y;

%% ------ omega_f --------
domega_f = sysL.A*omega_f + sysL.B*omega;

%-------------------------
dx = [dxp' dxm' duf' dyf' domega_f' dtheta']';    %Translation
```

# 3 Resultados das simulações

Nas simulações, procuramos avaliar o comportamento do sistema para as seguintes condições: (i) Condições iniciais $y(0)$; (ii) Parâmetros da planta e do modelo; (iii) ganho de adaptação $\Gamma$.

Apresentaremos os resultados obtidos através de simulações no ambiente `Matlab/Simulink` e os discutiremos na próxima seção.

## 3.1 Caso em que apenas $K_p$ é desconhecido

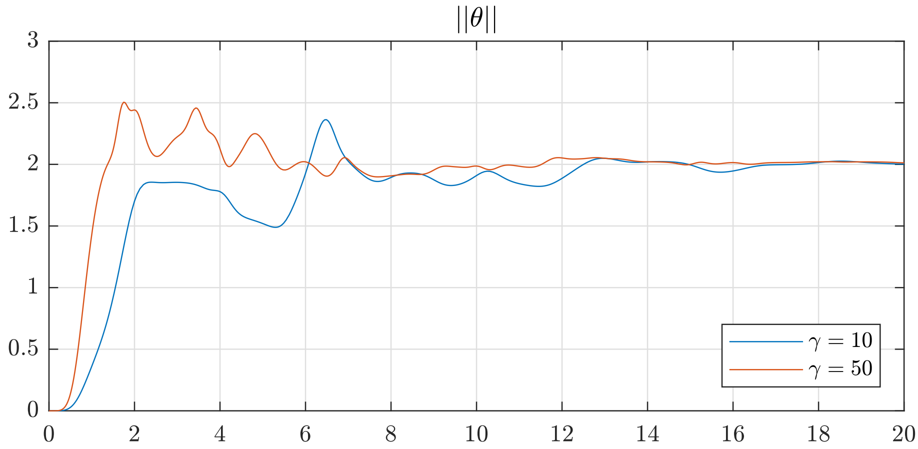
$$
\begin{aligned}
P(s) &= \begin{bmatrix} \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s^2} \end{bmatrix} K_p, \\
K_p &= \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -h\sin(\phi) & h\cos(\phi) \end{bmatrix} \\
M(s) &= \begin{bmatrix} \frac{\lambda^2}{s+\lambda^2} & 0 \\ 0 & \frac{\lambda^2}{s+\lambda^2} \end{bmatrix} \\
r_1 &= \sin(0.635t) + \sin(4.567t) \\
r_2 &= \sin(0.1t) + \sin(1.1t)
\end{aligned}
$$

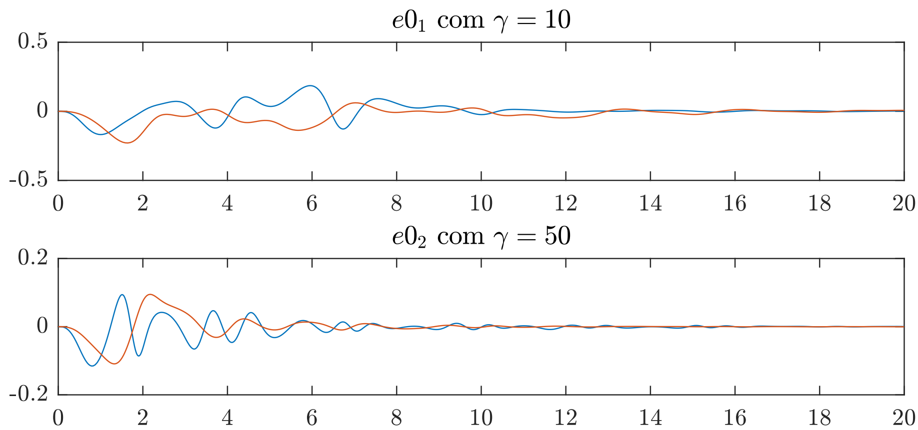
### 3.1.1 Simulação #1

Inicialmente, verificamos o comportamento do sistema para variações no **parâmetro de adaptação** $\Gamma$.

$$
\phi = \frac{\pi}{4}, \qquad h = 1,
$$

$$
y(0) = \mathbf{0}, \qquad \gamma = 10 \text{, e } 50
$$

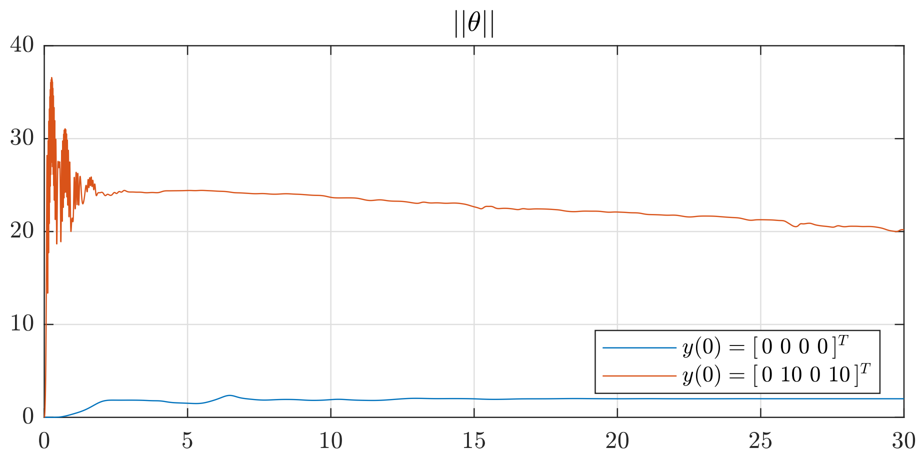


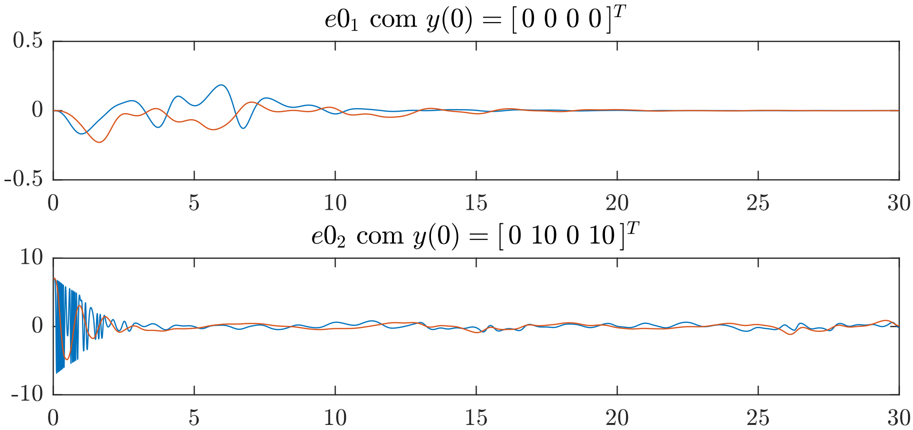




### 3.1.2 Simulação #2

Verificamos agora o comportamento do sistema para variações na **condição inicial** $y(0)$.

$$\phi = \frac{\pi}{4}\,, \qquad h = 1\,,$$
$$y(0) = \mathbf{0}\ e\ \mathbf{10}\,, \qquad \gamma = 10$$



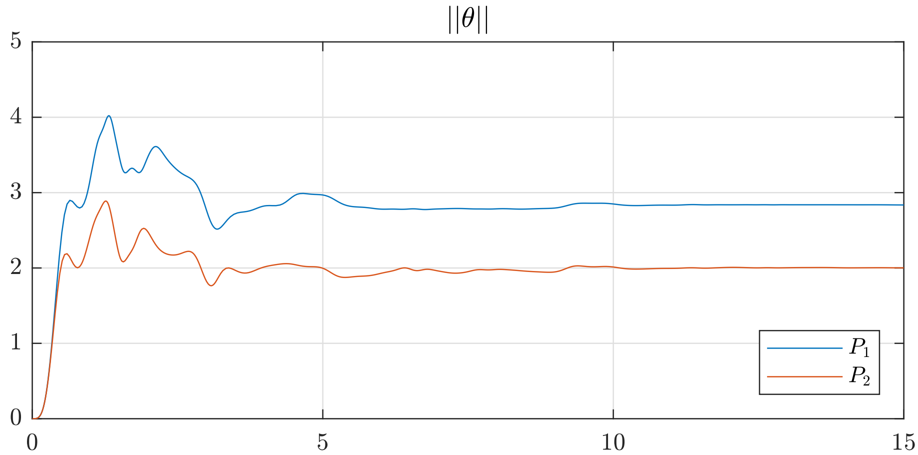


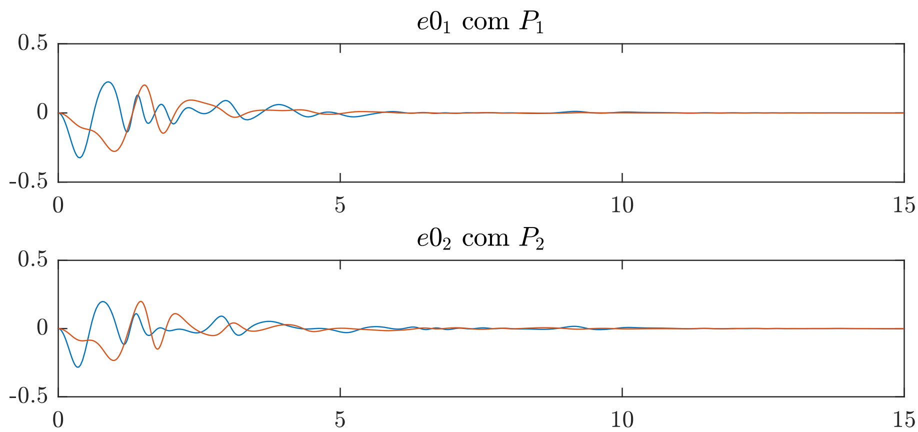

### 3.1.3 Simulação #3

Verificamos o comportamento do sistema para variações na **função de transferência da planta** $P(s)$.

$$\phi = \frac{\pi}{4} \; e \; \frac{\pi}{3}\,, \qquad h = 1 \; e \; 2\,,$$

$$y(0) = \mathbf{0}\,, \qquad \gamma = 10$$



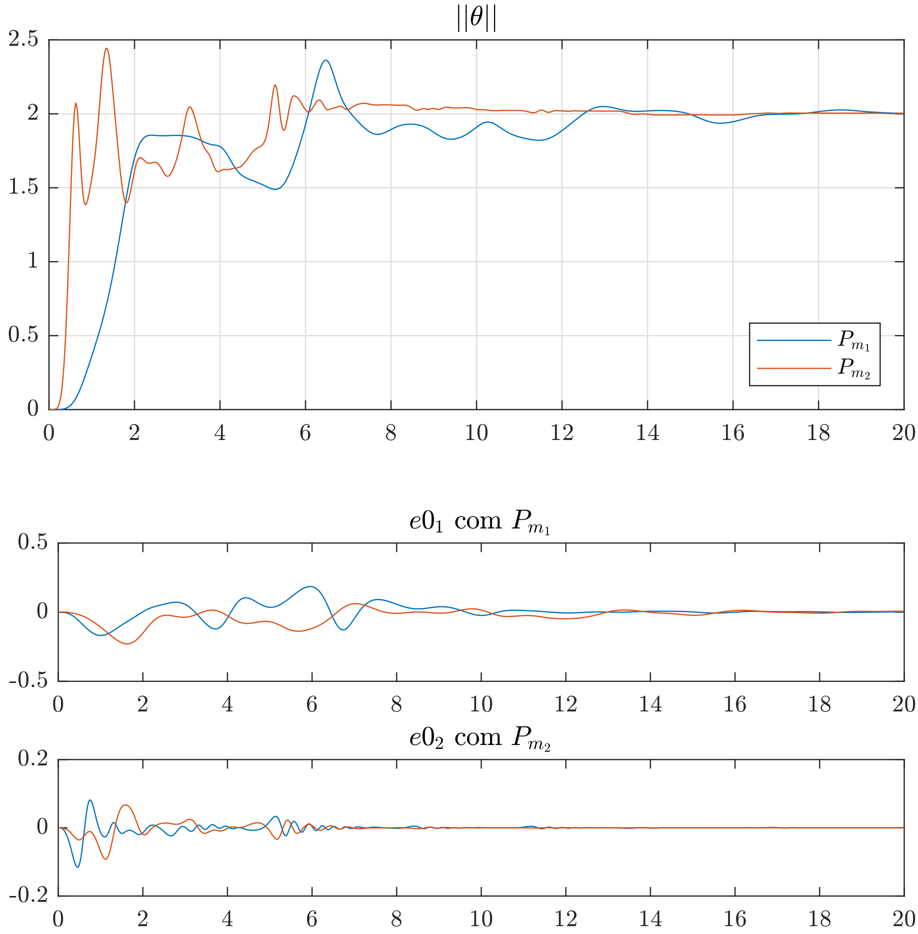



### 3.1.4 Simulação #4

Verificamos o comportamento do sistema para variações na **função de transferência do modelo de referência** $P_m(s)$.

$$\phi = \frac{\pi}{4}\,, \qquad h = 1$$

$$y(0) = \mathbf{0}\,, \qquad \gamma = 10\,, \qquad \lambda = 1 \; e \; 2$$




## 3.2 Caso em que todos os parâmetros são desconhecidos

$$P(s) = \begin{bmatrix} \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s^2} \end{bmatrix} K_p,$$

$$K_p = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -h\sin(\phi) & h\cos(\phi) \end{bmatrix}$$

$$M(s) = \begin{bmatrix} \frac{\lambda^2}{s+\lambda^2} & 0 \\ 0 & \frac{\lambda^2}{s+\lambda^2} \end{bmatrix}$$

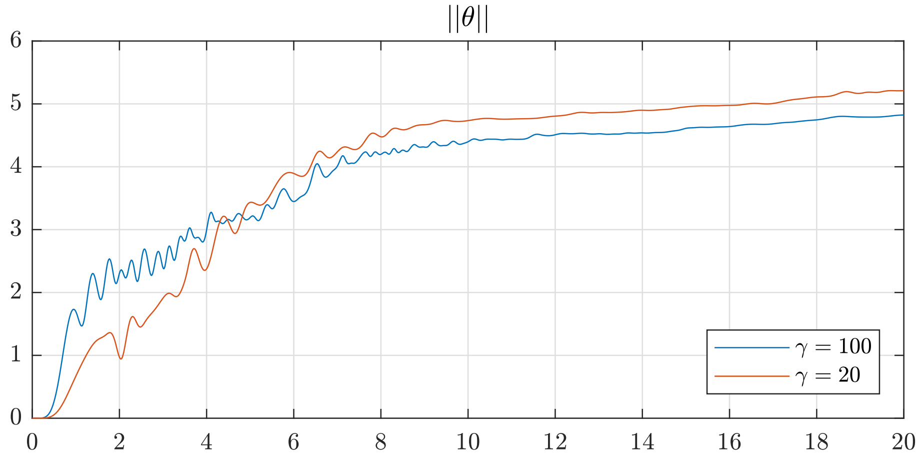
$$r_1 = \sin(0.635t) + \sin(4.567t)$$

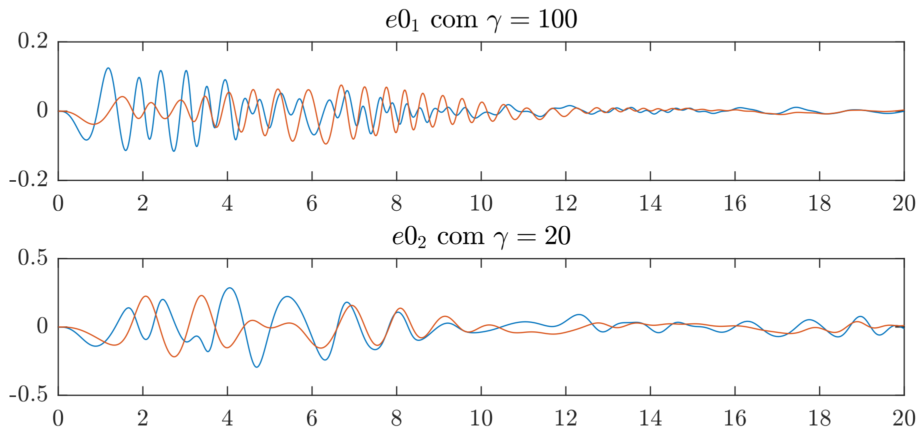
$$r_2 = \sin(0.1t) + \sin(1.1t)$$

### 3.2.1 Simulação #1

Inicialmente, verificamos o comportamento do sistema para variações no **parâmetro de adaptação** $\Gamma$.

$$\phi = \frac{\pi}{4}\,, \qquad h = 1\,,$$
$$y(0) = \mathbf{0}\,, \qquad \gamma = 100\,, \text{e } 20$$

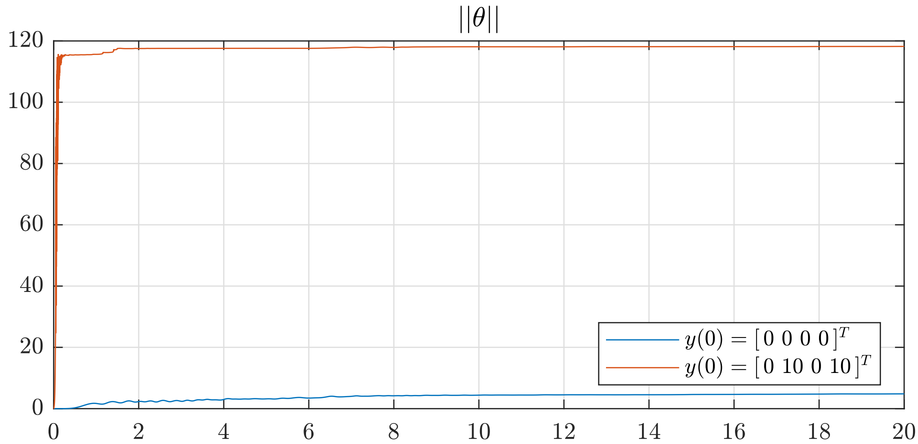


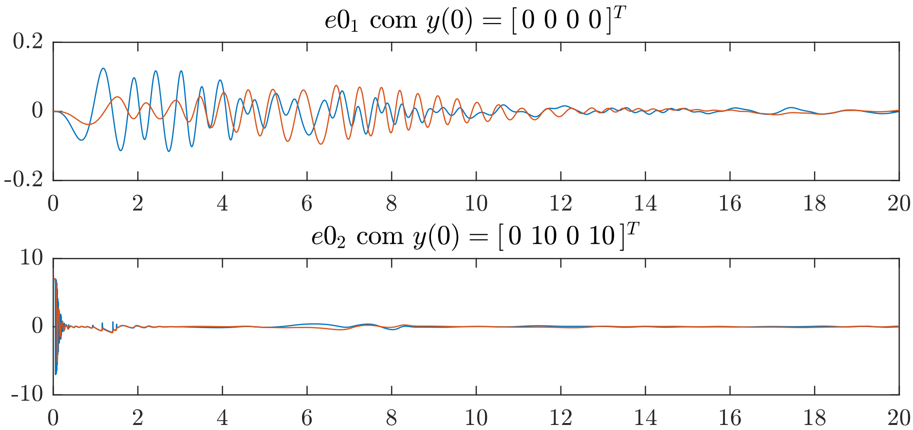




### 3.2.2 Simulação #2

Verificamos agora o comportamento do sistema para variações na **condição inicial** $y(0)$.

$$\phi = \frac{\pi}{4}\,, \qquad h = 1\,,$$
$$y(0) = \mathbf{0} \; e \; \mathbf{10}\,, \qquad \gamma = 100$$



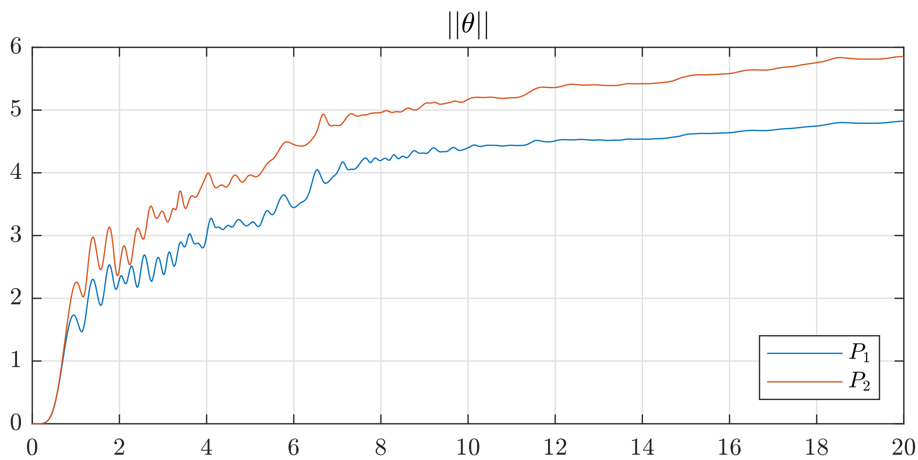



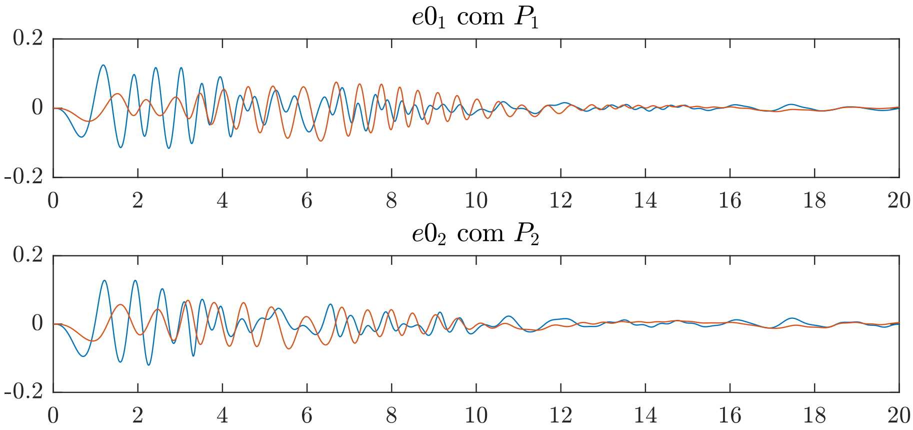

### 3.2.3 Simulação #3

Verificamos o comportamento do sistema para variações na **função de transferência da planta** $P(s)$.

$$\phi = \boxed{\frac{\pi}{4}}\ e\ \boxed{\frac{\pi}{3}}\,, \qquad h = \boxed{1}\ e\ \boxed{2}\,,$$

$$y(0) = \mathbf{0}\,, \qquad \gamma = 100$$



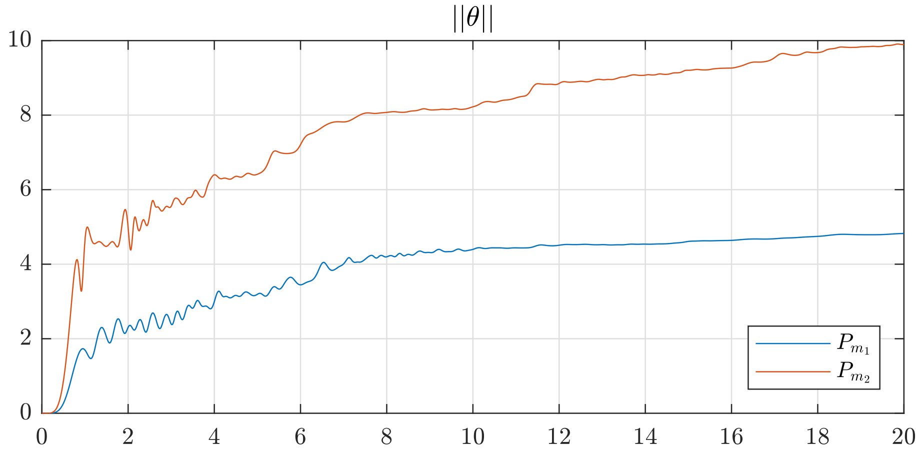


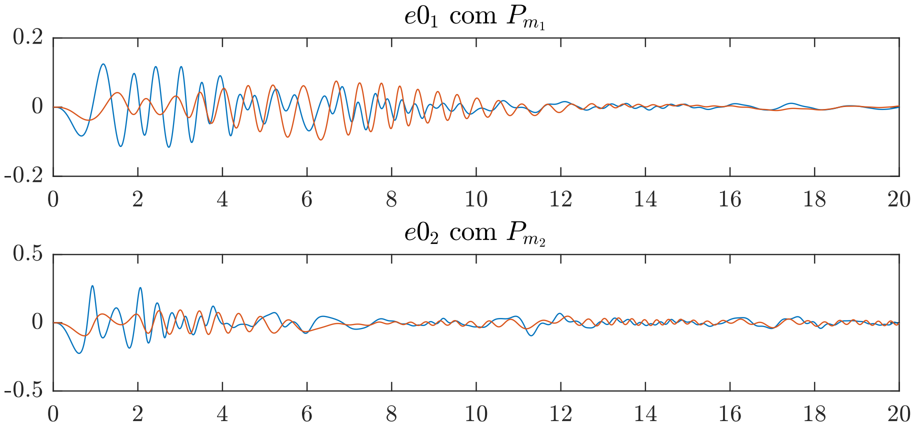

### 3.2.4 Simulação #4

Verificamos o comportamento do sistema para variações na **função de transferência do modelo de referência** $P_m(s)$.

$$
\phi = \frac{\pi}{4}\,, \qquad h = 1
$$

$$
y(0) = \mathbf{0}\,, \qquad \gamma = 100\,, \qquad \lambda = 1 \; e \; 2
$$



# 4 Discussão

Os resultados apresentados neste trabalho são semelhantes aos apresentados nos trabalhos anteriores sobre MRAC, só que aplicado a sistemas MIMO. Neste trabalho, são abordados sistemas com duas entradas e duas saídas e plantas de segunda ordem com grau relativo 2. O filtro foi definido como mais simples possível.

Comparativamente, quando apenas $K_p$ é desconhecido (5 parâmetros), o erro converge rapidamente para zero, aproximadamente 15 segundos, usando ganhos de adaptação $\Gamma$ entre 10 e 50. O erro leva mais tempo para convergir para zero quando todos os parâmetros são desconhecidos (17 parâmetros), aproximadamnte 30 segundos, o ganho de adaptação deve ser bem maior (100), e o tempo de simulação é, também, bem mais elevado, pois a *ode* calcula a derivada de 17 parâmetros.

A **simulação #1** mostra o comportamento do sistema para variações no ganho de adaptação $\Gamma$. Como nos trabalhos anteriores, aumentar o ganho de adaptação garante convergência mais rápida, porém mais oscilações.

A **simulação #2** mostra o comportamento do sistema para variações nas condições iniciais da planta $y(0)$. O sistema oscila mais para condições iniciais da planta longe de zero. O transitório, porém, foi menor para os sistemas com condição inicial diferente de zero. Isso ocorre, pois os parâmetros entraram rapidamente em quadratura com o vetor de regressor.

A **simulação #3** mostra o comportamento do sistema para variações no ganho de adaptação na planta. A planta com maior deslocamento, $\phi = \frac{\pi}{3}$, teve melhor transitório e convergiu mais rápido, mostrando que é difícil prever o comportamento comparativo quando alteramos a planta.

A **simulação #4** mostra o comportamento do sistema para variações no modelo. Modelos mais rápidos apresentaram maiores oscilações.