Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Engenharia Elétrica

Tópicos em Controle Moderno

Laboratório - Controle Adaptativo

Prof. Dr. Tatiana Pazelli
Professor Responsável

Integrantes do Grupo

Gabriel Souza Barbosa	RA: 790852
Giovanna Amorim Nascimento	RA: 784267
João Carlos Tonom Campi	RA: 769723

26 de julho de 2024

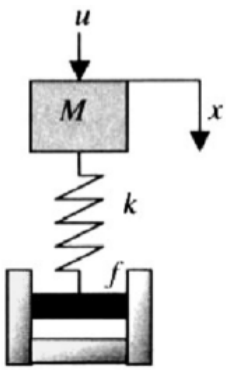
Lista de Figuras

Sumário

1. Experiência 1 - Modelos paramétricos

Seja o sistema massa-mola-amortecedor e seu modelo dinâmico apresentados na figura a seguir:

Figura 1: Sistema massa-mola-amortecedor



$$\boxed{M\ddot{x} = u - kx - f\dot{x}}$$

Fonte: Roteiro Laboratório 1

tal que k é a constante da mola, f é o coeficiente de amortecimento, M é a massa do sistema, u é a força de entrada e x é o deslocamento da massa M. Para as experiências 1, 2 e 3 a seguir considere $M = 100kg$, $f = 0,15kg/s$, $k = 7kg/s2$, $u(t) = 1 + cos(\frac{\pi}{3}t)$ e $0 \leq t \leq 25s$.

(a) Determine um modelo paramétrico admitindo k, f e M parâmetros constantes desconhecidos, sendo x e u disponíveis a cada instante t. Gere os sinais z e ϕ do modelo paramétrico.

(b) Determine um modelo paramétrico admitindo k e f parâmetros constantes desconhecidos e $M = 100kg$ um parâmetro conhecido, sendo x e u disponíveis a cada instante t. Gere os sinais z e ϕ do modelo paramétrico.

1.1. Resolução

1.1.1. a)

Considerando os parâmetros desconhecidos e sabendo que o modelo paramétrico pode ser dado da forma:

$$z = \theta^{*T}\phi \tag{1}$$

Admitindo-se x, $\dot{x}$, $\ddot{x}$ e u mensuráveis ,teremos:

$$u = \begin{pmatrix} M & f & k \end{pmatrix} \cdot \begin{pmatrix} \ddot{x} \\ \dot{x} \\ x \end{pmatrix} \tag{2}$$

sendo:

$$z = u \quad ; \quad \theta^* = \begin{pmatrix} M & f & k \end{pmatrix}^T \quad ; \quad \phi = \begin{pmatrix} \ddot{x} \\ \dot{x} \\ x \end{pmatrix} \tag{3}$$

Nota-se que, para a obtenção dos parâmetros $\dot{x}$ e $\ddot{x}$ será necessário aplicar um filtro de segunda ordem da forma:

$$\frac{1}{(s+\lambda)^2} \tag{4}$$

Portanto, um modelo paramétrico que representa o sistema será:

$$\frac{Ms^2x + Fsx + Kx}{(s+\lambda)^2} = \frac{u}{(s+\lambda)^2} \Rightarrow \frac{u}{(s+\lambda)^2} = \begin{pmatrix} M & f & k \end{pmatrix} \cdot \begin{pmatrix} \frac{s^2}{(s+\lambda)^2}x \\ \frac{s}{(s+\lambda)^2}x \\ \frac{1}{(s+\lambda)^2}x \end{pmatrix} \tag{5}$$

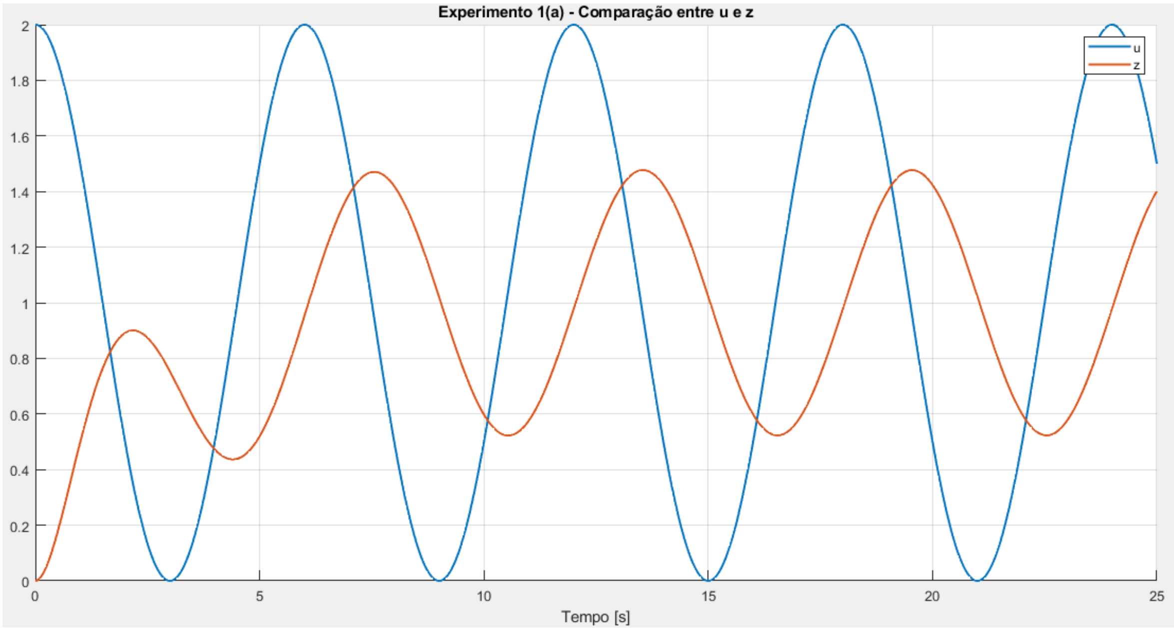
Logo abaixo é possível observar o script em Matlab responsável pela implementação do modelo paramétrico do sistema e obtenção do sinal z em comparação com a entrada original u, além do plot do sinal ϕ.

```
% Experimento 1
% Parametros do problema
M = 100;
f = 0.15;
k = 7;

t = linspace(0,25,1000); % Vetor de tempo
u = 1+cos(pi*(t/3)); % Sinal de entrada

lambda = 1; % Parametro do filtro

G = tf(1, [M f k]); % Resposta original da planta
x = lsim(G, u, t);

```

```
%Item (a)

s = tf("s");
filtro = tf(1 , [1 2*lambda lambda^2]); % Declaracao do filtro
z = lsim(filtro, u, t); % Sinal z do modelo parametrico

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot (t, u)
plot (t, z)
title ("Experimento 1(a) - Comparacao entre u e z")
legend ('u', 'z')
xlabel ('Tempo [s]')
hold off

% Aplicacoes de filtro para o vetor phi
xdd_f = lsim(s^2*filtro, x, t);
xd_f = lsim(s*filtro, x, t);
x_f = lsim(filtro, x, t);

phi = [xdd_f xd_f x_f]'; %Vetor phi com filtro

index_fig = index_fig + 1;
figure (index_fig)
hold on
grid on
plot (t, phi)
title ("Experimento 1(a) - Sinal \phi")
hLeg = legend ('$\ddot{x}$', "$\dot{x}$", '$x$');
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]')
```

Na Figura 2 é possível observar a representação do sinal z em comparação com a entrada original do sistema u. A amplitude do sinal de z varia de acordo com o valor de λ definido. Para este teste foi utilizado $\lambda = 1$.

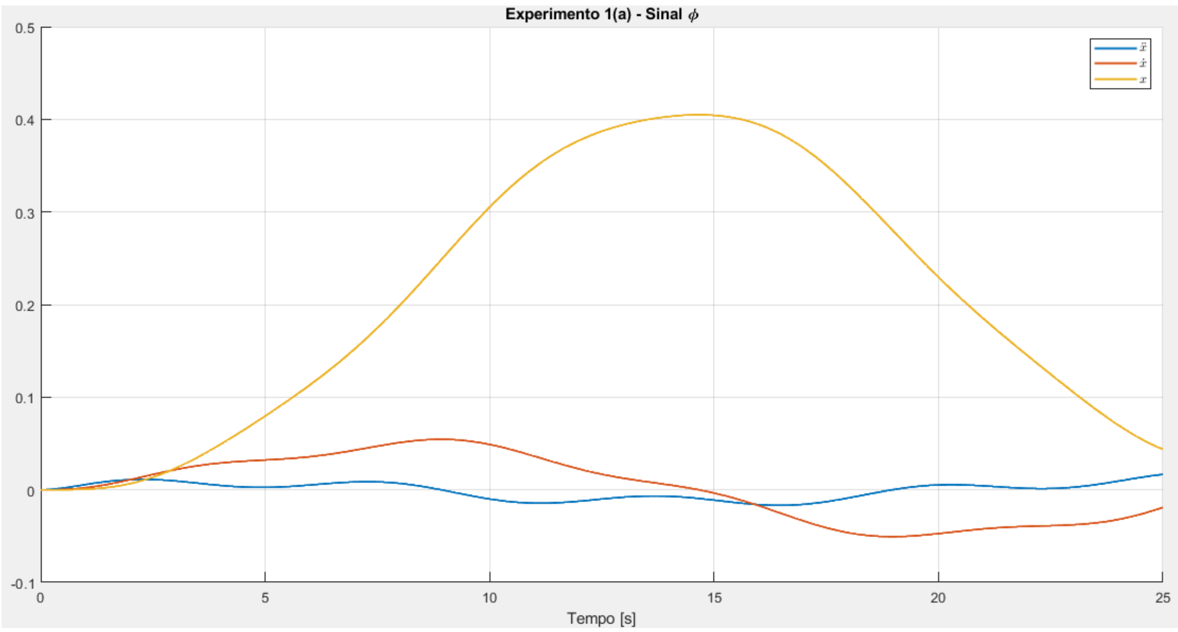
Figura 2: Comparação dos sinais z e u



Fonte: Autoria própria

Por fim, na Figura 3 consta o plot do sinal ϕ obtido:

Figura 3: Sinal ϕ do modelo paramétrico



Fonte: Autoria própria

1.1.2. b)

Agora, considerando o mesmo raciocínio do exercício anterior, porém com o parâmetro $M = 100$ conhecido, a representação do modelo paramétrico do sistema será:

$$u - M\ddot{x} = kx + f\dot{x} \;\Rightarrow\; u - M\ddot{x} \;=\; \begin{pmatrix} f & k \end{pmatrix} \cdot \begin{pmatrix} \dot{x} \\ x \end{pmatrix} \tag{6}$$

Aplicando o filtro de segunda ordem do tipo $\frac{1}{(s+\lambda)^2}$ para acessar todos os parâmetros desejados, teremos:

$$\frac{u - Ms^2x}{(s+\lambda)^2} = \begin{pmatrix} f & k \end{pmatrix} \cdot \begin{pmatrix} \frac{s}{(s+\lambda)^2}x \\ \frac{1}{(s+\lambda)^2}x \end{pmatrix} \tag{7}$$

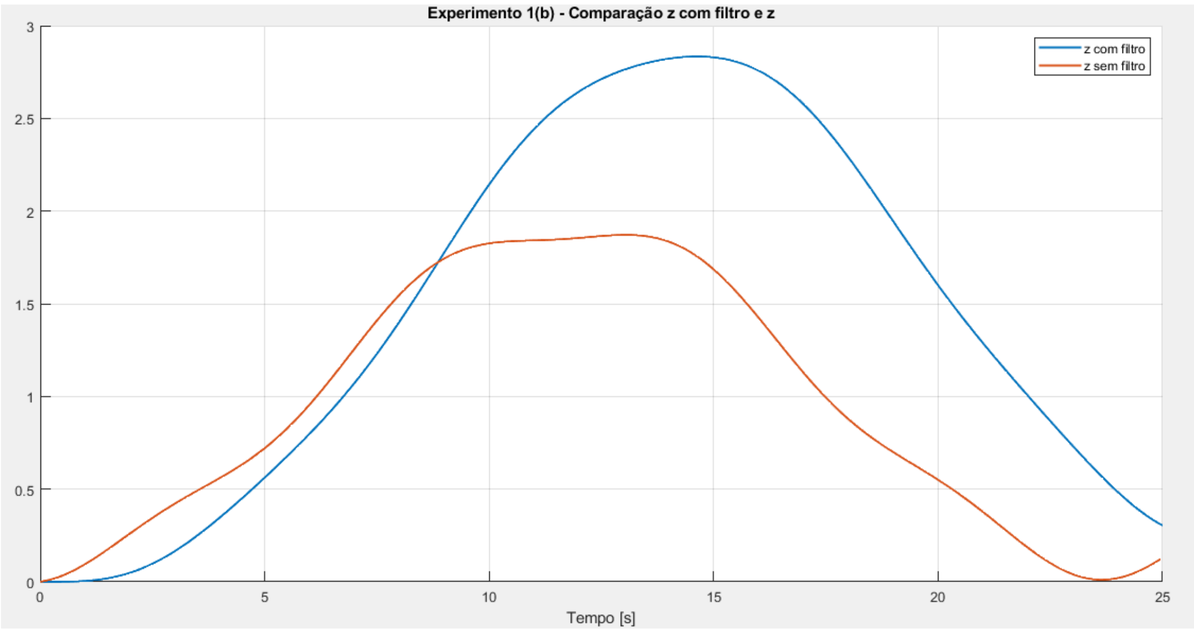
No script abaixo consta a implementação do novo modelo paramétrico e a obtenção dos novos sinais de z e ϕ.

```
%Item (b)
% Novo modelo parametrico
z_b = z - M*xdd_f; % Novo sinal de z com M conhecido
phi_b = [xd_f x_f]'; % Vetor phi com filtro

% Derivadas do sinal de saida sem filtro
xd= diff(x)./(diff(t)');
xdd= diff(xd)./(diff(t(1,1:end-1))');

z_sf= (u(1,1:end-2))' - M.*xdd;

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot(t, z_b)
plot(t(1,1:end-2), z_sf)
title ("Experimento 1(b) - Comparacao z com filtro e z")
legend ('z com filtro', 'z sem filtro')
xlabel ('Tempo [s]')

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot(t, phi_b)
```

```
title ("Experimento 1(b) -Sinal \phi")
hLeg = legend ("$\dot{x}$", '$x$');
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]')
```

Na Figura 4 é possível observar a representação gráfica do sinal z. Nota-se que foi realizada uma comparação do sinal com e sem a presença do filtro. O valor de λ foi mantido em $\lambda = 1$.

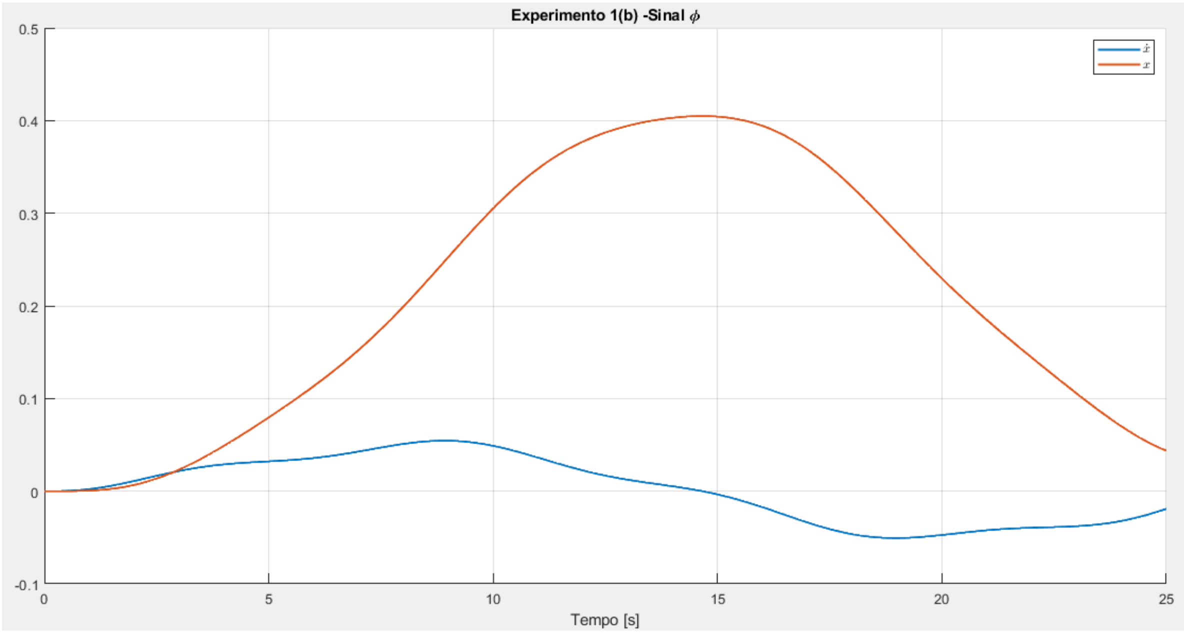
Figura 4: Sinal z do modelo atualizado



Fonte: Autoria própria

Além disso, na Figura 5 consta a representação do sinal ϕ do novo modelo paramétrico.

Figura 5: Sinal ϕ do modelo paramétrico atualizado



Fonte: Autoria própria

2. Experiência 2 - Modelos Paramétricos

Seja o sistema de direcionamento do telescópio Hubble representado pelo seguinte modelo dinâmico:

$$K_1 \ddot{y}(t) + K_2 \dot{y}(t) + K_3 y(t) = r(t) \tag{8}$$

tal que $K_1 = 1$; $K_2 = 12$ e $K_3 = 100$. Considere a trajetória de direcionamento da visada do telescópio dada por $r(t) = 2.7cos\left(4 \cdot 10^{-7} t\right)$

(a) Determine um modelo paramétrico admitindo K_1, K_2 e K_3 parâmetros constantes desconhecidos, sendo $y(t)$ e $r(t)$ disponíveis a cada instante t. Gere os sinais z e ϕ do modelo paramétrico.

(b) Admita agora que $z(t) = \frac{y(t)}{(s+\lambda)^2}$. Considerando K_1, K_2 e K_3 parâmetros constantes desconhecidos, sendo $y(t)$ e $r(t)$ disponíveis a cada instante t, determine um modelo paramétrico para o sistema proposto. Gere os sinais z e ϕ do modelo paramétrico.

(c) Determine um modelo paramétrico admitindo K_1 e K_3 parâmetros constantes desconhecidos e $K_2 = 12$ um parâmetro conhecido, sendo $y(t)$ e $r(t)$ disponíveis a cada instante t. Gere os sinais z e ϕ do modelo paramétrico.

2.1. Resolução:

2.1.1. a)

Para a resolução dessa questão foi considerado modelo paramétrico estático (SPM), conforme apresentado na Equação 1.

Dada a função de transferência apresentado e considerando todos os estados como acessíveis, o modelo pode ser escrito conforme apresentado na Equação 9:

$$r(t) = \begin{pmatrix} K_1 & K_2 & K_2 \end{pmatrix} \cdot \begin{pmatrix} \ddot{y}(t) \\ \dot{y}(t) \\ y(t) \end{pmatrix} \tag{9}$$

No entanto como somente $y(t)$ e $r(t)$ estão disponíveis em cada instante t, é necessário aplicar um filtro para mensurar $\dot{y}(t)$ e $\ddot{y}(t)$. O filtro aplicado é apresentado na Equação 10:

$$\frac{1}{(s+\lambda)^2} \tag{10}$$

Como o filtro a ser aplicado está no domínio da frequência, é necessário aplicar a transformada de *Laplace* no sistema para deixá-lo no mesmo domínio. Assim, o sistema a ser modelado será conforme apresentado na Equação 11:

$$K_1 s^2 Y + K_2 s Y + K_3 Y = R \tag{11}$$

Com isso, o modelo paramétrico do sistema pode ser descrito conforme a Equação 12:

$$\frac{r}{(s+\lambda)^2} = \begin{pmatrix} K_1 & K_2 & K_3 \end{pmatrix} \cdot \begin{pmatrix} \frac{s^2}{(s+\lambda)^2} y \\ \frac{s}{(s+\lambda)^2} y \\ \frac{1}{(s+\lambda)^2} y \end{pmatrix} \tag{12}$$

Na qual:

$$z = \frac{r}{(s+\lambda)^2};\ \theta^{*T} = \begin{pmatrix} K_1 & K_2 & K_3 \end{pmatrix} \text{ e } \phi = \begin{pmatrix} \frac{s^2}{(s+\lambda)^2} y \\ \frac{s}{(s+\lambda)^2} y \\ \frac{1}{(s+\lambda)^2} y \end{pmatrix}$$

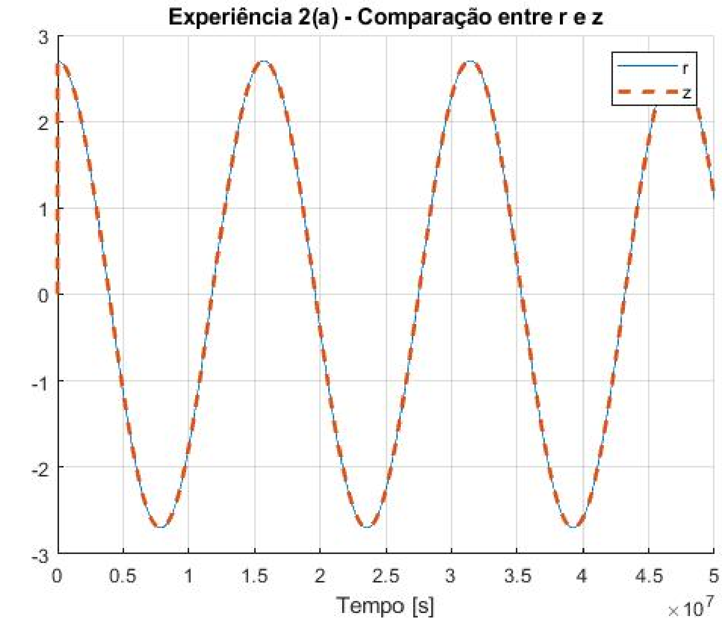
Para a implementação do modelo foi utilizado o script apresentado a seguir.

```

% Experiencia 2
% Parametros do problema
K1= 1;
K2= 12;
K3= 100;

t= linspace(1,5e7,10e5); % Vetor de tempo
r= 2.7*cos(4e-7*t); % Sinal de entrada

lambda= 1; % Parametro do filtro

G2= tf(1, [K1 K2 K3]); % FT real
y= lsim(G2, r, t); % Simulacao original do problema

%Item (a)

s= tf("s");

filtro = tf(1 , [1 2*lambda lambda^2]); % Implementacao do filtro
z= lsim(filtro, r, t); % Sinal z do modelo parametrico
```

```
index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot (t, r)
plot (t, z, '--', 'LineWidth', 2)
title ("Experiencia 2(a) - Comparacao entre r e z")
legend ('r', 'z')
xlabel ('Tempo [s]')
hold off

% Sinais obtidos a partir do filtro
ydd_f= lsim(s^2*filtro, y, t);
yd_f= lsim(s*filtro, y, t);
y_f= lsim(filtro, y, t);

phi= [ydd_f yd_f y_f]';  % Vetor phi com filtro

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot (t, phi)
title (" Experiencia 2(a) - Sinal \phi")
hLeg = legend ('$\ddot{y}$', "$\dot{y}$", '$y$');
set(hLeg,'Interpreter','latex');
xlabel ('tempo (s)')
```

A Figura 6 mostra o sinal z em comparação com a entrada original do sistema r. A resposta do sinal z muda de acordo com o valor de λ. Após alguns testes de variação desse parâmetro, foi definido $\lambda = 1$ para a resposta apresentada.

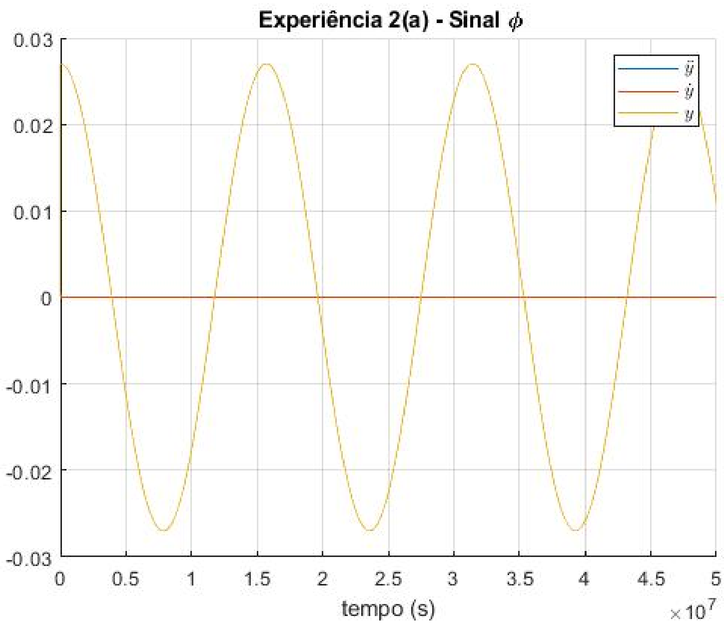
Figura 6: Comparação dos sinais z e r



Fonte: Autoria própria

A Figura 7 apresenta o sinal ϕ obtido da representação paramétrica utilizada.

Figura 7: Sinal ϕ do modelo paramétrico



Fonte: Autoria própria

2.1.2. b)

Para o item **b**, foram realizadas algumas manipulações no modelo dinâmico apresentado na Equação 8. A Equação 13 mostra o modelo dinâmico utilizado.

$$K_1 \ddot{y}(t) + K_2 \dot{y}(t) + K_3 y(t) = r(t)$$

$$-K_1 \ddot{y}(t) - K_2 \dot{y}(t) + r(t) = K_3 y(t))$$

$$-\frac{K_1}{K_3}\ddot{y}(t) - \frac{K_2}{K_3}\dot{y}(t) + \frac{1}{K_3} r(t) = y(t) \tag{13}$$

Com isso, o modelo paramétrico pode ser descrito conforme apresentado na Equação 14:

$$y(t) = \begin{pmatrix} \frac{K_1}{K_3} & \frac{K_2}{K_3} & \frac{1}{K_3} \end{pmatrix} \cdot \begin{pmatrix} -\ddot{y}(t) \\ -\dot{y}(t) \\ r(t) \end{pmatrix} \tag{14}$$

Assim como no item anterior, é necessário aplicar um filtro do tipo $\frac{1}{(s+\lambda)^2}$, para que os sinais de $\dot{y}(t)$ e $\ddot{y}(t)$ possam ser estimados. Desse modo, o modelo paramétrico final é dado pela Equação 15.

$$\frac{y}{(s+\lambda)^2} = \begin{pmatrix} \frac{K_1}{K_3} & \frac{K_2}{K_3} & \frac{1}{K_3} \end{pmatrix} \cdot \begin{pmatrix} -\frac{s^2}{(s+\lambda)^2} y \\ -\frac{s}{(s+\lambda)^2} y \\ \frac{1}{(s+\lambda)^2} r \end{pmatrix} \tag{15}$$

Na qual:

$$z = \frac{y}{(s+\lambda)^2};\ \theta^{*T} = \begin{pmatrix} \frac{K_1}{K_3} & \frac{K_2}{K_3} & \frac{1}{K_3} \end{pmatrix} \text{ e } \phi = \begin{pmatrix} -\frac{s^2}{(s+\lambda)^2} y \\ -\frac{s}{(s+\lambda)^2} y \\ \frac{1}{(s+\lambda)^2} r \end{pmatrix}$$

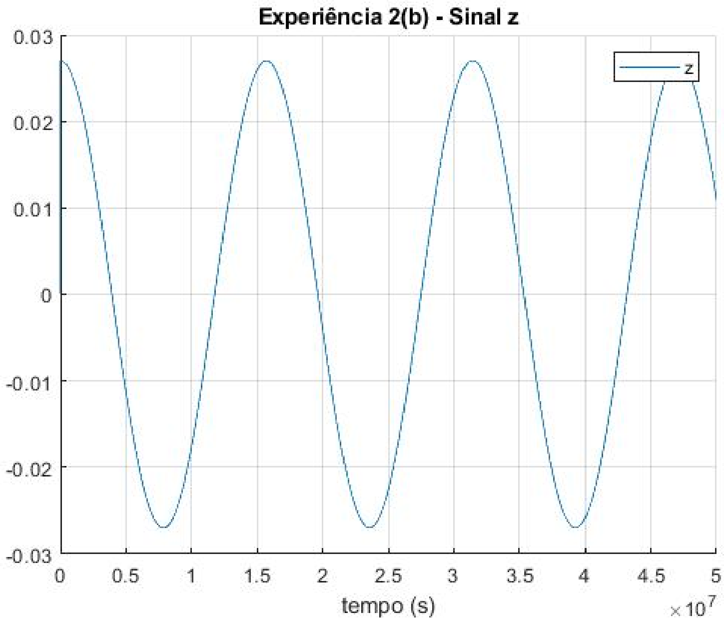
Para a implementação do modelo foi utilizado o script apresentado a seguir.

```

% Experiencia 2 - Item (b)
z_b= lsim(filtro, y, t);

ydd_f= lsim(s^2*filtro, y, t);
yd_f= lsim(s*filtro, y, t);
```

```

r_f = lsim(filtro, r, t);

phi_b= [ydd_f yd_f r_f]';

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot(t, z_b)
title ("Experiencia 2(b) - Sinal z")
legend ('z')
xlabel ('tempo (s)')

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot(t, phi_b)
title ("Experiencia 2(b) - Sinal \phi")
hLeg = legend ('$\ddot{y}$', "$\dot{y}$", '$r$');
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]')
```

A Figura 8 mostra o sinal z do modelo paramétrico da Equação 15, na qual o valor de λ foi definido como sendo 1.

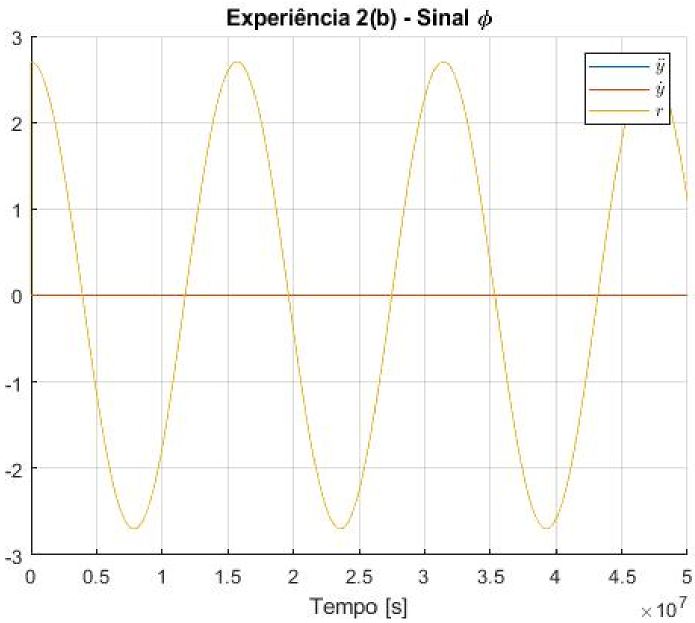
Figura 8: Sinal z do modelo paramétrico



Fonte: Autoria própria

Na Figura 9, são apresentados os sinais da matriz ϕ, considerando $\lambda = 1$.

Figura 9: Sinal ϕ do modelo paramétrico



Fonte: Autoria própria

2.1.3. c)

De forma análoga ao que foi realizado no item **a**, considerando o modelo paramétrico dinâmico da Equação 1 e o parâmetro $K_2 = 12$, também conhecido, foi obtido o modelo apresentado na Equação 16 apresentada.

$$r(t) - K_2\dot{y}(t) = \begin{pmatrix} K_1 & K_3 \end{pmatrix} \cdot \begin{pmatrix} \ddot{y}(t) \\ y(t) \end{pmatrix} \tag{16}$$

Assim como nos itens anteriores, é adicionado um filtro afim de estimar os sinais $\dot{y}(t)$ e $\ddot{y}(t)$. Com isso, o modelo paramétrico final é dado pela Equação 17.

$$\frac{r}{(s+\lambda)^2} - \frac{K_2 \cdot s}{(s+\lambda)^2} y = \begin{pmatrix} K_1 & K_3 \end{pmatrix} \cdot \begin{pmatrix} \frac{s^2}{(s+\lambda)^2} y \\ \frac{1}{(s+\lambda)^2} y \end{pmatrix} \tag{17}$$

Na qual:

$$z = \frac{r}{(s+\lambda)^2} - \frac{K_2 \cdot s}{(s+\lambda)^2} y;\ \theta^{*T} = \begin{pmatrix} K_1 & K_3 \end{pmatrix} \text{ e } \phi = \begin{pmatrix} \frac{s^2}{(s+\lambda)^2} y \\ \frac{1}{(s+\lambda)^2} y \end{pmatrix}$$

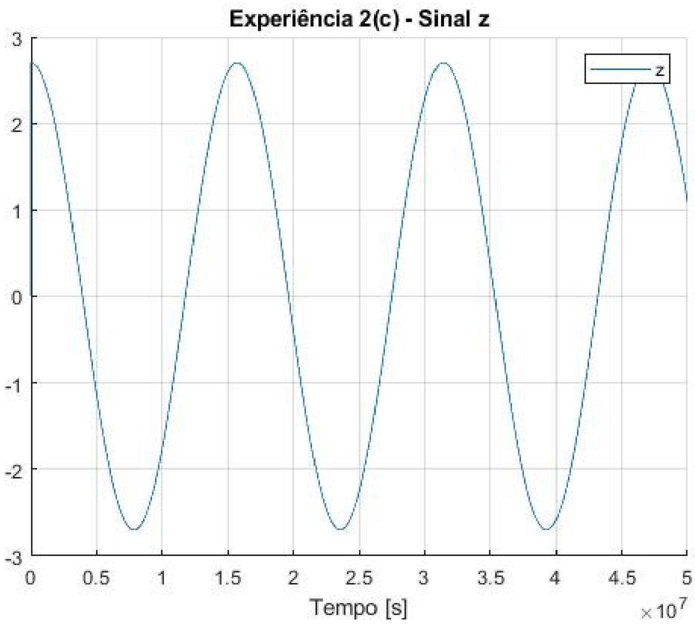
Para a implementação do modelo foi utilizado o script apresentado a seguir.

```

% Experiencia 2 - Item (c)
z_c= z - K2*yd_f;  % Sinal z com K2 conhecido
phi_c= [ydd_f y_f]; % Sinal phi com K2 conhecido

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot(t, z_c)
title ("Experiencia 2(c) - Sinal z")
legend ('z')
xlabel ('Tempo [s]')

index_fig= index_fig + 1;
figure (index_fig)
hold on
grid on
plot(t, phi_c)
title ("Experiencia 2(c) - Sinal \phi")
hLeg = legend ("$\ddot{y}$", '$y$');
```

```
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]')
```

Na Figura 10 é apresentado o sinal z para $\lambda = 1$.

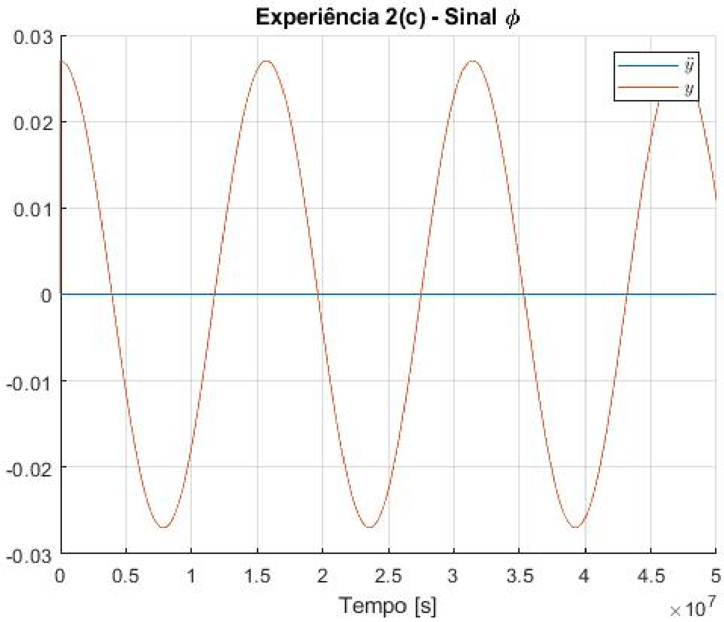
Figura 10: Sinal z do modelo paramétrico



Fonte: Autoria própria

Na Figura 11 são apresentados os sinais da matriz ϕ. É possível notar que ela possui um sinal a manos que as anteriores, pois nesse caso há um parâmetro desconhecido a menos.

Figura 11: Sinal ϕ do modelo paramétrico



Fonte: Autoria própria

3. Experiência 3 – Modelos Paramétricos

Analisando os resultados obtidos nas Experiências 1 e 2, responda:

(a) Por que foi aplicado um filtro para determinar o modelo paramétrico do sistema proposto?

(b) Como você determinou a ordem do filtro?

(c) Qual a consequência da introdução deste filtro para a resposta do sistema?

(d) O que acontece se aumentarmos o valor do polo do filtro? E se diminuirmos o valor do polo do filtro?

(e) Como você escolheu a variável z do modelo paramétrico? Por quê?

3.1. Resolução

3.1.1. a)

Para a determinação do modelo paramétrico das experiências 1 e 2, é necessário ter acesso aos parâmetros $\dot{x}$, $\ddot{x}$ e $\dot{y}$, $\ddot{y}$, respectivamente, porém inicialmente não é possível mensurar tais parâmetros a cada instante t. Por isso, faz-se necessário a adição de um filtro de 2º ordem da forma $\frac{1}{(s+\lambda)^2}$ em ambos os lados da equação e com $\lambda > 0$ para garantir a estabilidade do sistema.

3.1.2. b)

A ordem do filtro foi determinada de acordo com o estado de maior ordem não acessível do modelo a ser parametrizado.

3.1.3. c)

A resposta do sistema com a adição de um filtro geralmente resulta em um atraso e uma atenuação no sinal, na qual a sua influência dependerá sempre do valor de λ. Além disso, pode ser que a adição do filtro (em geral são polos) influencie diretamente na estabilidade do sistema.

3.1.4. d)

Quanto menor o valor do polo do filtro, pela resposta em frequência de um filtro de primeira ordem, maior é o seu ganho DC e o atraso aplicado no sinal. Quanto maior é seu valor, menos atraso é aplicado no sistema, mas em contrapartida maior será a atenuação do sinal.

3.1.5. e)

O objetivo do modelo paramétrico é identificar os parâmetros desconhecidos através dos sinais conhecidos, para isso, em todos os casos, foram isolados os sinais com parâmetros conhecidos, a esses foi definida a variável z.

4. Experiência 4 – Identificação de Parâmetros

Considere o sistema da Experiência 1. Projete um algoritmo de identificação de parâmetros baseado no método dos mínimos quadrados para estimar as constantes k, f e M, sendo x e u disponíveis a cada instante t. Gere os sinais de $\hat{k}$, $\hat{f}$ e $\hat{M}$.

4.1. Resolução

Retomando da Experiência 1, a equação do modelo paramétrico do sistema, considerando x e u disponíveis a cada instante t, é dado pela Equação 18.

$$\underbrace{\frac{u}{(s+\lambda)^2}}_{z} = \underbrace{\begin{bmatrix} M & f & k \end{bmatrix}}_{\theta^{*T}} \underbrace{\begin{bmatrix} \frac{s^2}{(s+\lambda)^2}x \\ \frac{s}{(s+\lambda)^2}x \\ \frac{1}{(s+\lambda)^2}x \end{bmatrix}}_{\phi} \tag{18}$$

Para realizar a estimativa dos parâmetros M, k e f, o método escolhido foi o dos Mínimos Quadrados Recursivo puro. Algumas expressões precisam ser definidas para a sua implementação, como a do erro ϵ e do fator de correção m_s são definidos pelas Equações 19 e 20.

$$\epsilon = \frac{z(t_i) - \hat{z}(t_i)}{(m_s)^2} \tag{19}$$

$$m_s = 1 + \phi(t_i)^T \phi(t_i) \tag{20}$$

sendo $z(t_i)$ o valor do sinal z original do modelo, $\hat{z}(t_i)$ o valor do sinal z a partir dos parâmetros estimados $\hat{M}$, $\hat{k}$ e $\hat{f}$, todos no instante de simulação t_i.

Além disso, as expressões que definem a variação dos parâmetros P e θ são dadas pelas Equações 21 e 22.

$$\dot{P} = -\frac{P\phi\phi^T P}{m_s^2} \tag{21}$$

$$\dot{\theta} = P\epsilon\phi(i) \tag{22}$$

O seu pseudocódigo para implementação em MATLAB pode ser visto na sequência pelo Algoritmo na sequência.

Algoritmo 1: Método dos Mínimos Quadrados Puro Recursivo

Require: t, z, ϕ
Ensure: $\hat{\theta}$

```
Definir P_0
Definir θ_0
for i ← 1, ..., length(t) do
    m_s ← 1 + φ(i)^T φ(i)
    Ṗ = −(P_0 φ(i) φ(i)^T P_0)/m_s^2
    P = Ṗ.dt + P_0
    ẑ(i) = θ_0 φ(i)
    ε = (z(i) − ẑ(i))/m_s^2
    θ̇ = P ε φ(i)
    θ = θ̇.dt + θ_0
    P_0 = P
    θ_0 = θ
end for
```

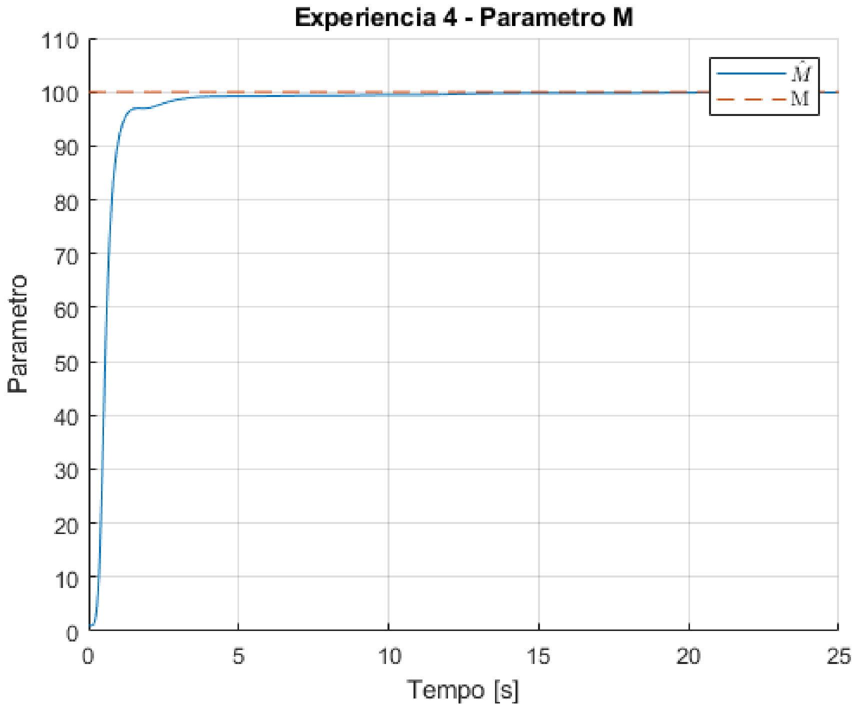
Os valores iniciais escolhidos arbitrariamente para o experimento, considerando o desempenho final obtido, são mostrados abaixo.

$$P_0 = 10000 \begin{bmatrix} 200 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

$$\theta_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

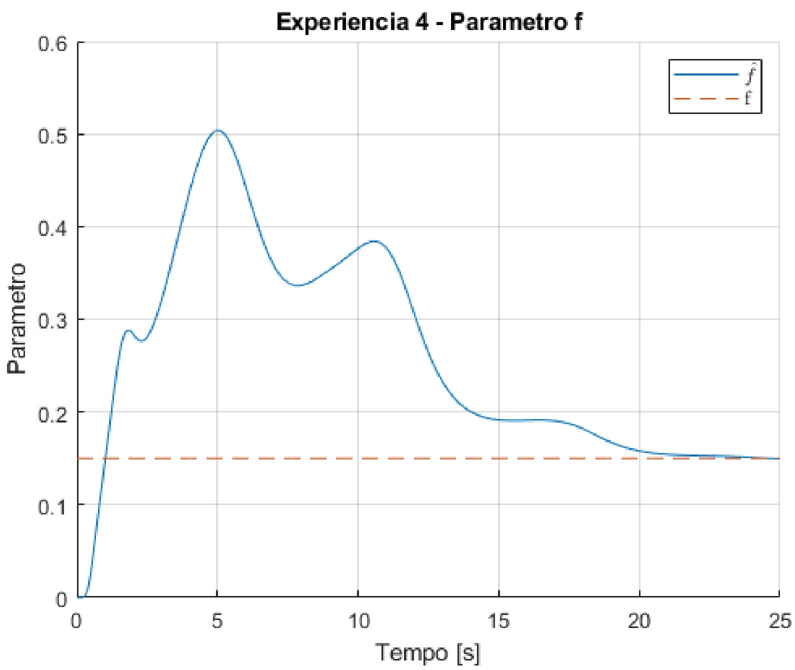
Aplicando o Método dos Mínimos Quadrados Puro Recursivo para o modelo massa-mola do Experimento 1, os gráficos dos parâmetros estimados $\hat{M}$, $\hat{f}$ e $\hat{k}$ são mostrados nas Figuras 12, 13 e 14, respectivamente.

Figura 12: Estimativa do parâmetro $\hat{M}$



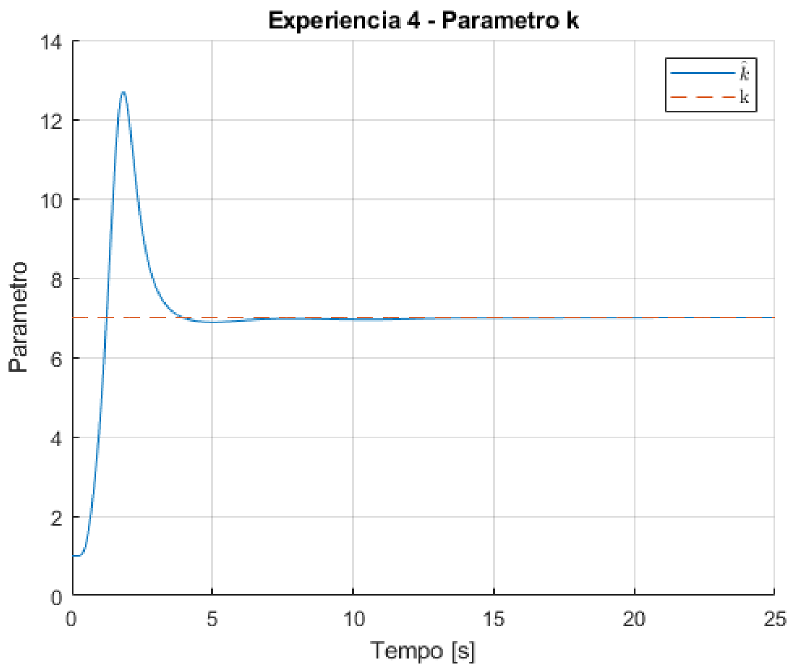
Fonte: Autoria própria

Figura 13: Estimativa do parâmetro $\hat{f}$



Fonte: Autoria própria

Figura 14: Estimativa do parâmetro $\hat{k}$



Fonte: Autoria própria

Como é possível notar pelos gráficos, todos os parâmetros foram encontrados, ou seja, convergiram para os valores originais tracejados em vermelho, sendo o parâmetro $\hat{f}$ mais custoso a convergir. Caso seja necessário aumentar a velocidade, basta alterar a matriz P_0.

O algoritmo implementado em MATLAB para a execução do método pode ser visto na sequência.

```
clc
clear
close all

index_fig = 0;

%% Experiencia 4

% Sistema da questao 1

% Parametros do problema
M = 100;
f= 0.15;
k= 7;
```

```

lambda= 1; % Parametro do filtro

t = linspace(0,25,1000); % Vetor de tempo
u = 1+cos(pi*(t/3)); % Sinal de entrada

filtro = tf(1 , [1 2*lambda lambda^2]); % Implementacao do filtro
z= lsim(filtro, u, t); % Sinal z do modelo parametrico

lambda= 1; % Parametro do filtro

G= tf(1, [M f k]); % Resposta original da planta
x= lsim(G, u, t);

s= tf("s");
filtro = tf(1 , [1 2*lambda lambda^2]); % Declaracao do filtro

% Aplicacoes de filtro para o vetor phi
xdd_f= lsim(s^2*filtro, x,t);
xd_f= lsim(s*filtro, x, t);
x_f= lsim(filtro, x, t);

%Vetor phi com filtro
phi= [xdd_f xd_f x_f]';

%% Experiencia 4 - MMQ

% Matriz inicial P
P0 = 10000*[200 0 0;
      0 1 0;
      0 0 100];

% Matriz inicial theta
theta_0 = [1;
           0;
           1;];

% Passo de simulacao
dt = t(2)-t(1);

% Progressao de theta
theta_v = [theta_0];
P_m= [];
```

```
% Algoritmo MMQ
for i=1:length(t)
    phi_i = phi(:,i); % Valor de phi pro instante i

    ms = 1+phi_i'*phi_i; % Valor de ms para o instante i

    P_d = -P0*phi_i*phi_i'*P0/(ms^2); % Calculo da derivada de P
    P = P0 + P_d*dt; % Atualizacao de P

    z_i = z(i);  % Valor de z para o instante i
    z_hat = theta_0'*phi_i; % Estimativa de z com os parametros atuais

    e = (z_i - z_hat)/ms^2; % Calculo de erro

    theta_d = P*e*phi_i;  % Calculo da derivada de theta
    theta = theta_0 + theta_d*dt; % Atualizacao de theta

    theta_v(:, end+1) = theta; % Armazenamento da progressao de theta_v
    P_sin= svds(P);
    P_m(:, end+1)= P_sin;

    % Valores para a proxima iteracao
    P0 = P;
    theta_0 = theta;
end

%% Experimento 4 - Plots

% Plot M

linha_referencia = 100*ones(1,length(t)); % Valor de referencia para M

index_fig = index_fig + 1;
figure(index_fig)
grid on
hold on
plot(t, theta_v(1,1:length(t)))
plot(t, linha_referencia, '--')
ylim([0,110])
title("Experiencia 4 - Parametro M");
hLeg = legend ("$\hat{M}$", 'M');
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]');
```

```
ylabel("Parametro");

%Plot f

linha_referencia = 0.15*ones(1,length(t)); % Valor de referencia para f

index_fig = index_fig + 1;
figure(index_fig)
grid on
hold on
plot(t, theta_v(2,1:length(t)))
plot(t, linha_referencia, '--')
title("Experiencia 4 - Parametro f");
hLeg = legend ("$\hat{f}$", 'f');
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]');
ylabel("Parametro");

%Plot k

linha_referencia = 7*ones(1,length(t)); % Valor de referencia para k

index_fig = index_fig + 1;
figure(index_fig)
grid on
hold on
plot(t, theta_v(3,1:length(t)))
plot(t, linha_referencia, '--')
title("Experiencia 4 - Parametro k");
hLeg = legend ("$\hat{k}$", 'k');
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]');
ylabel("Parametro");
```

5. Experiência 5 – Identificação de Parâmetros

Seja o sistema de controle de velocidade, $\omega(t)$, por tensão de armadura, $V_a(t)$, de um motor CC representado pela função de transferência a seguir:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_1}{s^2 + K_2 s + K_3} \tag{23}$$

tal que $K_1 = \frac{K_m}{JL_a}$, $K_2 = \frac{(JR_a + bL_a)}{JL_a}$ e $K_3 = \frac{K_b K_m}{JL_a}$, sendo J a inércia do rotor, b o coeficiente de atrito viscoso do rotor, Ra a resistência de armadura, La a indutância de armadura e $Km = Kb$ a constante do motor. Considere os dados do motor CC: $J = 0.001Nms^2/rad$, $b = 0.268Nms/rad$, $Ra = 1.36\Omega$, $La = 3.6mH$ e $Km = 0.838Nm/A$ e a tensão de armadura de entrada $V_a(t) = sen\left(\frac{\pi}{3}t\right)$.

(a) Determine um modelo paramétrico admitindo K_1, K_2 e K_3 parâmetros constantes desconhecidos, sendo $\omega(t)$ e $V_a(t)$ disponíveis a cada instante t.

(b) Projete um algoritmo de identificação de parâmetros baseado no método dos mínimos quadrados para estimar as constantes K_m, K_1, K_2 e K_3, sendo $\omega(t)$ e $V_a(t)$ disponíveis a cada instante t. Gere os sinais de $\hat{K}_1,\ \hat{K}_2\ e\ \hat{K}_3$.

5.1. Resolução

5.1.1. a)

A representação do modelo paramétrico será desenvolvida utilizando o mesmo raciocínio dos exercícios anteriores. Dessa forma, inicialmente teremos:

$$\frac{\omega(s)}{V_a(s)} = \frac{K_1}{s^2 + K_2 s + K_3} \tag{24}$$

Isolando os termos conhecidos, teremos:

$$K_1 V_a = s^2\omega + sK_2\omega + K_3\omega \tag{25}$$

Por fim, isolando V_a, chegamos na seguinte equação:

$$V_a = \frac{s^2}{K_1}\omega + \frac{sK_2}{K_1}\omega + \frac{K_3}{K_1}\omega \tag{26}$$

Por fim, escrevendo a equação da forma $z = \theta^{*T}\phi$ e adicionando o filtro de segunda ordem $\frac{1}{(s+\lambda)^2}$, teremos:

$$\frac{1}{(s+\lambda)^2}V_a = \begin{pmatrix} \frac{1}{K_1} & \frac{K_2}{K_1} & \frac{K_3}{K_1} \end{pmatrix} \cdot \begin{pmatrix} \frac{s^2}{(s+\lambda)^2}\,\omega \\ \frac{s}{(s+\lambda)^2}\,\omega \\ \frac{1}{(s+\lambda)^2}\,\omega \end{pmatrix} \tag{27}$$

No script abaixo consta a definição dos parâmetros iniciais e a implementação do modelo paramétrico no MATLAB.

```
% Experimento 5 - modelo parametrico

% Parametros do problema
Ra= 1.36;
La= 0.0036;
Km= 0.838;
Kb=Km;
J= 0.001;
b= 0.268;
t= linspace(0,10,10000);

lambda= 1;

% Constantes simplificadas do problema
K1= Km/(J*La);
K2= (J*Ra + b*La)/(J*La);
K3= (Kb*Km)/(J*La);

% Vetor de tempo
t = linspace(0,25,1000);
% Sinal de entrada (Nesse caso, um sinal rico)
Va= sin(pi*t/3)+sin(pi*t/2)+sin(5*pi*t/3);%+sin(100*pi*t/3)+sin(0.1*pi*t/3)+sin(400*pi*t/3)+sin(38*pi*t/3)+sin(0.001*pi*t/3)+sin(42*pi*t/3);

% Funcao de transferencia
G_5= tf(K1, [1 K2 K3]);
% Obtencao do sinal de entrada w
w= lsim(G_5, Va, t);

% Declaracao do filtro
s= tf("s");
filtro= 1/(s+lambda)^2;

% Sinais do modelo parametrico
z= lsim(filtro,Va,t);
```

```

phi_1= lsim(filtro*s^2,w,t);
phi_2= lsim(filtro*s,w,t);
phi_3= lsim(filtro,w,t);

phi= [phi_1 phi_2 phi_3]';
```

5.1.2. b)

Logo após a parametrização, foi montado um algoritmo para estimação dos parâmetros K_m, K_1, K_2 e K_3, o método escolhido foi o dos Mínimos Quadrados Recursivo puro.

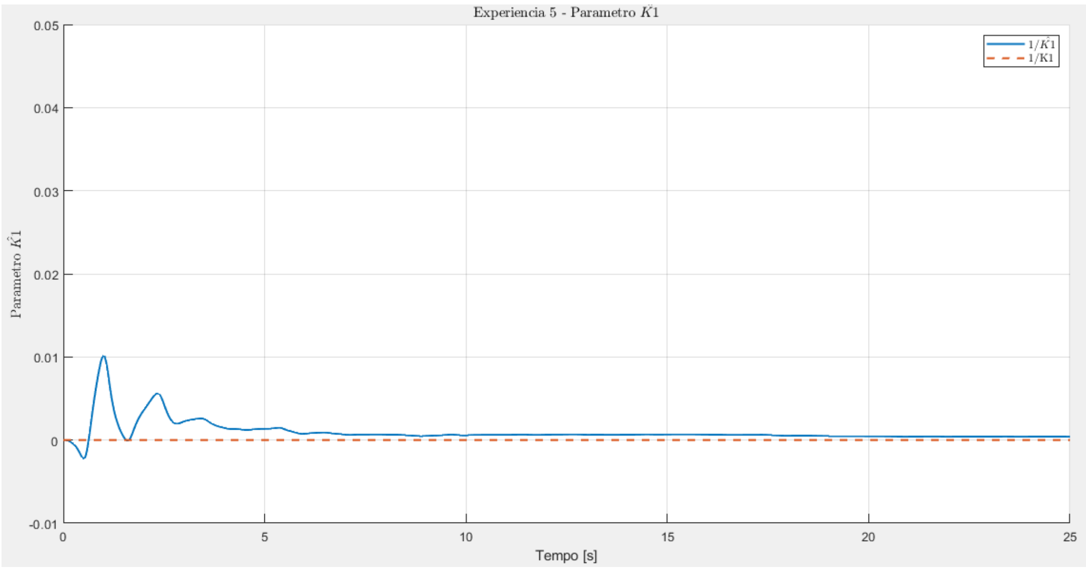
O raciocínio utilizado foi semelhante ao apresentado no exercício 4 e novamente foram utilizadas as Equações 19, 20, 21 e 22 com o objetivo de definir $\epsilon, m_s, \dot{P}$ e $\dot{\theta}$, que são necessários para a implementação.

Os valores iniciais de P_0 e θ_0 que foram escolhidos arbitrariamente para o experimento, considerando o desempenho final obtido, são mostrados a seguir:

$$P_0 = 10 \cdot \begin{pmatrix} 100 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix} \quad ; \quad \theta_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

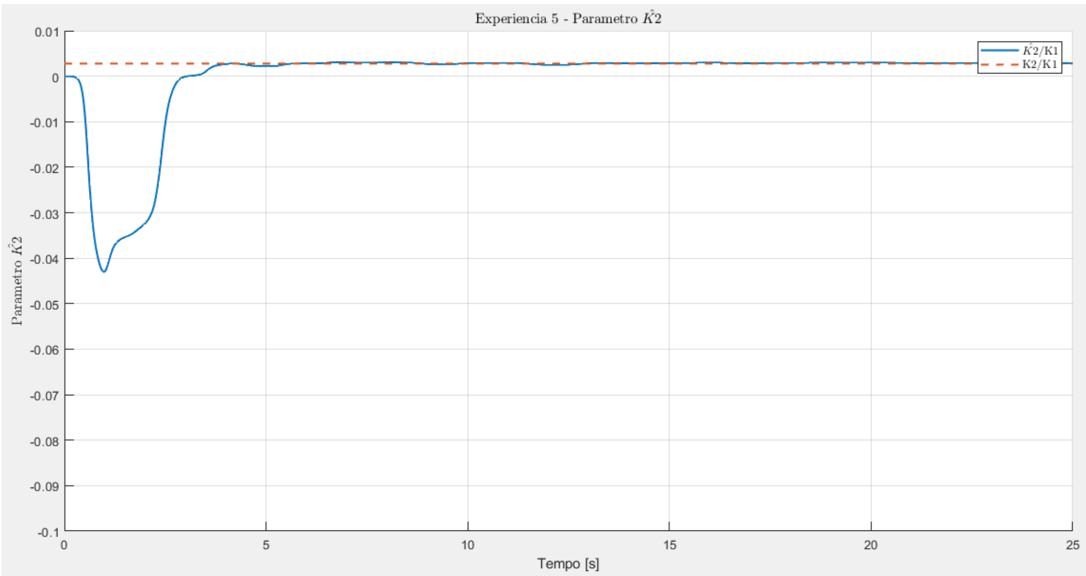
Portanto, é possível obter os gráficos dos parâmetros estimados $\hat{K}_1$, $\hat{K}_2$ e $\hat{K}_3$ que são mostrados nas Figuras 15, 16 e 17, respectivamente.

Figura 15: Estimativa do parâmetro $\hat{K}_1$



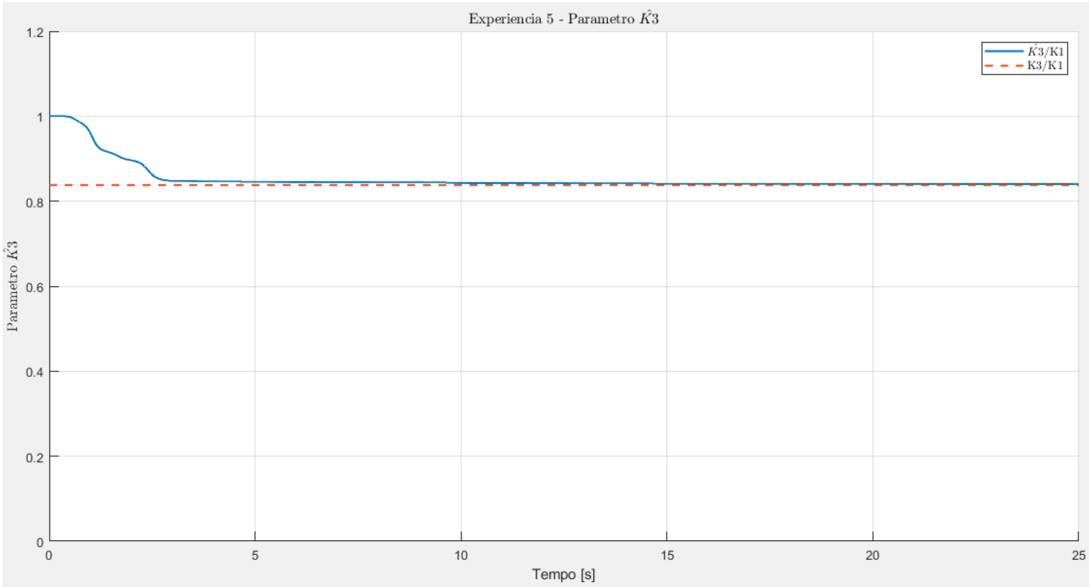
Fonte: Autoria própria

Figura 16: Estimativa do parâmetro $\hat{K}_2$



Fonte: Autoria própria

Figura 17: Estimativa do parâmetro $\hat{K}_3$



Fonte: Autoria própria

No script abaixo é possível observar a implementação do algoritmo utilizado no MATLAB e a plotagem dos sinais de $\hat{K}_1$, $\hat{K}_2$ e $\hat{K}_3$ estimados.

```

% Experimento 5 - MMQ

% Matriz inicial P
P0= 10*[100 0 0; 0 10 0; 0 0 10];

% Matriz inicial theta
theta_0= [0;0;1];

dt = t(2)-t(1);
theta_m= [];

for i=1:length(t)
    phi_i = phi(:,i); % Valor de phi pro instante i

    ms = 1+phi_i'*phi_i; % Valor de ms para o instante i

    P_d = -P0*phi_i*phi_i'*P0/(ms^2); % Calculo da derivada de P
    P = P0 + P_d*dt; % Atualizacao de P

    z_i = z(i);  % Valor de z para o instante i
    z_hat = theta_0'*phi_i; % Estimativa de z com os parametros atuais
```

```

    e = (z_i - z_hat)/(ms^2); % Calculo de erro

    theta_d = P*e*phi_i;  % Calculo da derivada de theta
    theta = theta_0 + theta_d*dt; % Atualizacao de theta

    theta_m(:, end+1) = theta; % Armazenamento da progressao de theta_v

    % Valores para a proxima iteracao
    P0 = P;
    theta_0 = theta;
end

%% Experimento 5 - plots

one= ones(1,length(t));
linha_referencia1 = one*(1/K1);
linha_referencia2 = one*(K2/K1);
linha_referencia3 = one*(K3/K1);

index_fig = index_fig + 1;
figure(index_fig)
grid on
hold on
plot(t, theta_m(1,:)', 'LineWidth', 1.5)
plot(t, linha_referencia1,'--','LineWidth', 1.5)
htitle = title("Experiencia 5 - Parametro $\hat{K1}$");
set(htitle,'Interpreter','latex');
hLeg = legend ("1/$\hat{K1}$", "1/K1");
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]');
hylabel = ylabel("Parametro $\hat{K1}$");
set(hylabel,'Interpreter','latex');
ylim([-0.01,0.05]);

index_fig = index_fig + 1;
figure(index_fig)
grid on
hold on
plot(t, theta_m(2,:)', 'LineWidth', 1.5)
plot(t, linha_referencia2,'--','LineWidth', 1.5)
htitle = title("Experiencia 5 - Parametro $\hat{K2}$");
set(htitle,'Interpreter','latex');
```

```
hLeg = legend ("$\hat{K2}$/K1", "K2/K1");
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]');
hylabel = ylabel("Parametro $\hat{K2}$");
set(hylabel,'Interpreter','latex');
ylim([-0.1,0.01]);

index_fig = index_fig + 1;
figure(index_fig)
grid on
hold on
plot(t, theta_m(3,:)', 'LineWidth', 1.5)
plot(t, linha_referencia3,'--','LineWidth', 1.5)
htitle = title("Experiencia 5 - Parametro $\hat{K3}$");
set(htitle,'Interpreter','latex');
hLeg = legend ("$\hat{K3}$/K1", "K3/K1");
set(hLeg,'Interpreter','latex');
xlabel ('Tempo [s]');
hylabel = ylabel("Parametro $\hat{K3}$");
set(hylabel,'Interpreter','latex');
ylim([0,1.2]);
```

6. Experiência 6 – Identificação de Parâmetros

(a) Os valores dos parâmetros estimados alcançaram os valores esperados? Comente.

(b) Por que o método aplicado é nomeado "Método dos Mínimos Quadrados"?

(c) Enumere os passos para implementação do "Método dos Mínimos Quadrados"no procedimento de identificação paramétrica.

(d) Quais características o sinal de entrada do sistema deve apresentar para obtermos um bom desempenho do sistema de identificação?

(e) Qual variável determina a velocidade do processo de identificação? Como ela atua?

6.1. Resolução

6.1.1. a)

Não. Isso ocorre pois os parâmetros são estimados com base na dinâmica do erro do sistema, o que garante apenas que esses parâmetros serão limitados. Existem infinitas soluções que fazem com que o modelo da planta siga o modelo de referência, logo não há garantia de que a solução encontrada seja a solução em que os parâmetros estimados sigam os valores ideais.

6.1.2. b)

O método aplicado recebe esse nome pois atua de uma forma a minimizar uma função custo que envolve o erro quadrático da diferença entre os valores observados e os valores estimados. Isso resulta em uma previsão adaptativa que pode reduzir ou quase zerar o erro do sistema.

6.1.3. c)

A implementação do método dos mínimos quadrados pode ser feita seguindo os seguintes passos:

1. Gerar os sinais z e ϕ do sistema a ser analisado.

2. Definir os valores iniciais P_0 e θ_0;

3. Fazer os passos 4 a 10 em *loop* até finalizar o procedimento;

4. Estimar a saída com os parâmetros atuais;
5. Calcular o fator de correção Ms;
6. Estimar a variação da matriz de adaptação P;
7. Calcular o próximo P em um intervalo de tempo dt;
8. Calcular o erro de estimativa;
9. Estimar a variação dos parâmetros θ;
10. Calcular o próximo θ em um intervalo de tempo dt.

Este procedimento foi detalhado no pseudocódigo do Algoritmo 1 deste documento.

6.1.4. d)

Para possibilitar um bom desempenho no sistema de identificação, o sinal de entrada deve ser um sinal "rico", ou seja, deve conter uma faixa de frequências ampla o suficiente para excitar toda a dinâmica do sistema, de forma permanente, sob diferentes condições ao longo do tempo

6.1.5. e)

A variável que determina a velocidade de processo é matriz P. Isso ocorre, pois ela atua como fator de aprendizagem no processo de atualização dos parâmetros, multiplicando o erro de estimação dos parâmetros. Quanto maior os valores de $P(0)$, mais rápido os parâmetros irão convergir para os valores esperados.

7. Experiência 7(a) – Controle Adaptativo

Considere o sistema de primeira ordem:

$$\dot{x} = ax + bu,\ x(0) = x_0. \tag{28}$$

onde a e b são constantes desconhecidas com $b > 0$. Projete e analise um sistema de Controle Adaptativo em Modelo de Referência (MRAC) direto que possa estabilizar o sistema e regular x tendendo a zero para qualquer condição inicial x_0. Simule seu sistema para diferentes condições iniciais e sinais de entrada de referência, e para os seguintes parâmetros: $a = -1,2$ e $b = 5$.

Para implementação no MATLAB considere $a_m = 10$; $b_m = 1$; $r = 5cos(t) + 2sen(\pi t)$. Gere os sinais de $x(t)$, $x_m(t)$, $e(t)$, $k(t)$ e $l(t)$. Confira se $k(t)$ e $l(t)$ alcançaram k^* e l^* respectivamente. Comente.

7.1. Resolução:

Para o desenvolvimento da sistema de controle será utilizado o modelo de referência apresentado na Equação 29:

$$\dot{x_m} = -a_m x_m + b_m r \tag{29}$$

A lei de controle ideal, adotada para que o modelo MRAC atinja o objetivo de controle, é dada pela Equação 30.

$$u^* = -k^* x + l^* r \tag{30}$$

onde k^* e l^* são os valores ideias de ganhos do controlador.

Para obter os ganhos ideias em função dos parâmetros a e b da planta apresentado na Equação 28, é necessário igualar o sistema com o modelo de referência. Substituindo a Equação 30 na planta, obtém-se a Equação 31.

$$\begin{aligned} \dot{x} &= ax + b(-k^* x + l^* r) \\ \dot{x} &= (a - k^*)x + bl^* r \end{aligned} \tag{31}$$

Para que a lei de controle acompanhe o modelo é necessário que $\dot{x} = \dot{x_m}$. Comparando as Equações 29 e 31, obtém-se as relações apresentadas na Equação 32.

$$\begin{cases} a_m = a + bk^* \\ b_m = bl^* \end{cases}$$

$$\begin{cases} k^* = \dfrac{a_m - a}{b} \\ l^* = \dfrac{b_m}{b} \end{cases} \tag{32}$$

Supondo que não seja possível obter os ganhos reais da planta, se faz necessário propor uma estratégia para estimar esses ganhos. Para isso, é proposta a lei de controle apresentada na Equação 33.

$$u = -k(t)x + l(t)r \tag{33}$$

onde $k(t)$ e $l(t)$, são aproximações de k^* e l^*, respectivamente.

Antes de substituir a lei de controle no modelo apresentado na Equação 28, é necessário realizar uma manipulação matemática somando e subtraindo o termo bu^* na equação, conforme apresentado a seguir:

$$\dot{x} = ax + bu - bk^*x + bl^*r + bk^*x - bl^*r$$

Substituindo então as Equações 32 e 33, obtém-se:

$$\dot{x} = ax + bu - bk^*x + bl^*r + bk^*x - bl^*r$$

$$\dot{x} = -a_m x + b_m r + b(k^*x - l^*r - k(t)x + l(t)r) \tag{34}$$

Com a equação de $\dot{x}$ baseada nos parâmetros ótimos e estimados, é possível encontrar a expressão para o erro do sistema, descrita da seguinte forma:

$$e = x - x_m$$

$$\dot{e} = \dot{x} - \dot{x_m}$$

$$\dot{e} = (-a_m x + b_m r + b(k^*x - l^*r - k(t)x + l(t)r)) - (-a_m x_m + b_m r)$$

$$\dot{e} = a_m(x_m - x) + b(k^*x - l^*r - k(t)x + l(t)r)) \tag{35}$$

Que pode ser reescrito como:

$$\dot{e} = a_m e + b(-\tilde{k}x + \tilde{l}r) \tag{36}$$

onde $\tilde{k}$ e $\tilde{l}$ são os erros de estimativa dos parâmetros dados por:

$$\begin{cases} \tilde{k} = k(t) - k^* \\ \tilde{l} = l(t) - l^* \end{cases} \tag{37}$$

Para gerar as estimativas dos parâmetros propostos é eleita a função de Lyapunov candidata da Equação 38.

$$V(e, \tilde{k}, \tilde{l}) = \frac{e^2}{2} + \frac{\tilde{k}^2}{2\gamma_1}|b| + \frac{\tilde{l}^2}{2\gamma_2}|b| \tag{38}$$

onde γ_1, $\gamma_2 > 0$

Derivando a função se tem:

$$\dot{V}(e, \tilde{k}, \tilde{l}) = -a_m e^2 - b\tilde{k}ex + b\tilde{l}er + \frac{|b|\tilde{k}}{\gamma_1}\dot{\tilde{k}} + \frac{|b|\tilde{l}}{\gamma_2}\dot{\tilde{l}} \tag{39}$$

Para garantir a estabilidade do sistema é necessário que $\dot{V}(e, \tilde{k}, \tilde{l}) <= 0$. O único termo que garante isso pelas condições conhecidas, é $-a_m e^2$. Logo, com o intuito de zerar o restante dos termos da função, são obtidas as seguintes expressões:

$$\begin{cases} b\tilde{k}ex = \frac{|b|\tilde{k}}{\gamma_1}\dot{\tilde{k}} \\ b\tilde{l}er = \frac{|b|\tilde{l}}{\gamma_l}\dot{\tilde{l}} \end{cases} \tag{40}$$

Como k^* e l^* são constantes, então $\dot{\tilde{k}} = \dot{k}$ e $\dot{\tilde{l}} = \dot{l}$. Logo, as leis de adaptações podem ser reescritas conforme apresentado nas Equações 41:

$$\begin{cases} \dot{k} = \gamma_1 exsing(b) \\ \dot{l} = -\gamma_2 ersing(b) \end{cases} \tag{41}$$

Por fim, para a implementação do controle adaptativo, foi desenvolvido o script apresentado a seguir. Os valores de γ_1, γ_2, k_0 e l_0 foram definidos de forma arbitrária.

```
%% Experiencia 7 - (a)

% Parametros da planta
a = -1;
b = 5;
% Parametros do modelo de referencia
am = 10;
bm = 1;
```

```

% Valores ideais de controle
k_star = (a+am)/b;
l_star = bm/b;

% Vetor de tempo
dt = 0.01;
t = 0:dt:10;

% Sinal de entrada
r = 5*cos(t) + 2*sin(pi*t);

% Valores de x para o controlador ideal
x0 = 0;
x = zeros(1,length(t));
x(1) = x0;

% Funcao de transferencia modelo de referencia
s = tf('s');
G = bm/(s+am);
xm = lsim(G, r, t);

% Valores iniciais de adaptacao
k0 = 1;
l0 = 1;
gamma1 = 50;
gamma2 = 0.5;

% Valores de x para o controlador adaptativo
x_hat = zeros(1,length(t));
x0_hat = 0;
x_hat(1) = x0_hat;

% Vetores para progressao dos ganhos e do erro
k = zeros(1,length(t));
k(1) = k0;
l = zeros(1,length(t));
l(1) = l0;
e = zeros(1,length(t));
e(1) = x_hat(1)-xm(1);

for i=2:length(t)
    % X com controlador adaptativo
```

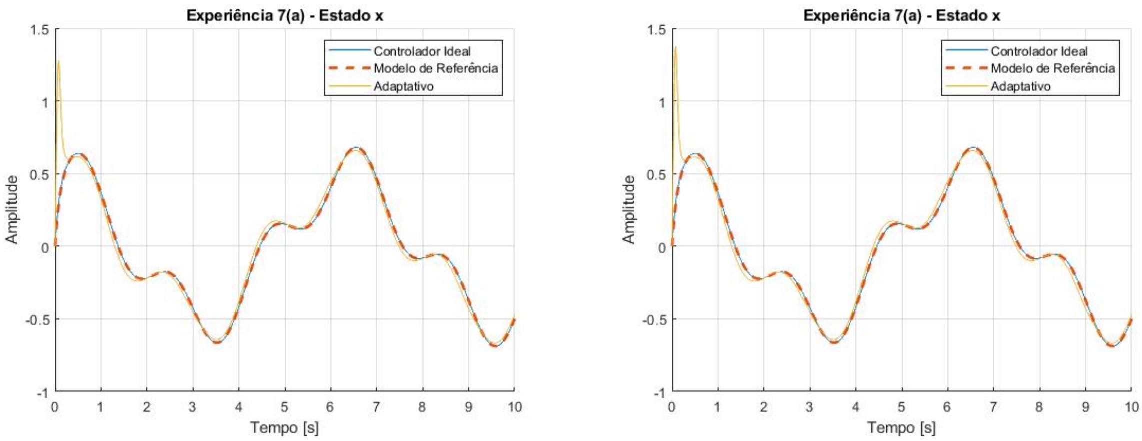
```
    e(i-1) = x_hat(i-1) - xm(i-1);

    k_dot = gamma1*e(i-1)*x_hat(i-1)*sign(b);
    k(i) = k_dot*dt + k(i-1);

    l_dot = -gamma2*e(i-1)*r(i-1)*sign(b);
    l(i) = l_dot*dt + l(i-1);

    u_hat = -k(i)*x_hat(i-1) + l(i)*r(i-1);
    x_hat_dot = a*x_hat(i-1) + b*u_hat;
    x_hat(i) = x_hat_dot * dt + x_hat(i-1);

    % X como controlador ideal
    u = -k_star*x(i-1) + l_star*r(i-1);
    x_dot = a*x(i-1) + b*u;
    x_i = x(i-1) + x_dot*dt;
    x(i) = x_i;
end

figure(1)
hold on
grid on
plot(t, x, "LineWidth", 1)
plot(t, xm, '--', "LineWidth", 2)
plot(t, x_hat, "LineWidth", 1)
legend("Controlador Ideal", "Modelo de Referencia", "Adaptativo")
xlabel("Tempo [s]")
ylabel("Amplitude")
title("Experiencia 7(a) - Estado x")


figure(2)
hold on
grid on
plot(t,e)
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(a) - Erro de adaptacao")

k_star_vector = k_star*ones(1, length(t));
figure(3)
hold on
grid on
```

```
plot(t,k)
plot(t, k_star_vector, '--')
legend("Adaptativo", "Ideal")
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(a) - Parametro de controle k")
ylim([0, 9])

l_star_vector = l_star*ones(1, length(t));
figure(4)
hold on
grid on
plot(t,l)
plot(t, l_star_vector, '--')
legend("Adaptativo", "Ideal")
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(a) - Parametro de controle l")
ylim([0, 1.5])
```

O sistema foi implementado considerando dois valores de a, -1 e 2. Para cada valor de a foram gerados os gráficos do Estado x, do erro e e dos parâmetros $k(t)$ e $l(t)$ em comparação com seus valores ideias.

Figura 18: Estado x em comparação com x_m

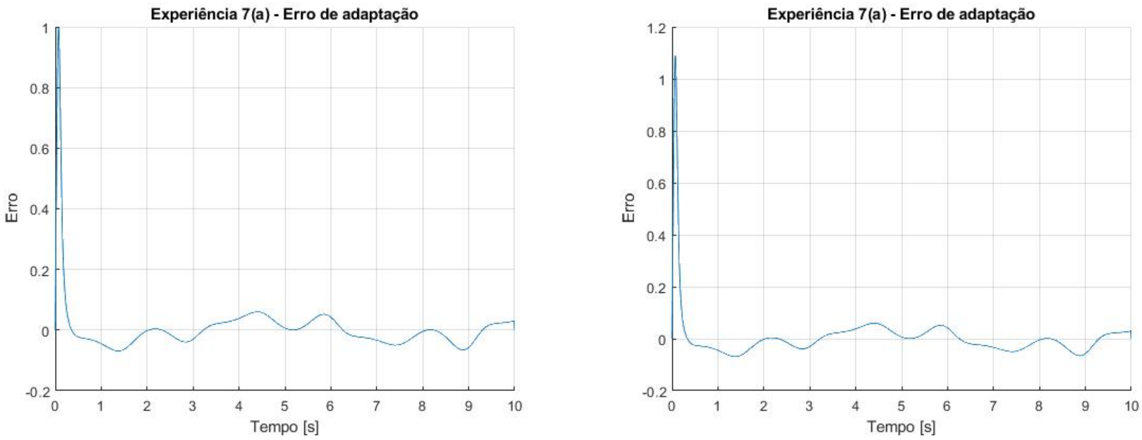
(a) Estado x para $a = -1$ (b) Estado x para $a = 2$



Fonte: Autoria própria.

Figura 19: Erro do sistema

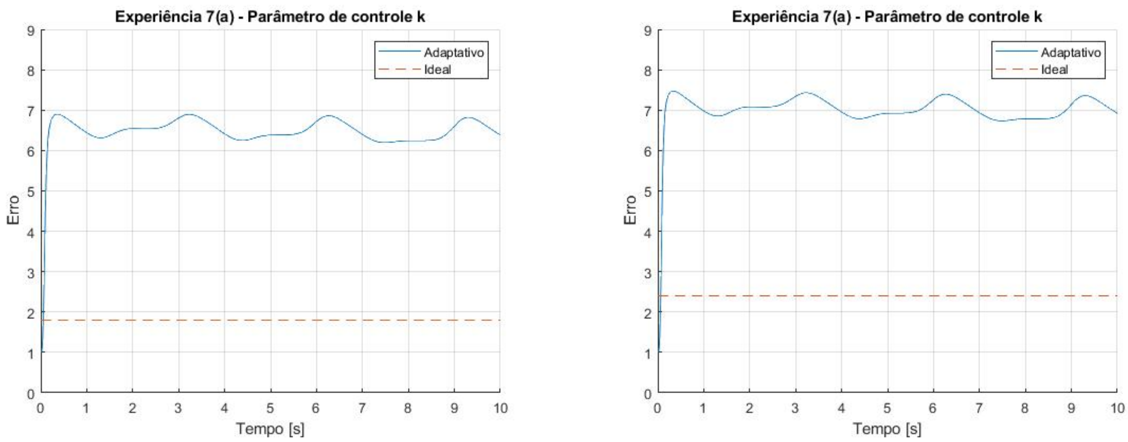
(a) Erro para $a = -1$ (b) Erro para $a = 2$



Fonte: Autoria própria.

Figura 20: Estimativa do parâmetro $k(t)$ em comparação do com k^*

(a) Parâmetro $k(t)$ para $a = -1$ (b) Parâmetro $k(t)$ para $a = 2$



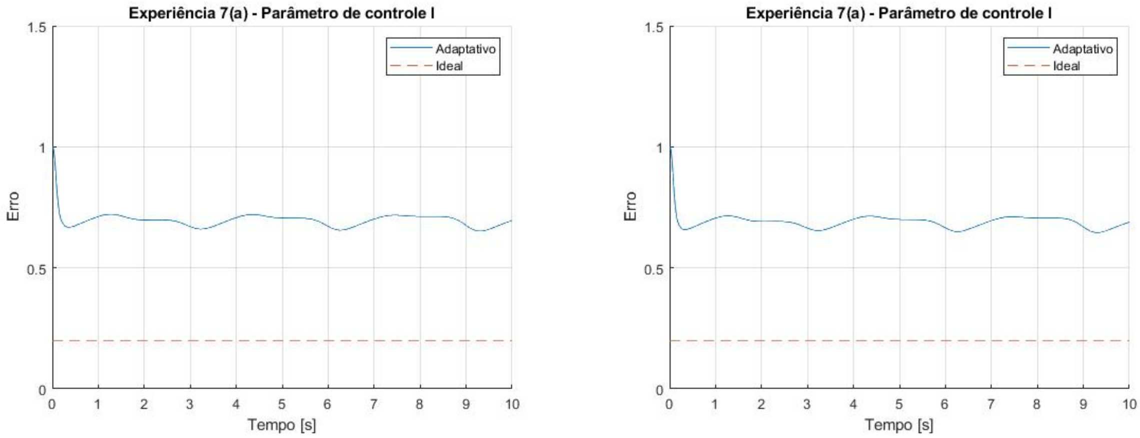
Fonte: Autoria própria.

É possível notar que o único parâmetro que apresentou mudança significativamente foi o $k(t)$, que conforme apresentado na Equação 32, é diretamente pendente do valor de a. No entanto, mesmo para esse parâmetro, a variação de erro se manteve. Isso ocorre, pois como o controle é adaptativo, mesmo mudando a planta ele vai tentar convergir os valores ideais para alcançar o modelo de referência.

O sinais de x e e, assim como o parâmetro l, como não são diretamente dependentes do valor de a, sofrem pouca, ou nenhuma, influência dessa mudança.

Figura 21: Estimativa do parâmetro $l(t)$ em comparação do com l^*

(a) Parâmetro $l(t)$ para $a = -1$ (b) Parâmetro $l(t)$ para $a = 2$



Fonte: Autoria própria.

Também é possível notar que, para nenhum dos valores utilizados o sistema obteve um erro zero no intervalo de tempo simulado. Além disso, mesmo com o controle chegando próximo ao modelo de referência, conforme visto na Figura 18, os valores de $k(t)$ e $l(t)$, não atingiram os valores ideais de ganhos k^* e l^*.

8. Experiência 7(b) – Controle Adaptativo

Considere o sistema de primeira ordem:

$$y_p = \frac{b}{s-1}u_p \tag{42}$$

onde $b > 0$ é o único parâmetro desconhecido. Projete e analise um sistema de Controle Adaptativo em Modelo de Referência (MRAC) direto que possa estabilizar o sistema e forçar y_p a seguir a saída y_m do modelo de referência.

$$y_m = \frac{2}{s+2}r \tag{43}$$

para qualquer sinal de referência r que seja limitado e contínuo. Simule seu sistema para diferentes condições iniciais e para o valor $r = 5$, $sin(2t)$, $\frac{1}{1+t}$ e $b = 12$.

Gere os sinais de $x(t)$, $x_m(t)$, $e(t)$, $k(t)$ e $l(t)$ para cada caso. Confira se $k(t)$ e $l(t)$ alcançaram $k*$ e $l*$ respectivamente, em todos os casos. Comente.

8.1. Resolução:

Inicialmente é definida uma lei de controle ideal do sistema, apresentada na Equação 46, que é aplicada ao sistema para igualar o mesmo ao modelo de referência.

$$u = -k^* y_p + l^* r \tag{44}$$

onde k^* e l^* são dados pelas expressões a seguir:

$$\begin{cases} k^* = \dfrac{a_m - a}{b} \\ l^* = \dfrac{b_m}{b} \end{cases} \tag{45}$$

No entanto, os ganhos do controlador, dados pela Equação 45, são dependentes do parâmetro b, que é desconhecido. Portanto se faz necessário o uso de uma estimativa dos parâmetros, dada pela lei de controle da Equação 46.

$$u_p = -k(t)y_p + l(t)r \tag{46}$$

Para encontrar $\dot{y_p}$ em função dos parâmetros ideias é necessário fazer uma manipulação matemática somando e subtraindo bu na expressão, conforme apresentado a seguir:

$$y_p = \frac{b}{s+a}\,u$$

$$\dot{y_p} = -ay_p + bu_p$$

$$\dot{y_p} = -ay_p + bu_p + bu - bu$$

$$\dot{y_p} = -ay_p + bu_p - bk^*y + bl^*r + bk^*y_p - bl^*r$$

$$\dot{y_p} = -ay_p + bu_p - a_my_p + ay_p + b_mr + bk^*y_p - bl^*r$$

$$\dot{y_p} = -a_my_p + b_mr + b(k^*y_p - bl^*r + u_p) \quad (47)$$

Com isso, é possível definir o erro de acompanhamento do sistema como $e = y_p - y_m$ e, consequentemente, $\dot{e} = \dot{y_p} = \dot{y_m}$. Assim a dinâmica do erro é definida por:

$$\dot{e} = (-a_my_p + b_mr_p + b(k^*y_p - l^*r + u_p)) - (-a_my_m + b_mr)$$

$$\dot{e} = -a_m(y_p - y_m) + b(k^*y_p - bl^*r + u_p) \quad (48)$$

Substituindo u_p e considerando $\tilde{k} = k(t) - k^*$ e $\tilde{l} = l(t) - l^*$, obtém-se a seguinte expressão para o erro:

$$-a_m(y_p - y_m) + b(k^*y_p - bl^*r - k(t)y_p + l(t)r)$$

$$\dot{e} = -a_me + b(-\tilde{k}y + \tilde{l}) \quad (49)$$

De forma análoga ao exercício anterior é proposta uma função de Lyapunov candidata, afim de projetar a lei de adaptação do sistema. A função proposta é apresentada na Equação 50:

$$V(e, \tilde{k}, \tilde{l}) = \frac{e^2}{2} + \frac{\tilde{k}^2}{2\gamma_1}|b| + \frac{\tilde{l}^2}{2\gamma_2}|b| \quad (50)$$

onde γ_1, $\gamma_2 > 0$

Derivando a função se tem a Equação 51:

$$\dot{V}(e, \tilde{k}, \tilde{l}) = -a_me^2 - b\tilde{k}ex + b\tilde{l}er + \frac{|b|\tilde{k}}{\gamma_1}\dot{\tilde{k}} + \frac{|b|\tilde{l}}{\gamma_2}\dot{\tilde{l}} \quad (51)$$

Assim como no exercício anterior, garantir a estabilidade do sistema é necessário que $\dot{V}(e, \tilde{k}, \tilde{l}) <= 0$. Como o valor de a_m é conhecido e positivo e e^2 é sempre positivo, isso garante que o termo $-a_me^2$, será sempre negativo.

Para os demais termos da função, não se tem informações suficientes para garantir o sinal. Logo, a lei de adaptação pode ser obtida zerando esses termos da função. Assim, são obtidas as seguintes relações:

$$\begin{cases} b\tilde{k}ex = \dfrac{|b|\tilde{k}}{\gamma_1}\dot{\tilde{k}} \\ b\tilde{l}er = \dfrac{|b|\tilde{l}}{\gamma_l}\dot{\tilde{l}} \end{cases} \tag{52}$$

Como k^* e l^* são constantes, então $\dot{\tilde{k}} = \dot{k}$ e $\dot{\tilde{l}} = \dot{l}$. Além disso, b pode ser escrito como $|b| \cdot sign(b)$, cancelando o módulo de b. Logo, as leis de adaptações podem ser reescritas conforme apresentado nas Equações 53:

$$\begin{cases} \dot{k} = \gamma_1 exsing(b) \\ \dot{l} = -\gamma_2 ersing(b) \end{cases} \tag{53}$$

Para a implementação desses sistema de controle, foi realizado o script apresentado a seguir. Nele o sistema é simulado considerando três entradas diferentes.

```
%% Experiencia 7 - (b)

index_fig = 0;

a = -1;
b = 12;
am = 2;
bm = 2;

% Valores ideais de controle
k_star = (a+am)/b;
l_star = bm/b;

% Intervalo de tempo
dt = 0.01;
t = 0:dt:10;

% Valores de referencia
r1 = 5*ones(1,length(t));
r2 = sin(2*t);
r3 = 1./(1+t);

r = [r1; r2; r3];

% Funcao de transferencia modelo de referencia
s = tf('s');
G = bm/(s+am);
```

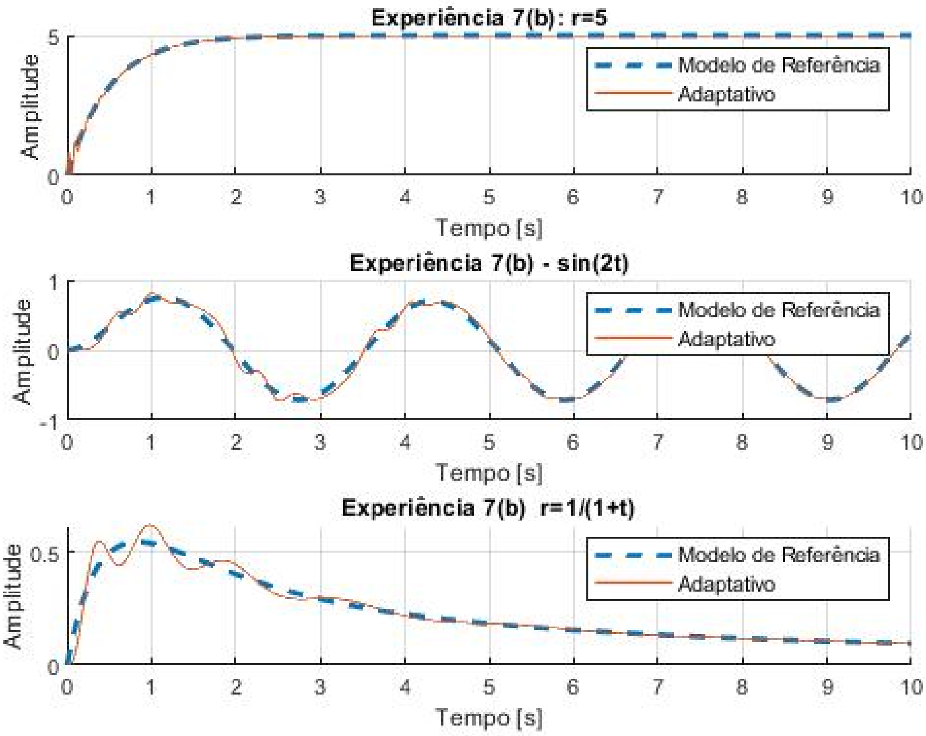
```
xm_1 = lsim(G, r1, t);
xm_2 = lsim(G, r2, t);
xm_3 = lsim(G, r3, t);

xm = [xm_1 xm_2 xm_3]';

% Valores iniciais de adaptacao
k0 = 1;
l0 = 1;
gamma1 = 1;
gamma2 = 20;

% Valores de x para o controlador adaptativo
x_hat = zeros(3,length(t));
x0_hat = 0;
x_hat(1) = x0_hat;

% Vetores para progressao dos ganhos e do erro
k = zeros(3,length(t));
k(1) = k0;
l = zeros(3,length(t));
l(1) = l0;
e = zeros(3,length(t));
e(1) = x_hat(1)-xm(1);

for j=[1,2,3]
    for i=2:length(t)
        % X com controlador adaptativo
        e(j,i-1) = x_hat(j,i-1) - xm(j,i-1);

        k_dot = gamma1*e(j,i-1)*x_hat(j,i-1)*sign(b);
        k(j,i) = k_dot*dt + k(j,i-1);

        l_dot = -gamma2*e(j,i-1)*r(j,i-1)*sign(b);
        l(j,i) = l_dot*dt + l(j,i-1);

        u_hat = -k(j,i)*x_hat(j,i-1) + l(j,i)*r(j,i-1);
        x_hat_dot = a*x_hat(j,i-1) + b*u_hat;
        x_hat(j,i) = x_hat_dot * dt + x_hat(j,i-1);
    end
end

```

```
% Plot de x

index_fig = index_fig + 1;
figure(index_fig)
subplot(3,1,1)
hold on
grid on
plot(t, xm(1,:), '--', "LineWidth", 2)
plot(t, x_hat(1,:), "LineWidth", 1)
legend("Modelo de Referencia", "Adaptativo")
xlabel("Tempo [s]")
ylabel("Amplitude")
title("Experiencia 7(b): r=5")
subplot(3,1,2)
hold on
grid on
plot(t, xm(2,:), '--', "LineWidth", 2)
plot(t, x_hat(2,:), "LineWidth", 1)
legend("Modelo de Referencia", "Adaptativo")
xlabel("Tempo [s]")
ylabel("Amplitude")
title("Experiencia 7(b) - sin(2t)")
subplot(3,1,3)
hold on
grid on
plot(t, xm(3,:), '--', "LineWidth", 2)
plot(t, x_hat(3,:), "LineWidth", 1)
legend("Modelo de Referencia", "Adaptativo")
xlabel("Tempo [s]")
ylabel("Amplitude")
title("Experiencia 7(b)  r=1/(1+t)")

% Plot de e

index_fig = index_fig + 1;
figure(index_fig)
subplot(3,1,1)
hold on
grid on
plot(t,e(1,:))
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(b): r=5")
```

```
subplot(3,1,2)
hold on
grid on
plot(t,e(2,:))
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(b) - sin(2t)")
subplot(3,1,3)
hold on
grid on
plot(t,e(3,:))
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(b)  r=1/(1+t)")

k_star_vector = k_star*ones(1, length(t));

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,k)
plot(t, k_star_vector, '--')
legend("r=5", "r=sin(2t)", "r=1/(1+t)", "Ideal")
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(b) - Parametro de controle k")

l_star_vector = l_star*ones(1, length(t));

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,l)
plot(t, l_star_vector, '--')
legend("r=5", "r=sin(2t)", "r=1/(1+t)", "Ideal")
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(b) - Parametro de controle L")
```

As Figuras 22, 23, 24 e 25, apresentam os sinais y_p e y_m, o erro e do sistema e os parâmetros $k(t)$ e $l(t)$, respectivamente.

Figura 22: Estado y_p em comparação com y_m para várias entradas

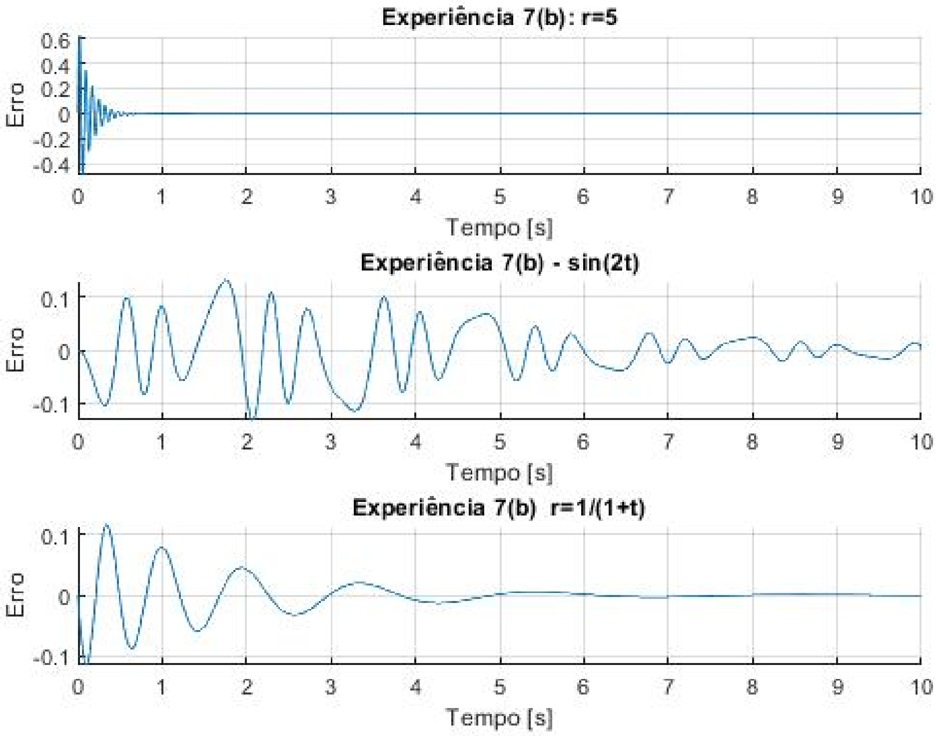


Fonte: Autoria própria

Em relação aos estados, é possível notar que para todas as entradas o sistema segue o modelo de referência. No entanto, analisando o gráfico de erro, nota-se que para a segunda entrada o sistema não chega em erro nulo como acontece nas outras duas.

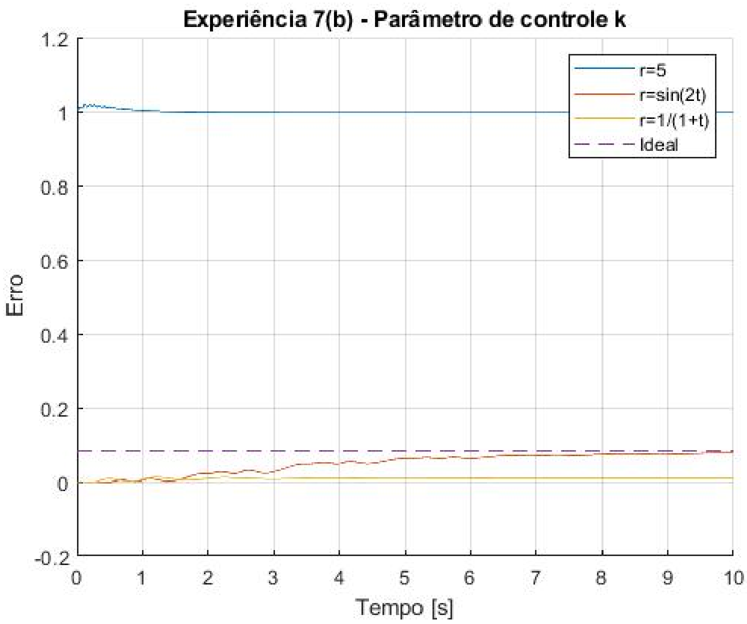
Agora em relação aos ganhos $k(t)$ e $l(t)$, é possível notar que a entrada que obteve o sinal de erro mais estável, foi justamente a que obteve o maior erro de estimativa dos ganhos. As demais entradas chegam próximas dos valores ideais de ganho do controlador.

Figura 23: Erro do sistema para várias entradas



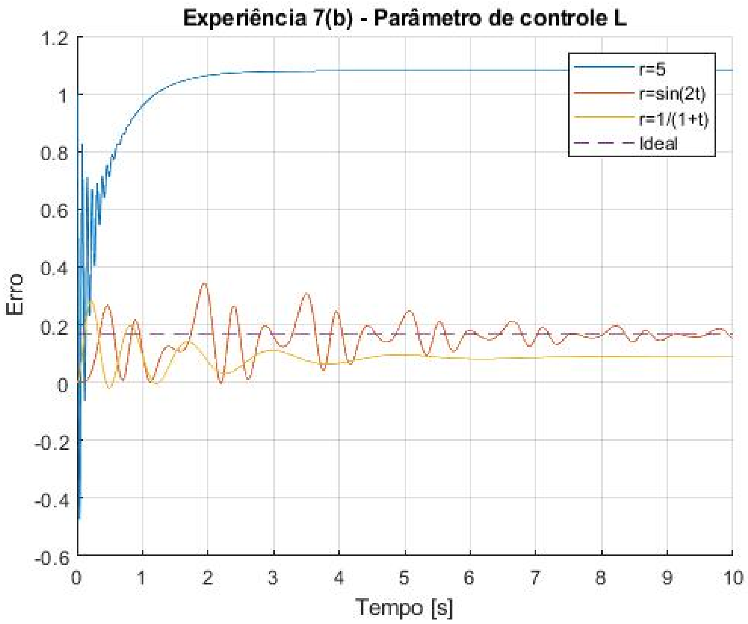
Fonte: Autoria própria

Figura 24: Estimativa do parâmetro $k(t)$ em comparação do com k^*



Fonte: Autoria própria

Figura 25: Estimativa do parâmetro $l(t)$ em comparação do com l^*



Fonte: Autoria própria

9. Experiência 7(c) – Controle Adaptativo

A dinâmica do acelerador para o subsistema de velocidade de um veículo pode ser representada por um sistema de primeira ordem da Equação 54.

$$V = \frac{b}{s+a}\theta + d \tag{54}$$

onde V é a velocidade do veículo, θ é o ângulo do acelerador e d é o distúrbio constante de carregamento. Os parâmetros $b > 0$ e a são constantes desconhecidas cujos valores dependem do estado operacional do veículo, definido pelo estado da engrenagem, pela velocidade em estado estacionário, pelo arrasto, etc. Projete um sistema CC escolhendo o ângulo do acelerador θ de modo que V seguirá a velocidade desejada V_m gerada pelo modelo de referência da Equação 55.

$$V_m = \frac{0.5}{s+0.5}V_s \tag{55}$$

onde V_s é a velocidade desejada definida pelo motorista.

(a) Assuma que a, b e d são exatamente conhecidos. Projete uma lei MRC que atinja o objetivo de controle.

(b) Projete e analise um sistema MRAC direto para ser usado no caso de a, b e d serem desconhecidos (com $b > 0$).

(c) Projete um esquema MRAC indireto para ser usado no caso de a, b e d serem desconhecidos (com $b > 0$).

(d) Simule e compare seus sistemas em (b) e (c), assumindo $V_s = 55$ e usando os seguintes valores para a, b e d:

(i) $a = 0.5$, $b = 1.5$ e $d = 10$;

(ii) $a = 0.5 + \dfrac{0.04}{1+V}$, $b = 1.5$ e $d = 0.2 + \sin(0.02t)$;

Escolha $\theta = -k^*V + l^*V_s + \delta^*$. Gere os sinais de $V(t)$, $V_m(t)$, $e(t)$, $k(t)$, $l(t)$ e $\delta(t)$ para cada caso. Confira se $k(t)$, $l(t)$ e $\delta(t)$ alcançaram k^*, l^* e δ^*, respectivamente, em todos os casos.

(e) (Exercício proposto adicional) Use o controlador indireto MRAC considerando a lei de controle $\theta = -kV + lV_s$, sem a parcela adicional de controle. Analise os resultados.

9.1. Resolução

9.1.1. a)

A lei de controle para que o modelo MRC atinja o objetivo de controle pode ser formalizada pela Equação 56:

$$\theta^* = -k^* V + l^* V_s + \delta^* \tag{56}$$

onde k^*, l^* e δ^* são os valores ideais para os ganhos do controlador.

O objetivo do projeto do modelo MRC é encontrar os valores ideais da lei de controle em função dos parâmetros da planta a, b e d. Para isso, é necessário escrever a Equação 56 de modo que seja comparável com o modelo de referência. Substituindo então a Equação 54 nela, obtém-se como resultado a Equação 57.

$$\begin{gathered} V = \frac{b}{s+a}(-k^* V + l^* V_s + \delta^*) + d \\ V(s + a + bk^*) = V_s(bl^*) + [\delta^* b + d(s+a)] \end{gathered} \tag{57}$$

Reescrevendo o modelo de referência da Equação 55 no mesmo formato acima, substituindo as constantes para valores genéricos a_m e b_m, a Equação 58 é obtida.

$$V_m = \frac{b_m}{s + a_m} V_s$$

$$V_m(s + a_m) = V_s b_m \tag{58}$$

Para que a planta seja capaz de acompanhar o modelo de referência, ou seja, em que $V_m = V$, os valores ideias de ganhos do controlador são obtidos pela comparação entre a Equação 57 com a 58. Os valores obtidos por este método são vistos na Equação 59

$$\begin{gathered} \begin{cases} a_m = a + bk^* \\ b_m = bl^* \\ 0 = \delta^* b + d(s+a) \end{cases} \\ \begin{cases} k^* = \dfrac{a_m - a}{b} \\ l^* = \dfrac{b_m}{b} \\ \delta^* = -\dfrac{da}{b} \end{cases} \end{gathered} \tag{59}$$

Observação: O termo ds que aparece no sistema é considerado 0 pois, nesse caso, d é tratado como uma constante.

9.1.2. b)

Iniciando o projeto de um MRAC direto, é necessário voltar em 54 para somar e subtrair $b\theta^*$.

$$V(s+a) = b\theta + d(s+a)$$

$$\dot{V} + Va = b\theta + da + b\theta^* - b\theta^*$$

$$\dot{V} = -V(a+bk^*) + bl^*V_s + da + b\delta^* + b(k^*V - l^*V_s - \delta^* + \theta) \tag{60}$$

Algumas modificações podem ser feitas na Equação 60, sabendo que $\delta^* = -\dfrac{da}{b}$, $a_m = a + bk^*$ e $b_m = bl^*$, é obtida a Equação 61

$$\dot{V} = -a_m V + b_m V_s + b(k^*V - l^*V_s - \delta^* + \theta) \tag{61}$$

Tendo a equação de $\dot{V}$ com base nos parâmetros ótimos e do modelo de referência, é possível encontrar a dinâmica de erro do sistema:

$$e = v - V_m$$

$$\dot{e} = \dot{V} - \dot{V}_m$$

$$\dot{e} = -a_m e + b(k^*V - l^*V_s - \delta^* + \theta) \tag{62}$$

Substituindo em 62 a relação $\theta = -k(t)V + l(t)V_s + \delta(t)$ como a lei de controle adaptativa do processo, obtém-se:

$$\dot{e} = -a_m e + b(\tilde{k}V - \tilde{l}V_s - \tilde{\delta}) \tag{63}$$

onde $\tilde{k} = k(t) - k^*$, $\tilde{l} = l(t) - l^*$ e $\tilde{\delta} = \delta(t) - \delta^*$.

Para mostrar que o erro é estável para esse sistema adaptativo, é necessário propor a função de Liapunov candidata da Equação 64.

$$V_L(e, \tilde{k}, \tilde{l}, \tilde{\delta}) = \frac{e^2}{2} + \frac{\tilde{k}^2}{2\gamma_1}|b| + \frac{\tilde{l}^2}{2\gamma_2}|b| + \frac{\tilde{\delta}^2}{2\gamma_3}|b| \tag{64}$$

Derivando V_L a partir do método do gradiente de 65, obtém-se a Equação 66:

$$\dot{V}_L(e,\tilde{k},\tilde{l},\tilde{\delta}) = \frac{\partial V_L}{\partial e}\dot{e} + \frac{\partial V_L}{\partial \tilde{k}}\dot{\tilde{k}} + \frac{\partial V_L}{\partial \tilde{l}}\dot{\tilde{l}} + \frac{\partial V_L}{\partial \tilde{\delta}}\dot{\tilde{\delta}} \tag{65}$$

$$\dot{V}_L(e,\tilde{k},\tilde{l},\tilde{\delta}) = -a_m e^2 + be(-\tilde{k}V + \tilde{l}V_s + \tilde{\delta}) + \frac{\tilde{k}}{\gamma_1}|b|\dot{\tilde{k}} + \frac{\tilde{l}}{\gamma_2}|b|\dot{\tilde{l}} + \frac{\tilde{\delta}}{\gamma_3}|b|\dot{\tilde{\delta}} \tag{66}$$

Sabendo que $a_m e^2 < 0$, para garantir a estabilidade do sistema basta que as outras parcelas do polinômio se cancelem. Para isso, resolve-se o sistema da Equação 67:

$$\begin{cases} be\tilde{k}V = \dfrac{\tilde{k}|b|\dot{\tilde{k}}}{\gamma_1} \\ -be\tilde{l}V_s = \dfrac{\tilde{l}|b|\dot{\tilde{l}}}{\gamma_2} \\ -be\tilde{\delta} = \dfrac{\tilde{\delta}|b|\dot{\tilde{\delta}}}{\gamma_3} \end{cases} \tag{67}$$

Para que as relações de 67 sejam verdadeiras, precisamos definir a dinâmica dos parâmetros adaptativos. Como k^*, l^* e δ^* são constantes, então $\dot{\tilde{k}} = \dot{k(t)}$, $\dot{\tilde{l}} = \dot{l(t)}$ e $\dot{\tilde{\delta}} = \dot{\delta(t)}$. Portanto, a dinâmica dos parâmetros são dadas pelas Equações 68, 69 e 70, respectivamente.

$$\dot{k(t)} = e\gamma_1 V \text{sgn}(b) \tag{68}$$

$$\dot{l(t)} = e\gamma_2 V_s \text{sgn}(b) \tag{69}$$

$$\dot{\delta(t)} = e\gamma_3 \text{sgn}(b) \tag{70}$$

9.1.3. c)

Para projetar as leis de adaptação para o MRAC Indireto, é necessário voltar novamente na Equação 54. A partir de algumas manipulações é obtido o modelo de $\dot{V}$ somando e subtraindo o termo $a_m V$ e definindo $da = h$ na Equação 71

$$V = \frac{b}{s+a}\theta + d$$

$$\dot{V} = b\theta + da - aV + a_m V - a_m V$$

$$\dot{V} = a_m V - (a + a_m)V + b\theta + h \tag{71}$$

Dessa forma, a dinâmica de alteração da variável de tensão estimada $\hat{V}$ com os parâmetros adaptados $\hat{a}$, $\hat{b}$ e $\hat{h}$, baseado no modelo obtido anteriormente é mostrado na Equação 72.

$$\dot{\hat{V}} = a_m\hat{V} - (\hat{a} + a_m)V + \hat{b}\theta + \hat{h} \tag{72}$$

Como o erro de referência é definido por $e := V - \hat{V}$, a dinâmica de erro é dada pela Equação 73

$$\dot{e} = -a_m e + \tilde{a}V - \tilde{b}\theta - \tilde{h} \tag{73}$$

sendo $\tilde{a} = a - \hat{a}$, $\tilde{b} = b - \hat{b}$ e $\tilde{h} = h - \hat{h}$.

Para mostrar que o erro é estável para esse sistema adaptativo, é necessário propor a função de Liapunov candidata da Equação 74.

$$V_L(e, \tilde{a}, \tilde{b}, \tilde{h}) = \frac{e^2}{2} + \frac{\tilde{a}^2}{2\gamma_1} + \frac{\tilde{b}^2}{2\gamma_2} + \frac{\tilde{d}^2}{2\gamma_3} \tag{74}$$

Derivando V_L a partir do método do gradiente de 75, obtém-se a Equação 76

$$\dot{V}_L(e, \tilde{a}, \tilde{b}, \tilde{h}) = \frac{\partial V_L}{\partial e}\dot{e} + \frac{\partial V_L}{\partial \tilde{a}}\dot{\tilde{a}} + \frac{\partial V_L}{\partial \tilde{b}}\dot{\tilde{b}} + \frac{\partial V_L}{\partial \tilde{h}}\dot{\tilde{h}} \tag{75}$$

$$\dot{V}_L(e, \tilde{k}, \tilde{l}, \tilde{\delta}) = -a_m e^2 + e(\tilde{a}V - \tilde{b}\theta - \tilde{h}) + \frac{\tilde{a}}{\gamma_1}\dot{\tilde{a}} + \frac{\tilde{b}}{\gamma_2}\dot{\tilde{b}} + \frac{\tilde{h}}{\gamma_3}\dot{\tilde{h}} \tag{76}$$

Sabendo que $a_m e^2 < 0$, para garantir a estabilidade do sistema basta que as outras parcelas do polinômio se cancelem. Para isso, resolve-se o sistema da Equação 77

$$\begin{cases} e\tilde{a}V = -\dfrac{\tilde{a}\dot{\tilde{a}}}{\gamma_1} \\ e\tilde{b}V_s = \dfrac{\tilde{b}\dot{\tilde{b}}}{\gamma_2} \\ e\tilde{h} = -\dfrac{\tilde{h}\dot{\tilde{h}}}{\gamma_3} \end{cases} \tag{77}$$

Para que as relações de 77 sejam verdadeiras, precisamos definir a dinâmica dos parâmetros adaptativos. Como a, b e h são constantes, então $\dot{\tilde{a}} = \dot{\hat{a}}(t)$, $\dot{\tilde{v}} = \dot{\hat{v}}(t)$ e $\dot{\tilde{h}} = \dot{\hat{h}}(t)$. Portanto, a dinâmica dos parâmetros são dadas pelas Equações 78, 79 e 80, respectivamente.

$$\dot{a} = -e\gamma_1 V \tag{78}$$

$$\dot{b} = \begin{cases} e\gamma_2 V_s, \text{ para } |b| > b_{min} \\ 0, \text{ caso contrário} \end{cases} \tag{79}$$

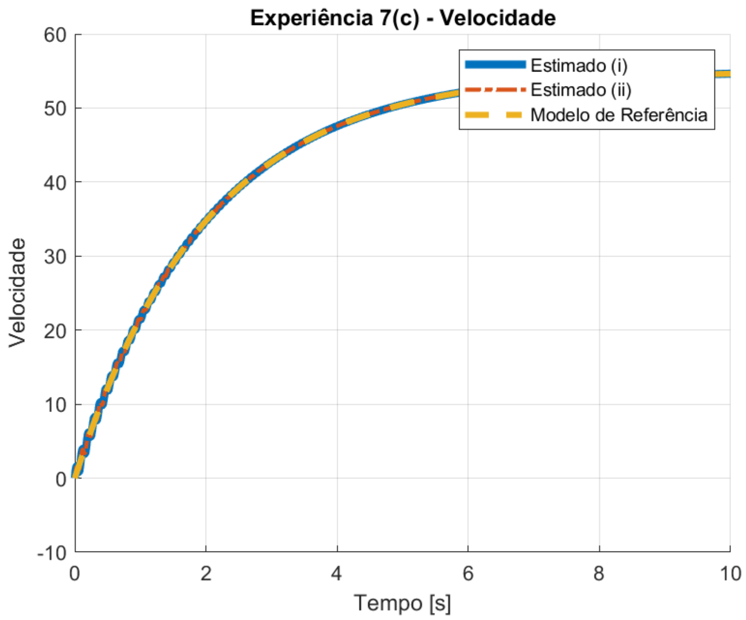
$$\dot{h} = -e\gamma_3 \tag{80}$$

9.1.4. d)

❖ **MRAC Direto**

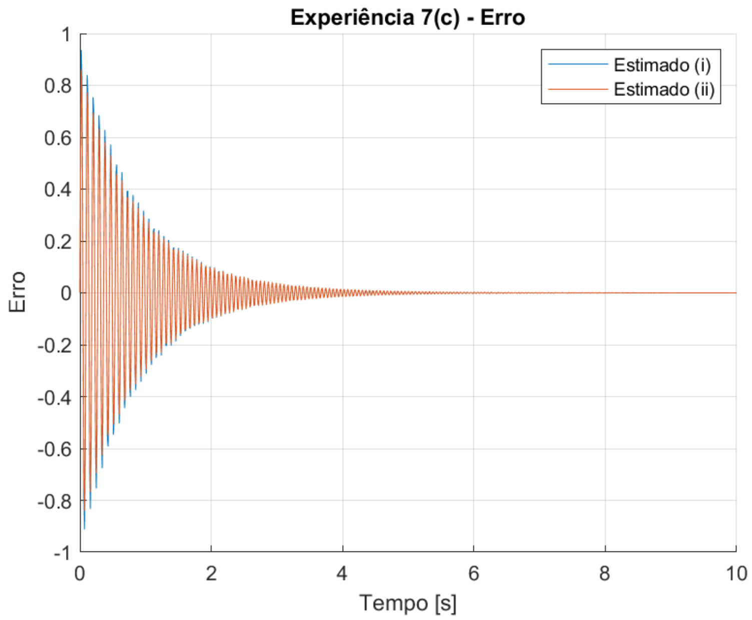
Primeiro, os gráficos para o MRAC direto solicitados para os dois modelos (i) e (ii) podem ser vistos nos gráficos de 26 a 30.

Figura 26: Saída de velocidade para (i) e (ii)



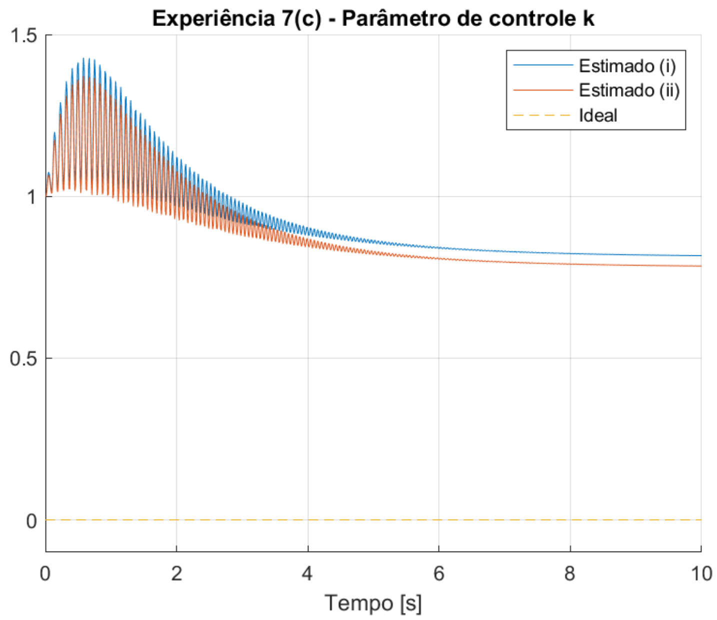
Fonte: Autoria própria

Figura 27: Sinal de erro para (i) e (ii)



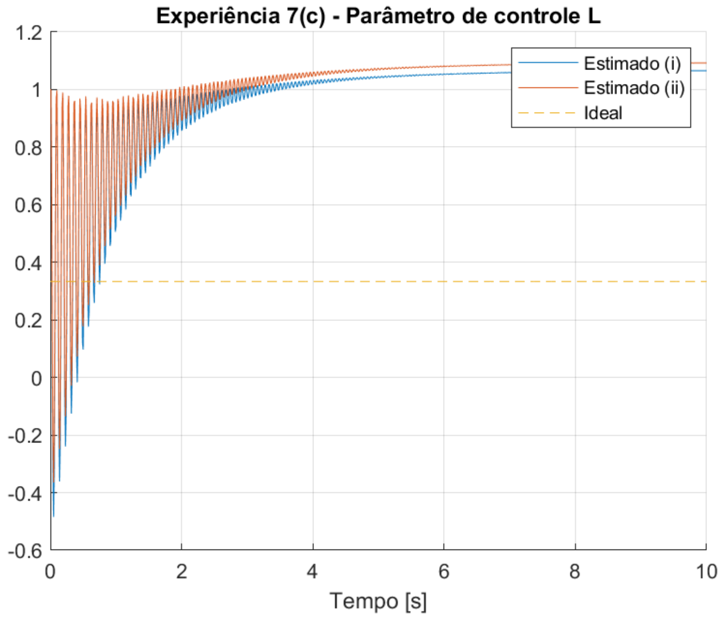
Fonte: Autoria própria

Figura 28: Adaptação de $\hat{k}$ para (i) e (ii)



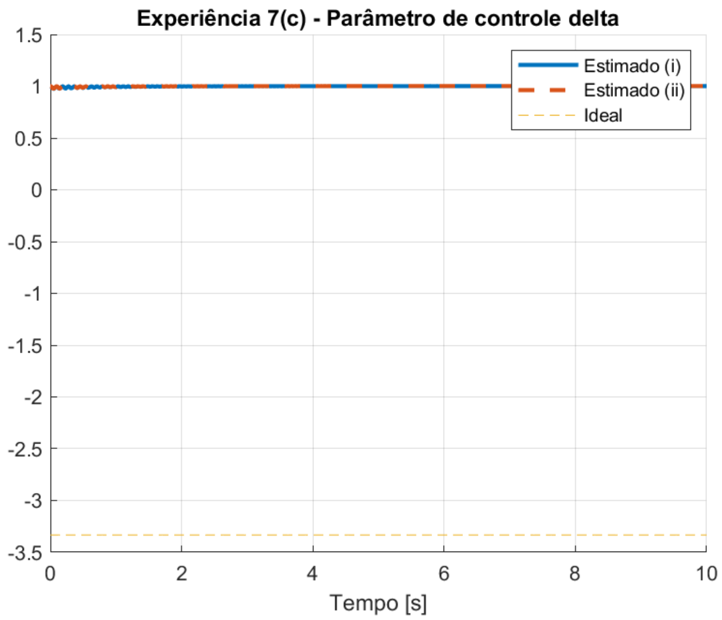
Fonte: Autoria própria

Figura 29: Adaptação de $\hat{l}$ para (i) e (ii)



Fonte: Autoria própria

Figura 30: Adaptação de $\hat{\delta}$ para (i) e (ii)



Fonte: Autoria própria

É possível notar que as Figuras 26 e 27 mostram que independente do tipo do sistema ser o (i) ou o (ii), o modelo adaptativo não possui alterações significativas no sinal de saída e converge o erro para 0 da mesma forma em ambos os casos.

Além disso, existem algumas diferenças maiores nos valores finais de ganhos do controlador $\hat{k}$ e $\hat{l}$ das Figuras 28 e 29, respectivamente, mostrando que para mínimos comportamentos diferentes da planta, como o caso de um pequeno fator de escala acrescido em a e uma pequena oscilação em d, o sistema de controle consegue se adaptar tranquilamente a essas perturbações. Vale ressaltar também que nenhum dos parâmetros chegou nos valores ideais, mas isso é esperado pois o modelo de referência prevê que o erro da saída convirja a 0, e não os parâmetros de controle.

O código utilizado para gerar os gráficos do MRAC Direto pode ser visto na sequência.

```
close all;
clc;
clear;

%% Experiencia 7 - (c)

index_fig = 0;

a = 0.5;
b = 1.5;
d = 10;

am = 0.5;
bm = 0.5;

% Valores ideais de controle
k_star = (am-a)/b;
l_star = bm/b;
delta_star = -d*a/b;

% Intervalo de tempo
dt = 0.01;
t = 0:dt:10;

% Valores de referencia
Vs = 55*ones(1,length(t));

% Funcao de transferencia modelo de referencia
s = tf('s');
G = bm/(s+am);

Vm = lsim(G, Vs, t);
```

```
% Valores iniciais de adaptacao
k0 = 1;
l0 = 1;
delta0 = 1;

gamma1 = 2;
gamma2 = 1;
gamma3 = 1;

% Valores de V para o controlador adaptativo
V_hat = zeros(1,length(t));
V0_hat = 0;
V_hat(1) = V0_hat;

% Vetores para progressao dos ganhos e do erro
k = zeros(1,length(t));
k(1) = k0;
l = zeros(1,length(t));
l(1) = l0;
delta = zeros(1,length(t));
delta(1) = delta0;

e = zeros(1,length(t));
e(1) = V_hat(1)-Vm(1);

for i=2:length(t)
    % X com controlador adaptativo
    e(i-1) = V_hat(i-1) - Vm(i-1);

    k_dot = gamma1*e(i-1)*V_hat(i-1)*sign(b);
    k(i) = k_dot*dt + k(i-1);

    l_dot = -gamma2*e(i-1)*Vs(i-1)*sign(b);
    l(i) = l_dot*dt + l(i-1);

    delta_dot = -gamma3*e(i-1)*sign(b);
    delta(i) = delta_dot*dt + delta(i-1);

    u_hat = -k(i)*V_hat(i-1) + l(i)*Vs(i-1) + delta(i);
    V_hat_dot = - a*V_hat(i-1) + b*u_hat + d*a;
    V_hat(i) = V_hat_dot * dt + V_hat(i-1);
end
```

```


%% Experiencia 7 - (c) - (ii)
%  RODAR SECAO ANTERIOR ANTES!

% Valores iniciais de adaptacao
k0 = 1;
l0 = 1;
delta0 = 1;

gamma1 = 2;
gamma2 = 1;
gamma3 = 1;

V0_hat = 0;

% Valores de V para o controlador adaptativo
V_hat_ii = zeros(1,length(t));
V_hat_ii(1) = V0_hat;

% Vetores para progressao dos ganhos e do erro
k_ii = zeros(1,length(t));
k_ii(1) = k0;
l_ii = zeros(1,length(t));
l_ii(1) = l0;
delta_ii = zeros(1,length(t));
delta_ii(1) = delta0;

e_ii = zeros(1,length(t));
e_ii(1) = V_hat_ii(1)-Vm(1);

for i=2:length(t)
    % X com controlador adaptativo
    e_ii(i-1) = V_hat_ii(i-1) - Vm(i-1);

    k_ii_dot = gamma1*e_ii(i-1)*V_hat_ii(i-1)*sign(b);
    k_ii(i) = k_ii_dot*dt + k_ii(i-1);

    l_ii_dot = -gamma2*e_ii(i-1)*Vs(i-1)*sign(b);
    l_ii(i) = l_ii_dot*dt + l_ii(i-1);

    delta_ii_dot = -gamma3*e_ii(i-1)*sign(b);
    delta_ii(i) = delta_ii_dot*dt + delta_ii(i-1);
```

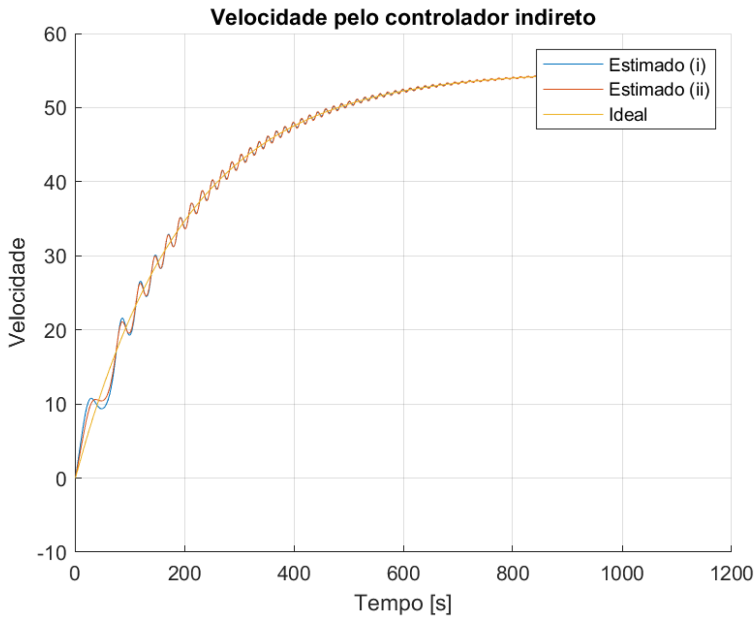
```

    u_hat = -k_ii(i)*V_hat_ii(i-1) + l_ii(i)*Vs(i-1) + delta_ii(i);
    a1 = a + 0.04/(1+V_hat_ii(i-1));
    d1 = 0.2 + sin(0.02*t(i-1));
    V_hat_dot = - a1*V_hat_ii(i-1) + b*u_hat + d1*a1;
    V_hat_ii(i) = V_hat_dot * dt + V_hat_ii(i-1);
end

%% Plots

% Plot de x

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, V_hat, "LineWidth", 2)
plot(t, Vm, '--', "LineWidth", 3)
plot(t, V_hat, '-.', "LineWidth", 2)
legend("Adaptativo (i)", "Modelo de Referencia", "Adaptativo (ii)")
xlabel("Tempo [s]")
ylabel("Velocidade")
title("Experiencia 7(c) - Velocidade")

% Plot de e

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,e)
plot(t,e_ii)
legend("Adaptativo (i)", "Adaptativo (ii)")
xlabel("Tempo [s]")
ylabel("Erro")
title("Experiencia 7(c) - erro")

% Plot de k

k_star_vector = k_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
```

```
grid on
plot(t,k)
plot(t, k_star_vector, '--')
plot(t,k_ii)
legend("Estimado (i)", "Ideal", "Estimado (ii)")
xlabel("Tempo [s]")
ylim([-0.1, 1.5])
title("Experiencia 7(c) - Parametro de controle k")

% Plot de l

l_star_vector = l_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,l)
plot(t, l_star_vector, '--')
plot(t,l_ii)
legend("Estimado (i)", "Ideal","Estimado (ii)")
xlabel("Tempo [s]")
title("Experiencia 7(c) - Parametro de controle L")

% Plot de delta

delta_star_vector = delta_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,delta)
plot(t, delta_star_vector, '--')
plot(t,delta_ii, '--')
legend("Estimado (i)", "Ideal","Estimado (ii)")
xlabel("Tempo [s]")
title("Experiencia 7(c) - Parametro de controle delta")
```

❖ MRAC indireto

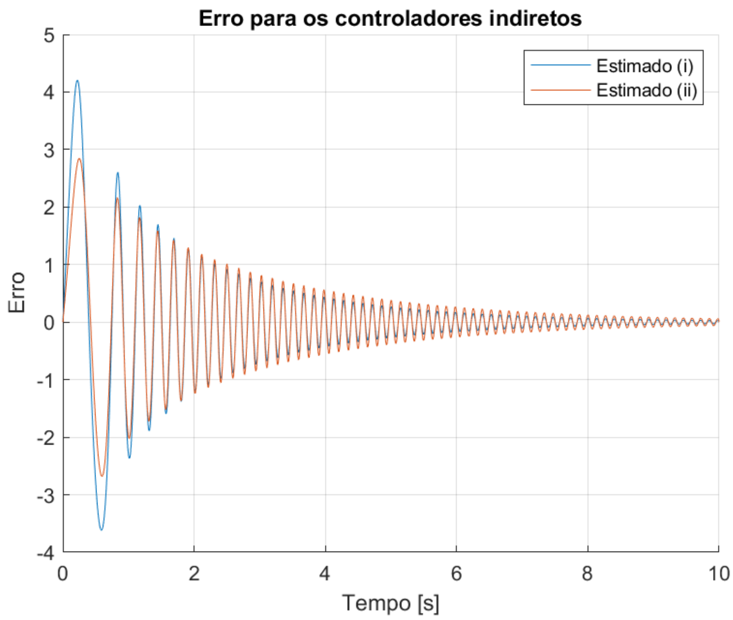
Para o controlador MRAC Indireto projetado no item (c), as respostas obtidas pela planta de velocidade e erro para os casos (i) e (ii) são vistas nas Figuras 31 e 32.

Figura 31: Saída de velocidade para o MRAC Indireto



Fonte: Autoria própria

Figura 32: Sinal de erro para o MRAC Indireto

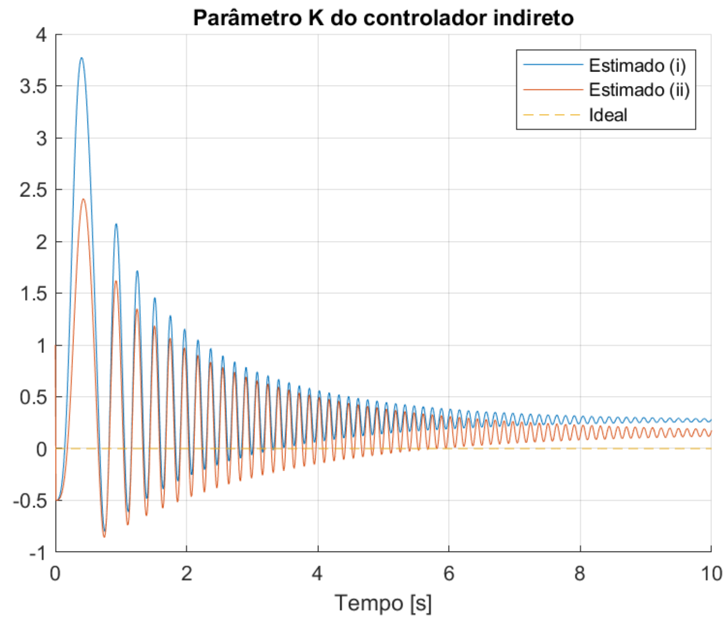


Fonte: Autoria própria

Percebe-se que a resposta do controlador indireto possui um transitório com mais oscilações em relação ao direto. Entretanto, ambos os casos de teste (i) e (ii) não apresentam diferenças significativas entre eles, acompanhando de perto o modelo de referência.

A progressão das constantes do controlador k, l e δ podem ser conferidas nos gráficos das Figuras 33, 34 e 35, respectivamente.

Figura 33: Progressão de k para (i) e (ii) no MRAC Indireto



Fonte: Autoria própria

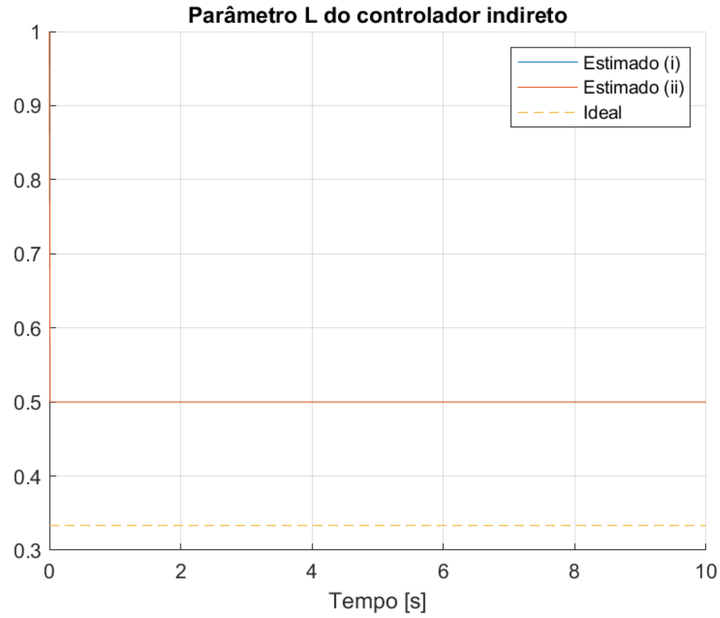
Novamente, todos os parâmetros de controle não alancaram os valores ideais k^*, l^* e δ^*, mas eles apresentaram estabilização ao longo do tempo com a adaptação dos parâmetros. Exite também uma mínima diferença no comportamento encontrado no caso (i) e (ii), estabilizando em valores diferentes de constantes e mostrando que a alteração da planta faz com que o controlador se adapte para gerar o mesmo desempenho na saída.

Com relação aos parâmetros da planta, a Figura 36 mostra a adaptação do parâmetro $\hat{b}$ para ambos os controladores, tendo em vista que o valor b_m é o mesmo para ambos os modelos.

Os valores encontrados para $\hat{b} = 1$ constantes são explicados pela limitação imposta pelo algoritmo do MRAC Indireto, que exige um valor limite definido como $b_{min} = 1$ para este experimento. Logo, durante toda a adaptação dos parâmetros, o valor de $\hat{b}$ tendeu a decrescer e não o foi permitido. Esse comportamento reflete diretamente no valor do parâmetro do controlador 34, tendo em vista que seu calculo depende do valor de $\hat{b}$.

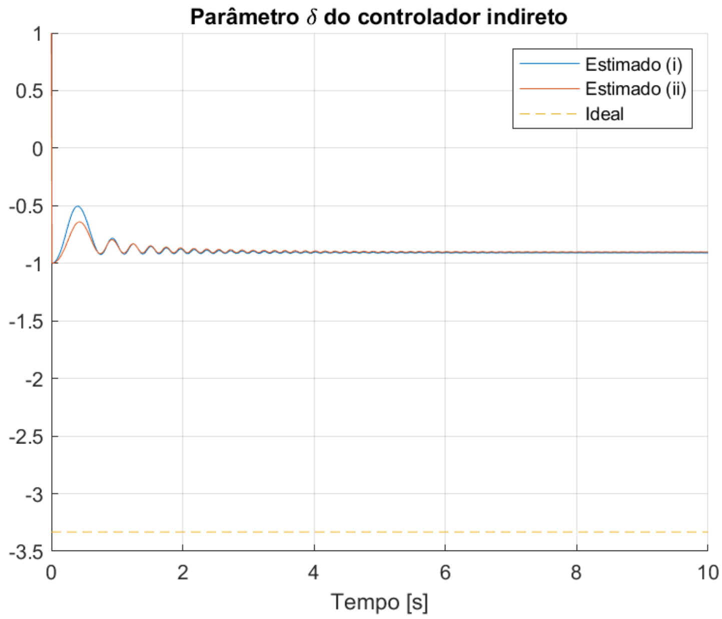
Agora, para o parâmetro $\hat{a}$ os resultados dos itens (i) e (ii) serão vistos separadamente, já que os valores esperados para ambos também são diferentes. A Figura 37 mostra para o caso (i), enquanto a Figura 38 mostra para o (ii).

Figura 34: Progressão de l para (i) e (ii) no MRAC Indireto



Fonte: Autoria própria

Figura 35: Progressão de δ para (i) e (ii) no MRAC Indireto



Fonte: Autoria própria

Figura 37: Adaptação de a para (i) no MRAC Indireto

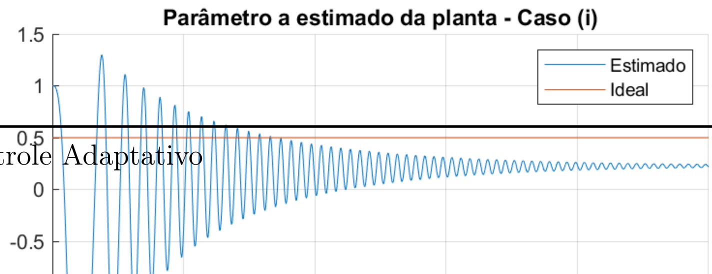
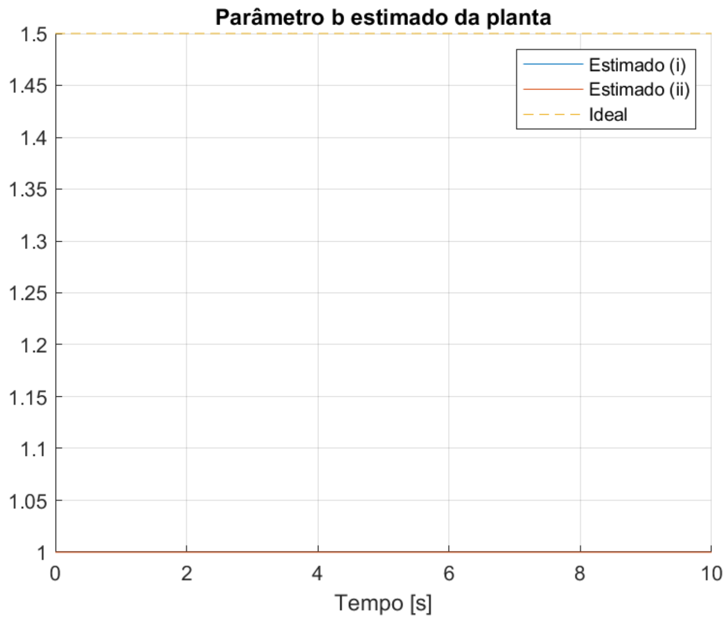
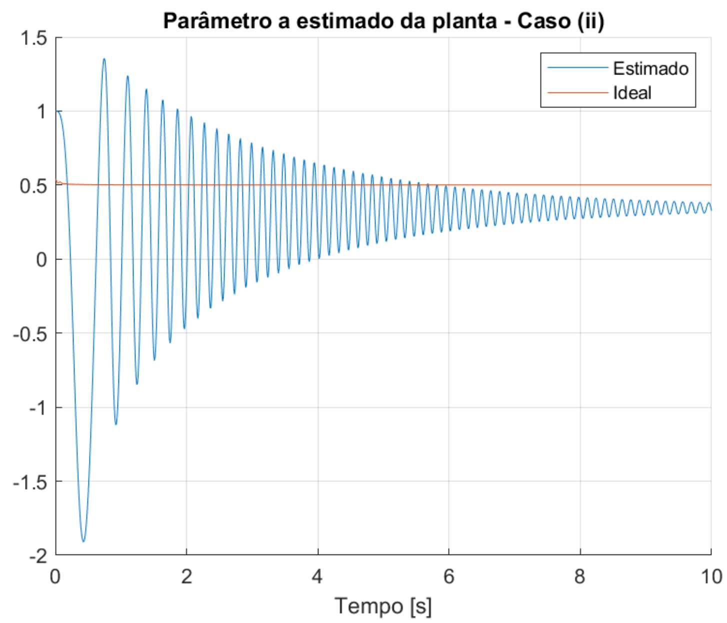


Figura 36: Adaptação de b para (i) e (ii) no MRAC Indireto



Fonte: Autoria própria

Figura 38: Adaptação de a para (ii) no MRAC Indireto

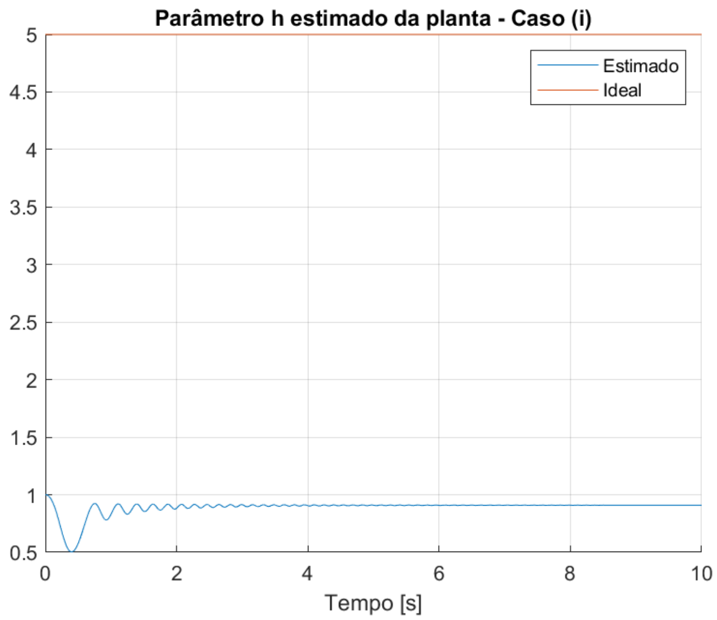


Fonte: Autoria própria

Ambos os parâmetros possuem uma progressão muito similar entre eles, mostrando que mesmo no caso (ii) o parâmetro a_m gerar uma pequena diferença inicial de valor esperado, a estimativa não é afetada.

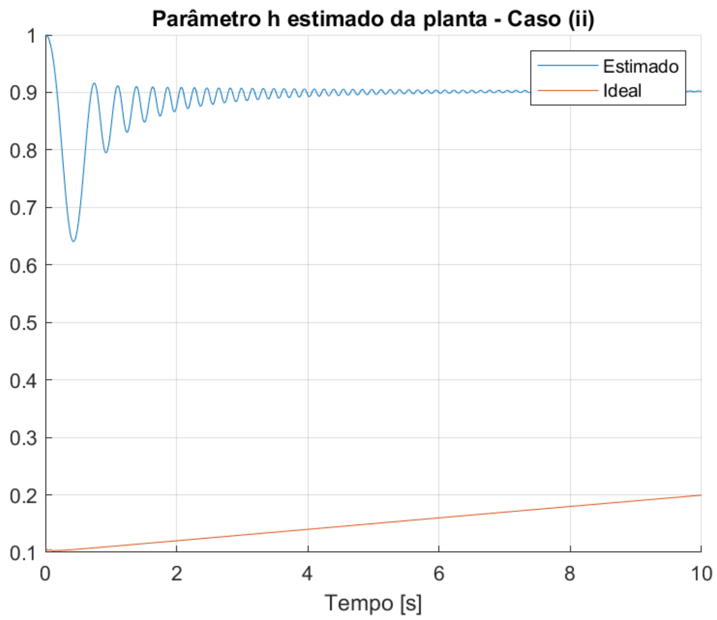
Da mesma forma, a comparação para o parâmetro $\hat{h}$ é feita nas Figuras 39 e 40.

Figura 39: Adaptação de h para (i) no MRAC Indireto



Fonte: Autoria própria

Figura 40: Adaptação de h para (ii) no MRAC Indireto



Fonte: Autoria própria

É possível perceber que, mesmo no caso (ii) o parâmetro ideal ser totalmente diferente do que no caso (i), em ambos os gráficos a estimativa do modelo segue o mesmo comportamento. Logo, conclui-se que a alteração mínima nos parâmetros já garante o acompanhamento do modelo de referência e, além disso, que o parâmetro $\hat{h}$ não possui tanta influência na saída final da planta.

9.1.5. e)

Como exercício extra, foi testado o desempenho do controlador MRAC Direto e Indireto sem a parcela δ de adaptação, ou seja, com os modelos originais discutidos em aulas. O resultado obtido foi o mesmo, com os mesmos gráficos a uma diferença mínima de 5% dos valores obtidos anteriormente. Logo, essa nova parcela inserida no controlador não possui grande influência na adaptação do controlador em geral. O teste se encontra no repositório de códigos como *Experiencia7c_ extra.m* e os gráficos serão omitidos deste relatório.

10. Experiência 7(d) – Controle Adaptativo

Considere a planta

$$y_p = \frac{b}{s+a} u_p \tag{81}$$

onde $b \geq 1$ e a são constantes desconhecidas. O modelo de referência é dado por

$$y_m = \frac{10}{s+10} r \tag{82}$$

Faça os seguintes itens:

a) Projete uma lei MRAC direta baseada no algoritmo do gradiente

b) Repita (a) para o algoritmo LS

c) Simule os projetos dos itens (a) e (b). Para ela, utilize $b = 3$, $a = 5$ e r um sinal de sua escolha.

10.1. Resolução

10.1.1. a)

A planta da Equação 81 pode ser reescrita pelo modelo de primeira ordem da Equação 83

$$\dot{y_p} = -a y_p + b u_p \tag{83}$$

A Lei de controle da Equação 84, com os valores definidos em 85 garantem que o sinal em malha fechada é limitada e que o estado y_p converge exponencialmente para o estado x_m do modelo de referência da Equação 82, com $a_m = 10$ e $b_m = 10$.

$$u = -k^* y_p + l^* r \tag{84}$$

$$\begin{cases} k^* = \dfrac{a_m - a}{b} \\ l^* = \dfrac{b_m}{b} \end{cases} \tag{85}$$

Como a e b são desconhecidos, os parâmetros k^* e l^* viram $k(t)$ e $l(t)$ por serem estimativas online ao longo do tempo. Somando e subtraindo $-bk^*x + bl^*r$ na Equação 81 e usando 85 para eliminar o parâmetro desconhecido a, obtem-se a Equação 86.

$$\dot{y_p} = -a_m y_p + b_m r + b(u_p + k^* y_p - l^* r) \quad (86)$$

onde junto com 82 e definindo $e = y_p - y_m$, tem-se a Equação 87.

$$e = \frac{b}{s + a_m}[u_p + k^* y_p - l^* r] \quad (87)$$

Essa pode ser reescrita na forma B-SPM conforme mostra a Equação

$$e = b(\theta^{*T}\phi + u_f), \quad (88)$$

onde $\theta^* = [k^*, l^*]$, $\phi = \frac{1}{s+a_m}[y_p, -r]^T$, $u_f = \dfrac{1}{s + a_m}u_p$. Usando as técnicas de adaptação, as estimativas para θ^* e b são dadas por

$$\begin{cases} \dot{\theta} = \Gamma\epsilon\phi\text{sgn}(b), \theta(t) = [k(t), l(t)]^T, \\ \dot{\hat{b}} = \gamma\epsilon\xi, \xi = \theta^T\phi + u_f, \\ \epsilon = \dfrac{e - \hat{b}(\theta^T\phi + u_f)}{m_s^2} = \dfrac{e - \hat{b}\xi}{m_s^2} \end{cases} \quad (89)$$

sendo o sinal normalizado $m_s^2 = 1 + \phi^T\phi + u_f^2$

10.1.2. c)

Para implementar as equações de adaptação para esse caso, a entrada adotada foi um degrau unitário e a seguinte ordem de cálculos foi adotada:

1- Calcular u_f filtrado;

2- Calcular ϕ filtrado;

3- Calcular ξ;

4- Estimar $\hat{b}$;

5- Calcular ϵ;

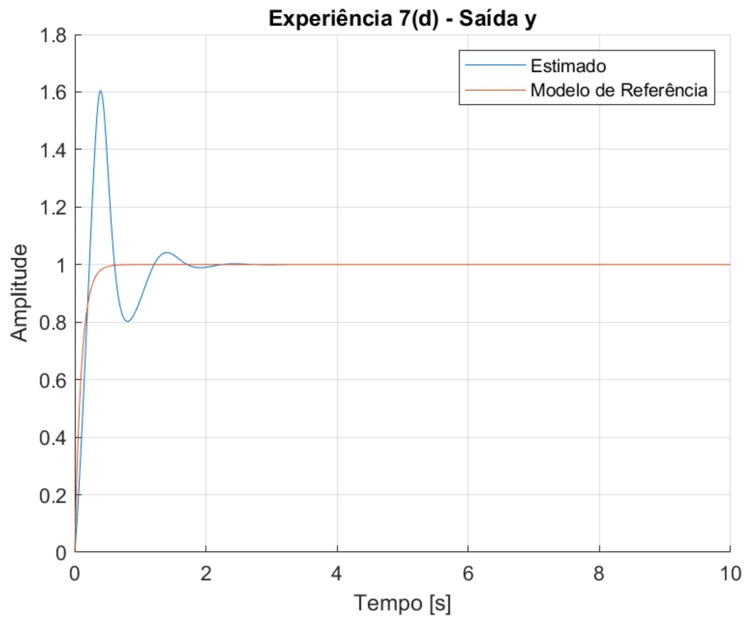
6- Estimar θ;

7- Calcular a nova ação de controle;

8- Simular a nova saída da planta.

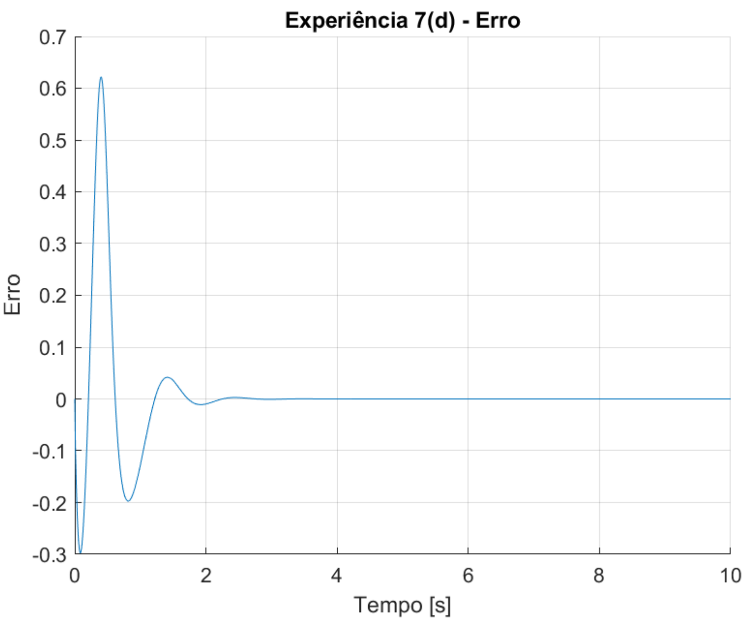
Dessa forma, os gráficos das Figuras 41 à 44 mostram os resultados encontrados para o método proposto.

Figura 41: Saída do sistema MRAC com gradiente



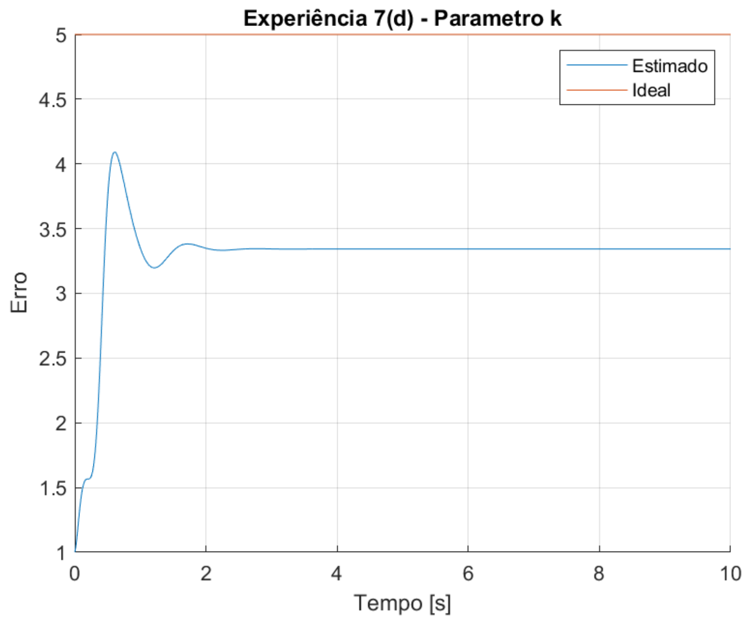
Fonte: Autoria própria

Figura 42: Erro do sistema MRAC com gradiente



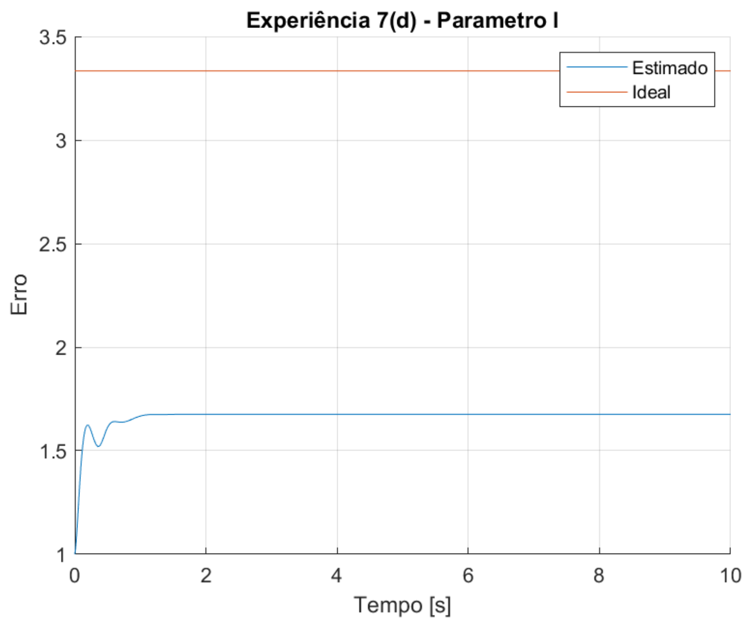
Fonte: Autoria própria

Figura 43: Parâmetro K do sistema MRAC com gradiente



Fonte: Autoria própria

Figura 44: Parâmetro L do sistema MRAC com gradiente



Fonte: Autoria própria

11. Experiência 8 – Controle Adaptativo

Considere a planta escalar representada por:

$$\dot{x} = ax + u \quad , \quad x(0) = x_0 \tag{90}$$

tal que a é uma constante desconhecida. A lei de controle por realimentação de estados definida por $u = -k(t)\,x$ faz com que o polo do sistema convirja para um polo desejado em $a_m > 0$ quando $t \to \infty$. Sendo $k^* = a + a_m$ e $\tilde{k} = k(t) - k^*$ o erro de estimativa do parâmetro k do regulador, dada a função de Lyapunov:

$$v\left(x,\, \tilde{k}\right) = \frac{x^2}{4} - \beta\tilde{k}^2 \tag{91}$$

demonstre que a lei de adaptação do parâmetro k que faz com que todos os sinais de malha fechada sejam limitados é dada por:

$$\dot{k} = -\frac{1}{4\beta}x^2 \tag{92}$$

11.1. Resolução

De acordo com Lyapunov, uma forma de provar que a lei de adaptação do parâmetro k dada por $\dot{k} = -\frac{1}{4\beta}x^2$ faz com que todos os sinais de malha fechada sejam limitados é demonstrando que:

$$\dot{V} = \frac{\partial V}{\partial x}\,\dot{x} + \frac{\partial\, V}{\partial \tilde{k}}\,\dot{\tilde{k}} \leq 0 \tag{93}$$

Então, com base na equação 91 podemos dizer que:

$$\dot{V} = \frac{x}{2}\,\dot{x} + 2\beta\tilde{k}\,\dot{\tilde{k}} \tag{94}$$

De acordo com as informações do enunciado, sabemos que $\dot{x} = ax + u$ e $\tilde{k} = k - k^*$. Então, substituindo essas informações na Equação 94, teremos:

$$\dot{V} = \frac{x}{2}\,(ax + u) + 2\beta\,(k - k^*)\,\dot{k} \tag{95}$$

Além disso, temos que $u = -kx$ e $k^* = a + a_m$. Portanto, substituindo esses valores e manipulando algebricamente a expressão, reescrevemos $\dot{V}$ como:

$$\dot{V} = \frac{x^2}{2}\,(a - k) + 2\beta\ \ (a - k + a_m)\,\dot{k} \tag{96}$$

Dessa forma, rearranjando os termos, podemos escrever como será mostrado na Equação 97 e garantir que os termos α e γ sejam ≤ 0.

$$\dot{V} = \underbrace{\frac{x^2}{2}\left(a-k\right) + 2\beta\ \left(a-k\right)\dot{k}}_{\alpha} + \underbrace{2\beta a_m \dot{k}}_{\gamma} \tag{97}$$

Inicialmente, manipulando o termo α ao igualá-lo a zero, teremos:

$$\dot{k} \cdot 2\beta\left(a-k\right) = -\frac{x^2}{2}\left(a-k\right) \tag{98}$$

Portanto, isolando $\dot{k}$, chegamos em:

$$\dot{k} = -\frac{x^2}{4\beta} \tag{99}$$

Nota-se que a Equação 99 é exatamente a Equação 92 fornecida pelo enunciado, então prova-se que para α ser nulo, este deve ser o valor de $\dot{k}$.

Além disso, após a obtenção de uma expressão para $\dot{k}$, podemos reescrever γ substituindo $\dot{k}$:

$$\gamma = 2\beta \cdot a_m \cdot \left(-\frac{x^2}{4\beta}\right) \tag{100}$$

Na Equação 100 a variável β será anulada, então γ pode ser reescrita como:

$$\gamma = -a_m \frac{x^2}{2} \tag{101}$$

Sabemos de acordo com o enunciado que $a_m > 0$, então a Equação 101 mostra que $\alpha \leq 0\,, \forall\, x \neq 0$.

Dessa forma podemos garantir que:

$$\dot{V} = \underbrace{\frac{x^2}{2}\left(a-k\right) + 2\beta\ \left(a-k\right)\dot{k}}_{\alpha} + \underbrace{2\beta a_m \dot{k}}_{\gamma} \leq 0 \tag{102}$$

Esta afirmação garante a estabilidade do sistema de acordo com o teorema de Lyapunov a partir da condição de que $\dot{k} = -\frac{x^2}{4\beta}$.

12. Experiência 9 – Controle Adaptativo

Seja o circuito de um filtro passa-baixa representado pelo seguinte modelo:

$$y = \frac{b}{s+a}\,u \tag{103}$$

Tem-se por objetivo ajustar a resposta deste circuito para que ele opere com frequência de corte em 1.061 Hz e ganho DC de 0.5. Dessa forma, considere o modelo de referência dado por:

$$y_m = \frac{3.3}{s+6.7}\,r \tag{104}$$

(a) Admita que os parâmetros a e b são exatamente conhecidos. Projete uma lei de controle por modelo de referência que faça com que todos os sinais de malha fechada sejam limitados e y convirja para y_m quando $t \to \infty$ para qualquer sinal de referência limitado r. Determine k^* e l^*.

(b) Admita que os parâmetros a e b são constantes e desconhecidos, com $b > 0$. Projete um MRAC direto de forma que todos os sinais de malha fechada sejam limitados e y convirja para y_m quando $t \to \infty$ para qualquer sinal de referência limitado r.

(c) Admita que os parâmetros a e b são constantes e desconhecidos, com $b > 0$. Projete um MRAC indireto de forma que todos os sinais de malha fechada sejam limitados e y convirja para y_m quando $t \to \infty$ para qualquer sinal de referência limitado r.

(d) Implemente as leis de controle obtidas em a), b) e c) considerando $a = 1$, $b = 5$ e $r = 15$. Gere os sinais de $y(t)$, $y_m(t)$, $u(t)$, $e(t)$, $k(t)$ e $l(t)$ para cada caso.

- ➢ Para cada caso, plote os sinais $y(t)$, $y_m(t)$ e $e(t)$ em um mesmo gráfico.
- ➢ Plote o sinal $u(t)$ em um mesmo gráfico para os três casos.
- ➢ Plote o sinal $k(t)$ em um mesmo gráfico para os três casos.
- ➢ Plote o sinal $l(t)$ em um mesmo gráfico para os três casos.

(e) Implemente as leis de controle obtidas em a), b) e c) considerando $a = 1$, $b = 5$ e $r = 2\,sen\,(10t) + 5\,sen\,(3t)$. Gere os sinais de $y(t)$, $y_m(t)$, $u(t)$, $e(t)$, $k(t)$ e $l(t)$ para cada caso.

- ➢ Para cada caso, plote os sinais $y(t)$, $y_m(t)$ e $e(t)$ em um mesmo gráfico.

- ➢ Plote o sinal $u(t)$ em um mesmo gráfico para os três casos.
- ➢ Plote o sinal $k(t)$ em um mesmo gráfico para os três casos.
- ➢ Plote o sinal $l(t)$ em um mesmo gráfico para os três casos.

(f) Os sinais $k(t)$ e $l(t)$ alcançaram k^*e l^*, respectivamente, em todos os casos? Comente.

12.1. Resolução

12.1.1. a)

Para o casso de a e b conhecidos, é necessário propor uma lei de controle com ganhos k^* e l^*, relacionados ao estado y e a entra r, respectivamente, que farão com que y se iguale ao modelo de referência. Logo, é proposta uma lei de controle do tipo:

$$u = -k^* y + l^* r \tag{105}$$

Substituindo a Equação 105 em 103 e igualando a equação ao modelo de referência da Equação 104, obtém-se:

$$\begin{gathered}
y = \frac{b}{s+a}\left(-k^* y + l^* r\right) \\
y(s + a + bk^*) = bl^* r \\
y = \frac{bl^*}{s + a + bk^*} \\
y = y_m \\
\frac{bl^*}{s + a + bk^*} = \frac{3.3}{s + 6.7}
\end{gathered} \tag{106}$$

Através da Equação 106, é possível obter as relações que descrevem k^* e l^*, apresentadas na Equação 107.

$$\begin{cases} k^* = \dfrac{6.7 - a}{b} \\ l^* = \dfrac{3.3}{b} \end{cases} \tag{107}$$

Substituindo esses valores na Equação 105, é possível obter a lei de controle em função dos parâmetros conhecidos da planta.

$$u = -\frac{6.7 - a}{b} x + \frac{3.3}{b} r \tag{108}$$

12.1.2. b)

No caso de a e b serem parâmetros desconhecidos, não é possível implementar a lei de controle do item anterior. Para isso, é necessário propor uma nova lei de controle na qual os parâmetros são estimações dos valores de k^* e l^*. Logo, a lei de controle proposta é apresentada na Equação 109.

$$u = -k(t)y + l(t)r \tag{109}$$

onde $k(t)$ e $l(t)$ são as estimativas de k^* e l^*, para cada instante t.

De forma análoga aos exercícios anteriores, é necessário manipular o modelo da planta e somar e subtrair bu^* para deixá-lo em função dos parâmetros do modelo de referência. A Equação 110 mostra o modelo comparável da planta.

$$y = \frac{b}{s+a}u$$
$$\dot{y} = -ay + bu$$
$$\dot{y} = -ay + bu + bu^* - bu^*$$
$$\dot{y} = -ay + bu - bk^*y + bl^*r + bk^*y - bl^*r$$
$$\dot{y} = -ay + bu - a_my + ay + b_mr + bk^*y - bl^*r$$
$$\dot{y} = -a_my + b_mr + b(k^*y - bl^*r + u) \tag{110}$$

É então definido o erro de acompanhamento do sistema como $e = y - y_m$, de forma análoga $\dot{e} = \dot{y} - \dot{y_m}$. Logo a dinâmica do erro é definida por:

$$\dot{e} = (-a_my + b_mr + b(k^*y - l^*r + u)) - (-a_my_m + b_mr)$$
$$\dot{e} = -a_m(y - y_m) + b(k^*y - bl^*r + u) \tag{111}$$

Substituindo u e considerando $\tilde{k} = k(t) - k^*$ e $\tilde{l} = l(t) - l^*$, é possível obter a seguinte expressão para o erro:

$$\dot{e} = -a_me + b(-\tilde{k}y + \tilde{l}) \tag{112}$$

Para projetar a lei adaptativa é necessário eleger uma função de Lyapunov candidata, nesse caso a função escolhida é apresentada na Equação 113:

$$V(e, \tilde{k}, \tilde{l}) = \frac{e^2}{2} + \frac{\tilde{k}^2}{2\gamma_1}|b| + \frac{\tilde{l}^2}{2\gamma_2}|b| \tag{113}$$

onde γ_1, $\gamma_2 > 0$

Derivando a função se tem:

$$\dot{V}(e, \tilde{k}, \tilde{l}) = -a_m e^2 - b\tilde{k}ex + b\tilde{l}er + \frac{|b|\tilde{k}}{\gamma_1}\dot{\tilde{k}} + \frac{|b|\tilde{l}}{\gamma_2}\dot{\tilde{l}} \tag{114}$$

Para garantir a estabilidade do sistema é necessário que $\dot{V}(e, \tilde{k}, \tilde{l}) <= 0$. O único termo que garante isso pelas condições conhecidas, é $-a_m e^2$.

A lei de adaptação pode ser obtida zerando os demais termos da função. Assim, são obtidas as seguintes relações:

$$\begin{cases} b\tilde{k}ex = \dfrac{|b|\tilde{k}}{\gamma_1}\dot{\tilde{k}} \\ b\tilde{l}er = \dfrac{|b|\tilde{l}}{\gamma_2}\dot{\tilde{l}} \end{cases} \tag{115}$$

Como k^* e l^* são constantes, então $\dot{\tilde{k}} = \dot{k}$ e $\dot{\tilde{l}} = \dot{l}$. Além disso, b pode ser escrito como $|b| \cdot sign(b)$, cancelando o módulo de b. Logo, as leis de adaptações podem ser reescritas conforme apresentado nas Equações 126:

$$\begin{cases} \dot{k} = \gamma_1 exsing(b) \\ \dot{l} = -\gamma_2 ersing(b) \end{cases} \tag{116}$$

12.1.3. c)

O modelo de referência a ser utilizado pode ser escrito conforme a Equação 117. Na qual $a_m = 6.7$ e $b_m = 3.3$.

$$\dot{y_m} = -a_m y_m + b_m r \tag{117}$$

Enquanto a planta é dada pela Equação 118.

$$\dot{y} = -ay + bu \tag{118}$$

Na qual a lei de controle u é definida pela Equação 119.

$$u = -k(t)y + l(t)r \tag{119}$$

Como $k(t)$ e $l(t)$ são parâmetros desconhecidos, são feitas estimativas para esses valores através dos parâmetros $\hat{a}$ e $\hat{b}$. Assim, as relações que descrevem $k(t)$ e $l(t)$ são apresentadas na Equação 120.

$$\begin{cases} k = \dfrac{a_m - \hat{a}}{\hat{b}} \\ l = \dfrac{b_m}{\hat{b}} \end{cases} \tag{120}$$

Os parâmetros $\hat{a}$ e $\hat{b}$ são gerados através das leis de adaptação. Para encontrar a lei de adaptação que dita esses parâmetros é realizada uma manipulação matemática somando e subtraindo o termo $a_m y$ no modelo da planta. Conforme apresentado na Equação 121.

$$\dot{y} = -ay + bu + a_m y - a_m y$$

$$\dot{y} = -a_m y + (a_m - a)y + bu \tag{121}$$

É então definido o erro de acompanhamento do sistema como $e = y - y_m$, de forma análoga $\dot{e} = \dot{y} - \dot{y_m}$. Para realizar essa comparação é utilizada uma aproximação de y_m, para $\hat{y}$, considerando que, em regime, $\dot{\hat{y}}$ tende para $\dot{y_m}$. A aproximação $\dot{\hat{y}}$ é apresentada na Equação122.

$$\dot{\hat{y}} = -a_m \hat{y} + (a_m - \hat{a})y + \hat{b}u \tag{122}$$

Definindo $\tilde{a} = \hat{a} - a$ e $\tilde{b} = \hat{b} - b$, a dinâmica do erro de acompanhamento pode ser dada pela expressão:

$$\dot{e} = -a_m e + \tilde{a}y - \tilde{b}u \tag{123}$$

Para projetar a lei adaptativa é necessário eleger uma função de Lyapunov candidata, nesse caso a função escolhida é apresentada na Equação 113:

$$V(e, \tilde{a}, \tilde{b}) = \frac{e^2}{2} + \frac{\tilde{a}^2}{2\gamma_1} + \frac{\tilde{b}^2}{2\gamma_2} \tag{124}$$

onde γ_1, $\gamma_2 > 0$

Derivando a função no tempo se tem:

$$\dot{V}(e, \tilde{a}, \tilde{b}) = -a_m e^2 - \tilde{a}ye - \tilde{b}ue + \frac{\dot{\tilde{a}}\tilde{a}}{\gamma_1} + \frac{\dot{\tilde{b}}\tilde{b}}{\gamma_2} \tag{125}$$

Para garantir a estabilidade do sistema é necessário que $\dot{V}(e, \tilde{a}, \tilde{l}) <= 0$. O único termo que garante isso pelas condições conhecidas, é $-a_m e^2$.

A lei de adaptação pode ser obtida zerando os demais termos da função. Como a e b são constantes, $\dot{\tilde{a}} = \dot{\hat{a}}$ e $\dot{\tilde{b}} = \dot{\hat{b}}$. Assim, são obtidas as seguintes relações para a lei de adaptação do MRAC indireto:

$$\begin{cases} \dot{\hat{a}} = ye\gamma_1 \\ \dot{\hat{b}} = ue\gamma_2 \end{cases} \tag{126}$$

12.1.4. d)

Considerando as Equações 126 e 126 obtidas nos itens (b) e (c), foi possível realizar a implementação dos sistemas MRAC Direto e Indireto. Os pseudocódigos para a implementação de ambos são vistos na sequência.

Algoritmo 2: MRAC Direto

Require: a_m, b_m
Ensure: $\hat{k}, \hat{l}$

```
  Definir K_0, L_0
  Definir γ_1, γ_2
  Adquirir x_m
  for i ← 2, ..., length(t) do
      e(i−1) ← x̂(i−1) − x_m(i−1)
      K̇ ← γ_1 e(i−1) x̂(i−1) sgn(b)
      K̂(i) ← K̇.dt + K̂(i−1)
      L̇ ← −γ_2 e(i−1) r̂(i−1) sgn(b)
      L̂(i) ← L̇.dt + L̂(i−1)
      û ← −K̂(i) x̂(i−1) + L̂(i) r̂(i−1)
      ẋ̂ ← a x̂(i−1) + b û                 ▷ Simulação do sistema controlado
      x̂(i) ← ẋ̂.dt + x̂(i−1)
  end for
```

Os algoritmos necessitam de parâmetros iniciais, tanto para o MRAC Direto quanto para o Indireto. Estes foram definidos como mostrados na sequência.

$$K_0 = 1, L_0 = 1, a_0 = 1, b_0 = 1$$

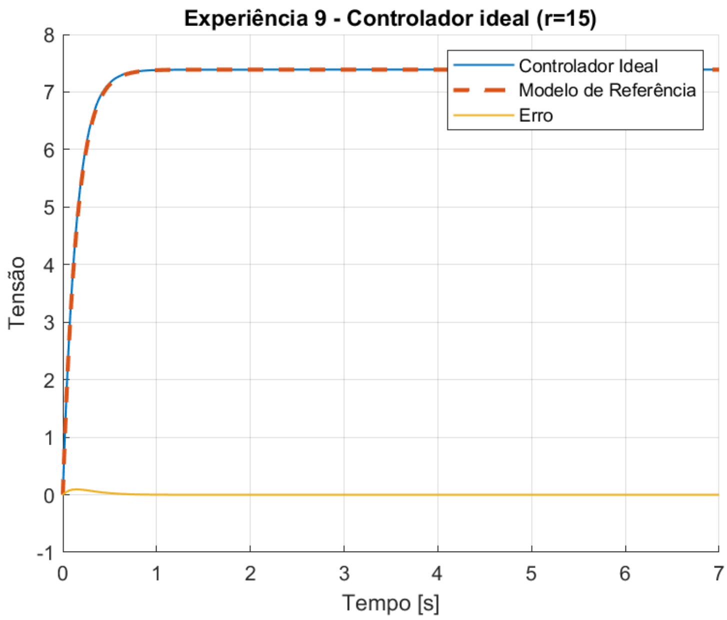
$$\gamma_1 = 1, \gamma_2 = 1$$

Dessa forma, os gráficos solicitados no enunciado, gerados em MATLAB, podem ser conferidos nas Figuras 45 até 52

Algoritmo 3: MRAC Indireto

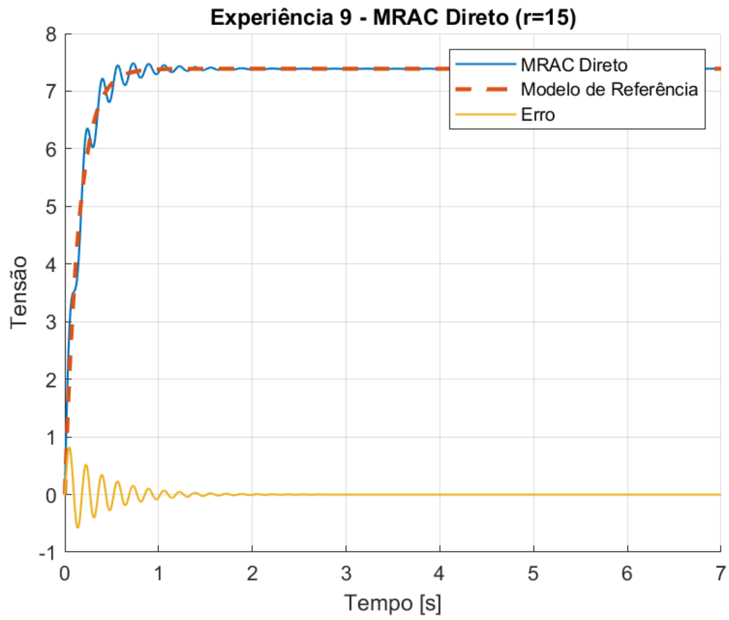
Require: a_m, b_m
Ensure: $\hat{a}, \hat{b}$
  `Definir` K_0, L_0, a_0, b_0, b_{min}
  `Definir` γ_1, γ_2
  `Adquirir` x_m
  for $i \leftarrow 2, \ldots, \texttt{length}(t)$ **do**
    $e(i-1) \leftarrow \hat{x}(i-1) - x_m(i-1)$
    $\hat{K}(i) \leftarrow (a_m + \hat{a}(i-1))/\hat{b}(i-1)$
    $\hat{L}(i) \leftarrow b_m/\hat{b}(i-1)$
    $\hat{u} \leftarrow -\hat{K}(i)\hat{x}(i-1) + \hat{L}(i)\hat{r}(i-1)$
    $\dot{\hat{a}} \leftarrow \gamma_1 e(i-1)\hat{x}(i-1)$
    $\hat{a}(i) \leftarrow \dot{\hat{a}}.dt + \hat{a}(i-1)$
    $\dot{\hat{b}} \leftarrow \gamma_2 e(i-1)\hat{u}$
    $\hat{b}(i) \leftarrow \dot{\hat{b}}.dt + \hat{b}(i-1)$
    $\hat{b}(i) \leftarrow \texttt{max}(\hat{b}(i), b_{min})$
    $\dot{\hat{x}} \leftarrow \hat{a}\hat{x}(i-1) + \hat{b}\hat{u}$ ▷ Simulação do sistema controlado
    $\hat{x}(i) \leftarrow \dot{\hat{x}}.dt + \hat{x}(i-1)$
  end for

Figura 45: Saída do sistema para o controlador ideal com $r = 15$



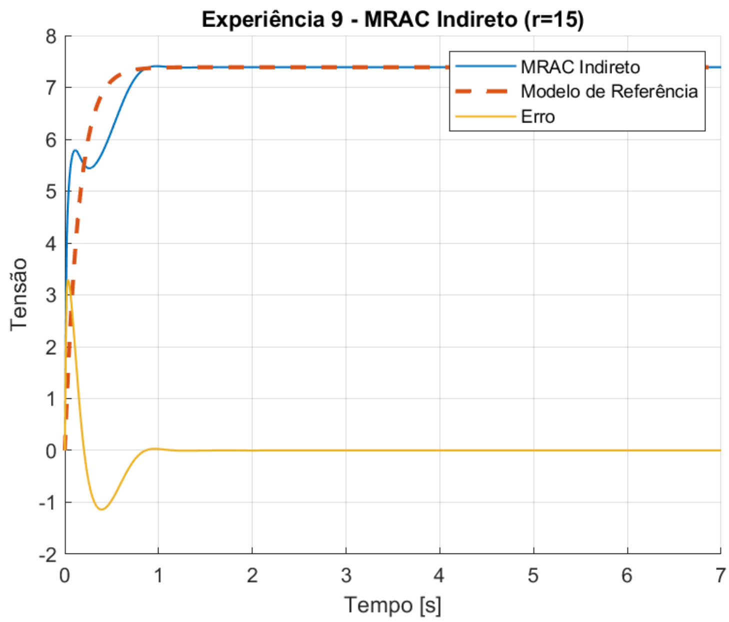
Fonte: Autoria própria

Figura 46: Saída do sistema para o MRAC Direto com $r = 15$



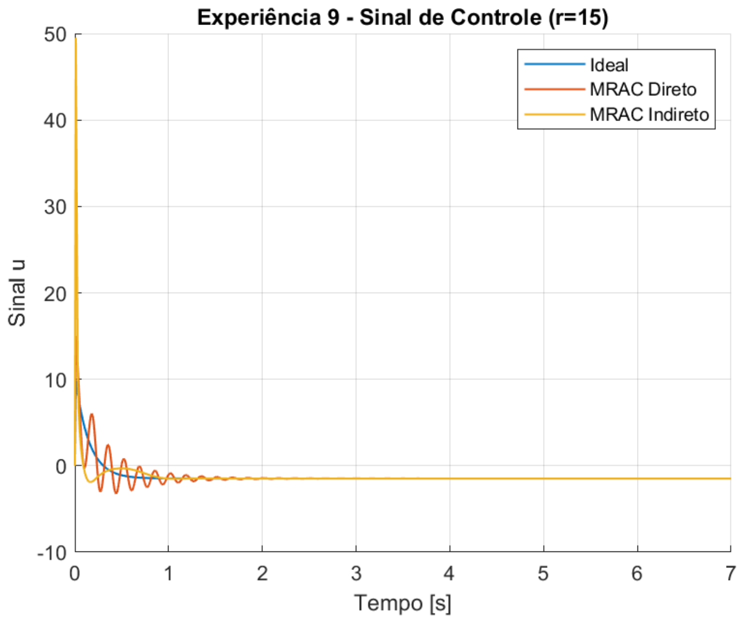
Fonte: Autoria própria

Figura 47: Saída do sistema para o MRAC Indireto com $r = 15$



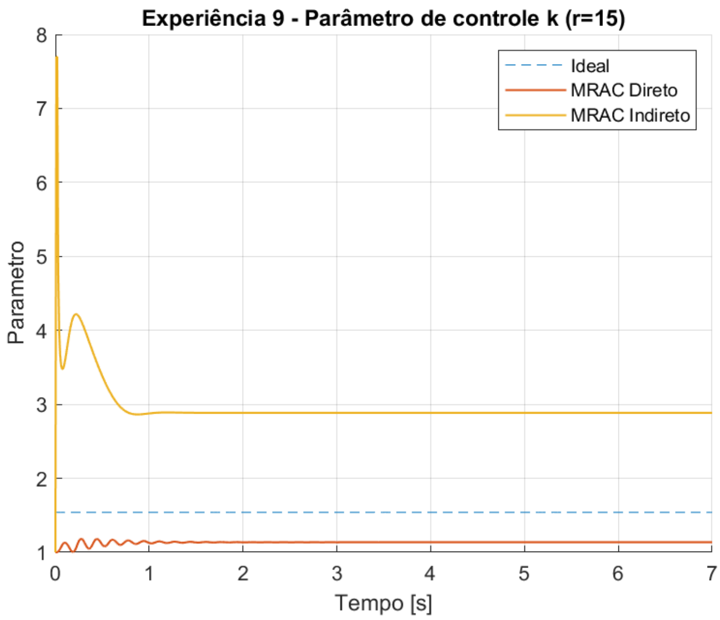
Fonte: Autoria própria

Figura 48: Sinal de controle para os três controladores com $r = 15$



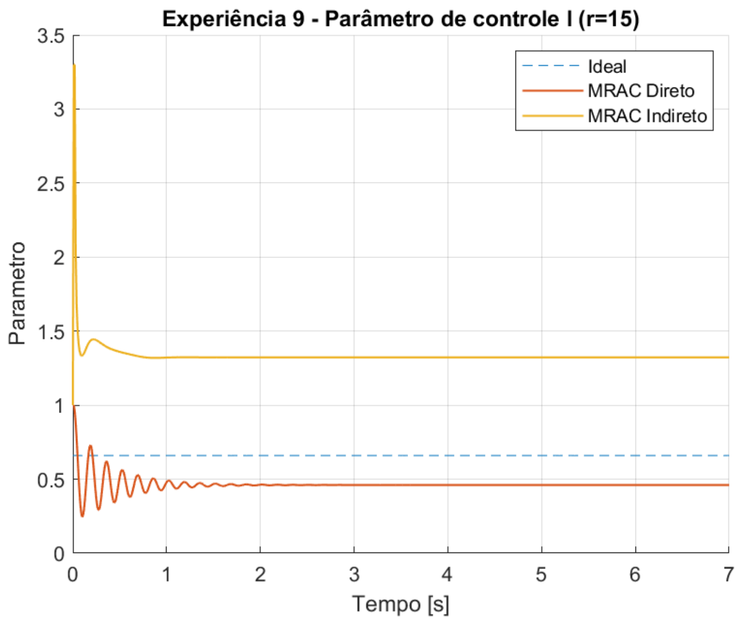
Fonte: Autoria própria

Figura 49: Parâmetro de controle $\hat{k}$ para os controladores adaptativos com $r = 15$



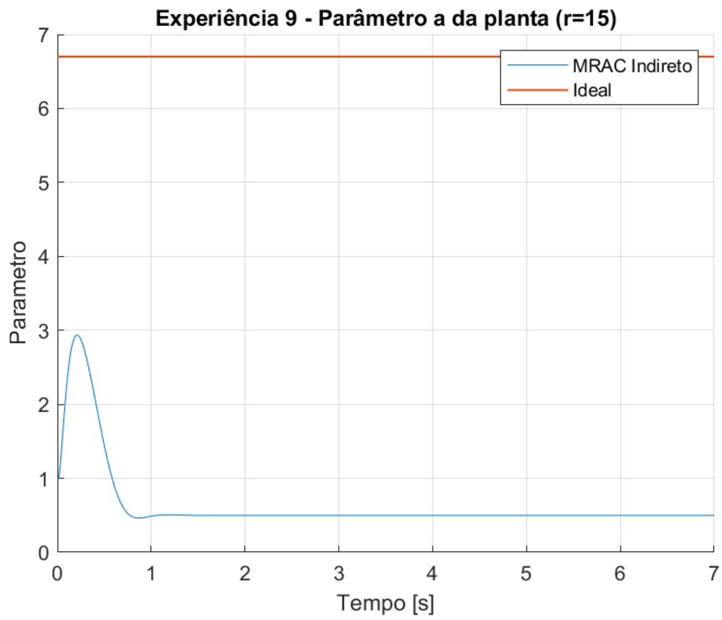
Fonte: Autoria própria

Figura 50: Parâmetro de controle $\hat{l}$ para os controladores adaptativos com $r = 15$



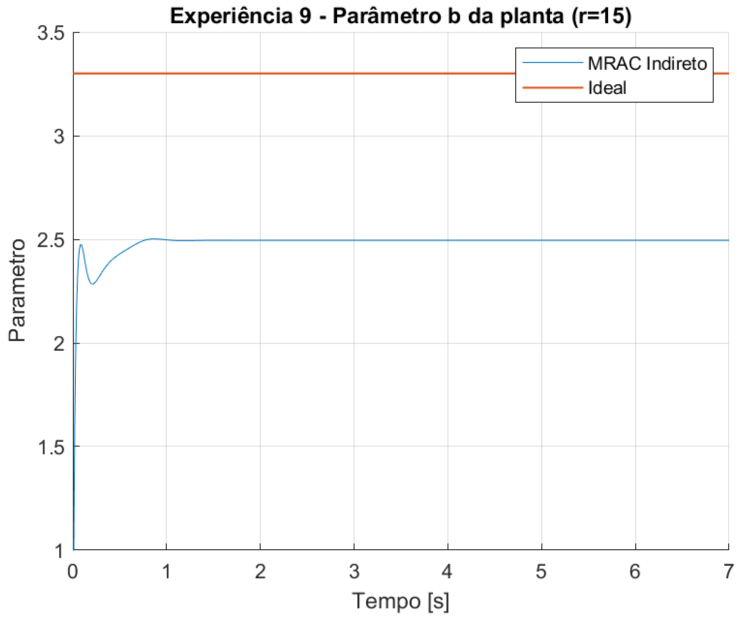
Fonte: Autoria própria

Figura 51: Parâmetro da planta $\hat{a}$ para o controlador indireto com $r = 15$



Fonte: Autoria própria

Figura 52: Parâmetro da planta $\hat{b}$ para o controlador indireto com $r = 15$



Fonte: Autoria própria

Na sequência é mostrado o algoritmo para a execução do controlador indireto.

```
%% Experiencia 7 (c) - Indireto (i)

% Valores
a = 0.5;
b = 1.5;
d = 10;

am = 0.5;
bm = 0.5;

% Valores ideais de controle
k_star = (am-a)/b;
l_star = bm/b;
delta_star = -d*a/b;

% Valores iniciais de adaptacao
a0 = 1;
b0 = 1;
d0 = 1;
b_limite=1;
k0 = 1;
l0 = 1;
```

```
delta0 = 1;

gamma1 = 0.5;
gamma2 = 0.5;
gamma3 = 0.5;


% Valores de V para o controlador adaptativo
V_hat_indireto = zeros(1,length(t));
V0_hat = 0;
V_hat_indireto(1) = V0_hat;

% Vetores para progressao dos ganhos, do erro e da planta
k_indireto = zeros(1,length(t));
k_indireto(1) = k0;
l_indireto = zeros(1,length(t));
l_indireto(1) = l0;
delta_indireto = zeros(1,length(t));
delta_indireto(1) = delta0;
e_indireto = zeros(1,length(t));
e_indireto(1) = V_hat_indireto(1)-Vm(1);

a_indireto = zeros(1,length(t));
a_indireto(1) = a0;
b_indireto = zeros(1,length(t));
b_indireto(1) = b0;
h_indireto = zeros(1,length(t));
h_indireto(1) = d0;
u_indireto = zeros(1,length(t));

for i=2:length(t)
    % V com controlador adaptativo
    e_indireto(i-1) = V_hat_indireto(i-1) - Vm(i-1);

    % Adaptacao dos parametros da planta
    a_indireto_dot = -gamma1*e_indireto(i-1)*V_hat_indireto(i-1);
    a_indireto(i) = a_indireto_dot*dt + a_indireto(i-1);

    if( abs(b_indireto(i-1)) > b_limite || ( (abs(b_indireto(i-1)) ==
        b_limite) && (sign(e_indireto(i-1)*u_indireto(i-1)*b)) >= 0))
        b_indireto_dot = gamma2*e_indireto(i-1)*u_indireto(i);
    else
        b_indireto_dot = 0;
```

```
        if b_indireto(i-1) > 0
            b_indireto(i-1) = abs(b_limite);
        else
            b_indireto(i-1) = -abs(b_limite);
        end
    end
    b_indireto(i) = b_indireto_dot*dt + b_indireto(i-1);

    h_indireto_dot = -gamma3*e_indireto(i-1);
    h_indireto(i) = h_indireto_dot*dt + h_indireto(i-1);

    % Atualizacao da lei de controle

    k_indireto_i = (am-a_indireto(i))/b_indireto(i);
    k_indireto(i) = k_indireto_i;

    l_indireto_i = (bm)/b_indireto(i);
    l_indireto(i) = l_indireto_i;

    delta_indireto_i = -(h_indireto(i))/b_indireto(i);
    delta_indireto(i) = delta_indireto_i;

    u_hat_indireto = -k_indireto(i)*V_hat_indireto(i-1) + l_indireto(i)*
       Vs(i-1) + delta_indireto(i);
    u_indireto(i) = u_hat_indireto;

    % Simulacao da planta

    V_hat_indireto_dot = -a*V_hat_indireto(i-1) + b*u_indireto(i) + d;
    V_hat_indireto(i) = V_hat_indireto_dot * dt + V_hat_indireto(i-1);
end

%% Experiencia 7 (c) - Indireto (ii)

b = 1.5;

% Valores iniciais de adaptacao
a0_ii = 1;
b0_ii = 1;
h0_ii = 1;
b_limite_ii=1;
k0_ii = 1;
l0_ii = 1;
```

```
delta0_ii = 1;

gamma1_ii = 0.5;
gamma2_ii = 0.5;
gamma3_ii = 0.5;


% Valores de V para o controlador adaptativo
V_hat_indireto_ii = zeros(1,length(t));
V0_hat_ii = 0;
V_hat_indireto_ii(1) = V0_hat_ii;

% Vetores para progressao dos ganhos, do erro e da planta
k_indireto_ii = zeros(1,length(t));
k_indireto_ii(1) = k0_ii;
l_indireto_ii = zeros(1,length(t));
l_indireto_ii(1) = l0_ii;
delta_indireto_ii = zeros(1,length(t));
delta_indireto_ii(1) = delta0_ii;
e_indireto_ii = zeros(1,length(t));
e_indireto_ii(1) = V_hat_indireto_ii(1)-Vm(1);

a_indireto_ii = zeros(1,length(t));
a_indireto_ii(1) = a0_ii;
b_indireto_ii = zeros(1,length(t));
b_indireto_ii(1) = b0_ii;
h_indireto_ii = zeros(1,length(t));
h_indireto_ii(1) = h0_ii;

a_planta = zeros(1,length(t));
a_planta(1) = 0.5 + 0.04/(1+V0_hat);
d_planta = zeros(1,length(t));
d_planta(1) = 0.2 + sin(0.02*0);


u_indireto_ii = zeros(1,length(t));

for i=2:length(t)
    % V com controlador adaptativo
    e_indireto_ii(i-1) = V_hat_indireto_ii(i-1) - Vm(i-1);

    % Adaptacao dos parametros da planta
    a_indireto_dot = -gamma1_ii*e_indireto_ii(i-1)*V_hat_indireto_ii(i
```

```
        -1);
    a_indireto_ii(i) = a_indireto_dot*dt + a_indireto_ii(i-1);

    if( abs(b_indireto_ii(i-1)) > b_limite_ii || ( (abs(b_indireto_ii(i
        -1)) == b_limite_ii) && (sign(e_indireto_ii(i-1)*u_indireto_ii(i
        -1)*b)) >= 0))
        b_indireto_dot = gamma2_ii*e_indireto_ii(i-1)*u_indireto_ii(i);
    else
        b_indireto_dot = 0;
        if b_indireto_ii(i-1) > 0
            b_indireto_ii(i-1) = abs(b_limite_ii);
        else
            b_indireto_ii(i-1) = -abs(b_limite_ii);
        end
    end
    b_indireto_ii(i) = b_indireto_dot*dt + b_indireto_ii(i-1);

    h_indireto_dot = -gamma3_ii*e_indireto_ii(i-1);
    h_indireto_ii(i) = h_indireto_dot*dt + h_indireto_ii(i-1);

    % Atualizacao da lei de controle

    k_indireto_i2 = (am-a_indireto_ii(i))/b_indireto_ii(i);
    k_indireto_ii(i) = k_indireto_i2;

    l_indireto_i2 = (bm)/b_indireto_ii(i);
    l_indireto_ii(i) = l_indireto_i2;

    delta_indireto_i2 = -(h_indireto_ii(i))/b_indireto_ii(i);
    delta_indireto_ii(i) = delta_indireto_i2;

    u_hat_indireto_ii = -k_indireto_ii(i)*V_hat_indireto_ii(i-1) +
        l_indireto_ii(i)*Vs(i-1) + delta_indireto_ii(i);
    u_indireto_ii(i) = u_hat_indireto_ii;

    % Simulacao da planta
    a_planta(i) = a + 0.04/(1+V_hat_ii(i-1));
    d_planta(i) = 0.2 + sin(0.02*t(i-1));

    V_hat_indireto_dot_ii = -a_planta(i)*V_hat_indireto_ii(i-1) + b*
        u_indireto_ii(i) + d_planta(i)*a_planta(i);
    V_hat_indireto_ii(i) = V_hat_indireto_dot_ii * dt +
        V_hat_indireto_ii(i-1);
```

```
end

%% Plot dos graficos indireto

% Plot de velocidade

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(V_hat_indireto)
plot(V_hat_indireto_ii)
plot(Vm)
legend("Estimado (i)", "Estimado (ii)", "Ideal")
xlabel("Tempo [s]")
ylabel("Velocidade")
title("Velocidade pelo controlador indireto")
if saveFig
    saveas(gcf,'Exp7c_Velocidade_Indireto.png')
end

% Plot de e

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,e_indireto)
plot(t,e_indireto_ii)
legend("Estimado (i)", "Estimado (ii)")
xlabel("Tempo [s]")
ylabel("Erro")
title("Erro para os controladores indiretos")
if saveFig
    saveas(gcf,'Exp7c_Erro_Indireto.png')
end

% Plot de k

k_star_vector = k_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
```

```
grid on
plot(t,k_indireto)
plot(t,k_indireto_ii)
plot(t, k_star_vector, '--')
legend("Estimado (i)", "Estimado (ii)", "Ideal")
xlabel("Tempo [s]")
title("Parametro K do controlador indireto")
if saveFig
    saveas(gcf,'Exp7c_ParametroK_Indireto.png')
end

% Plot de l

l_star_vector = l_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,l_indireto)
plot(t,l_indireto_ii)
plot(t, l_star_vector, '--')
legend("Estimado (i)","Estimado (ii)", "Ideal")
xlabel("Tempo [s]")
title("Parametro L do controlador indireto")
if saveFig
    saveas(gcf,'Exp7c_ParametroL_Indireto.png')
end

% Plot de delta

delta_star_vector = delta_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,delta_indireto)
plot(t,delta_indireto_ii)
plot(t, delta_star_vector, '--')
legend("Estimado (i)","Estimado (ii)", "Ideal")
xlabel("Tempo [s]")
title("Parametro \delta do controlador indireto")
if saveFig
    saveas(gcf,'Exp7c_ParametroDelta_Indireto.png')
```

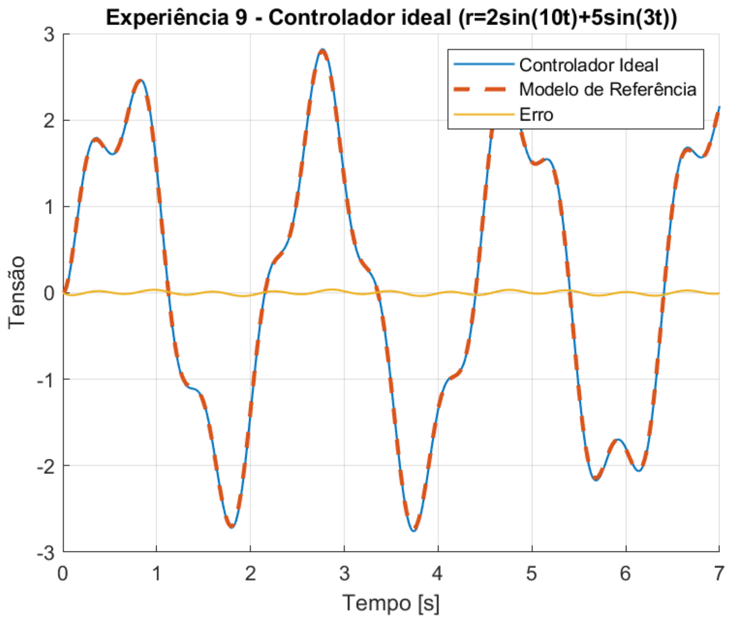
```
end

% Plot de b

b_vector = b*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,b_indireto)
plot(t,b_indireto_ii)
plot(t, b_vector, '--')
legend("Estimado (i)","Estimado (ii)","Ideal")
xlabel("Tempo [s]")
title("Parametro b estimado da planta")
if saveFig
    saveas(gcf,'Exp7c_ParametroB_Indireto.png')
end

% Plot de a - i

a_vector = a*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,a_indireto)
plot(t, a_vector)
legend("Estimado","Ideal")
xlabel("Tempo [s]")
title("Parametro a estimado da planta - Caso (i)")
if saveFig
    saveas(gcf,'Exp7c_ParametroA_i_Indireto.png')
end

% Plot de a - ii

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,a_indireto_ii)
plot(t, a_planta)
```

```
legend("Estimado","Ideal")
xlabel("Tempo [s]")
title("Parametro a estimado da planta - Caso (ii)")
if saveFig
    saveas(gcf,'Exp7c_ParametroA_ii_Indireto.png')
end

% Plot de h - i

h_vector = (a*d)*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,h_indireto)
plot(t, h_vector)
legend("Estimado","Ideal")
xlabel("Tempo [s]")
title("Parametro h estimado da planta - Caso (i)")
if saveFig
    saveas(gcf,'Exp7c_ParametroH_i_Indireto.png')
end

% Plot de h - ii

index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t,h_indireto_ii)
plot(t, d_planta.*a_planta)
legend("Estimado","Ideal")
xlabel("Tempo [s]")
title("Parametro h estimado da planta - Caso (ii)")
if saveFig
    saveas(gcf,'Exp7c_ParametroH_ii_Indireto.png')
end
```

12.1.5. e)

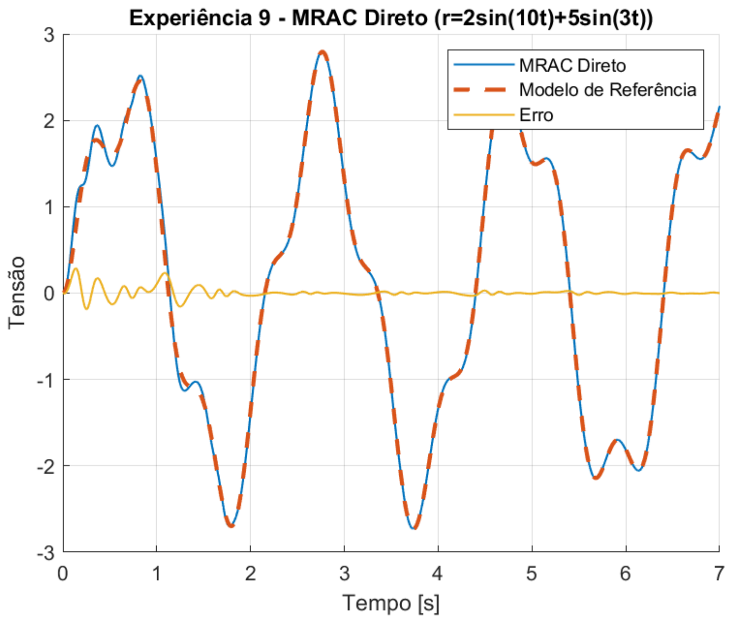
Considerando agora as mesmas equações e algoritmos, a entrada $r = 2\sin(10t) + 5\sin(3t)$ é inserida no sistema, gerando os gráficos das Figuras 53 até 60 a seguir.

Figura 53: Saída do sistema para o controlador ideal com $r = 2\sin(10t) + 5\sin(3t)$



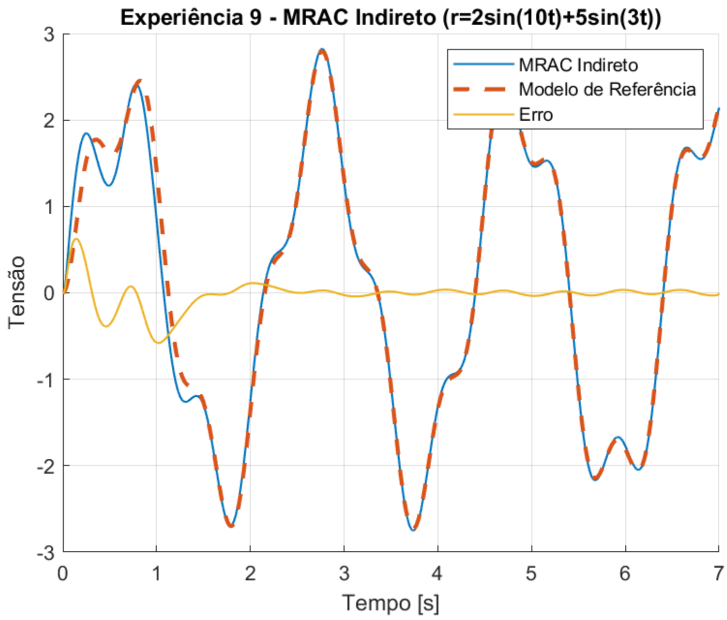
Fonte: Autoria própria

Figura 54: Saída do sistema para o MRAC Direto com $r = 2\sin(10t) + 5\sin(3t)$



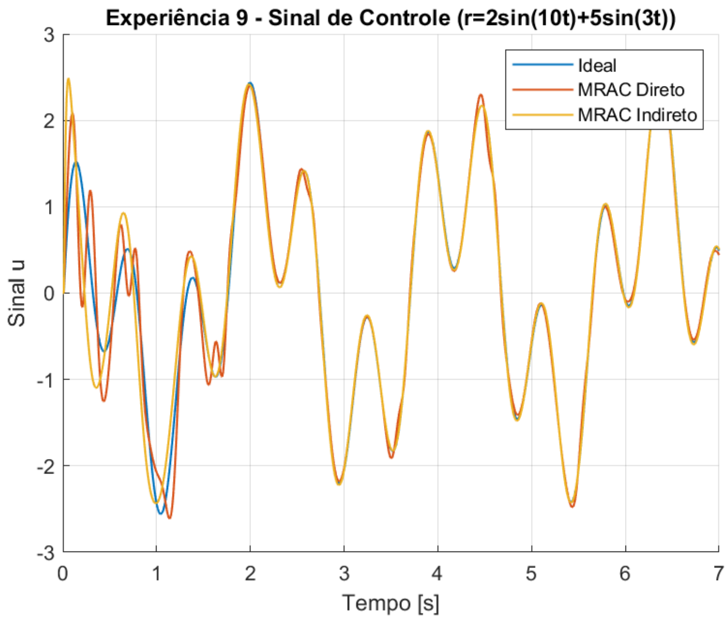
Fonte: Autoria própria

Figura 55: Saída do sistema para o MRAC Indireto com $r = 2\sin(10t) + 5\sin(3t)$



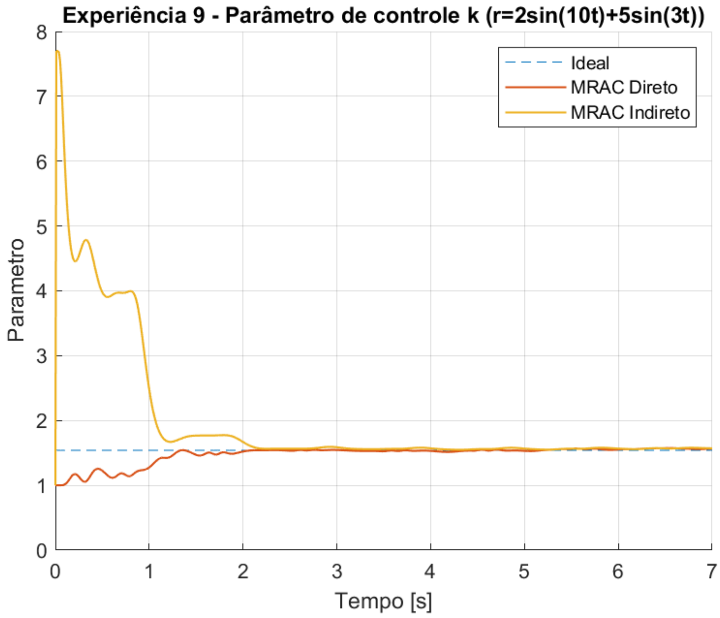
Fonte: Autoria própria

Figura 56: Sinal de controle para os três controladores com $r = 2\sin(10t) + 5\sin(3t)$



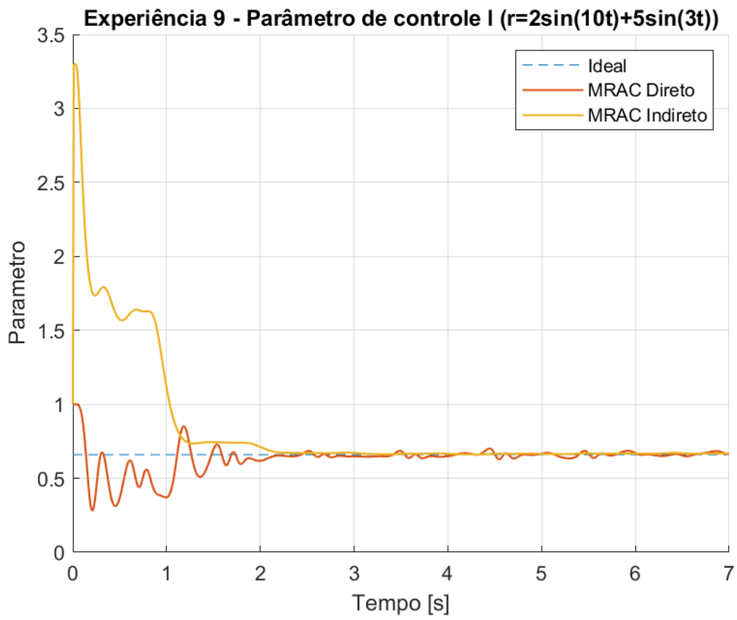
Fonte: Autoria própria

Figura 57: Parâmetro de controle $\hat{k}$ para os controladores adaptativos com $r = 2\sin(10t) + 5\sin(3t)$



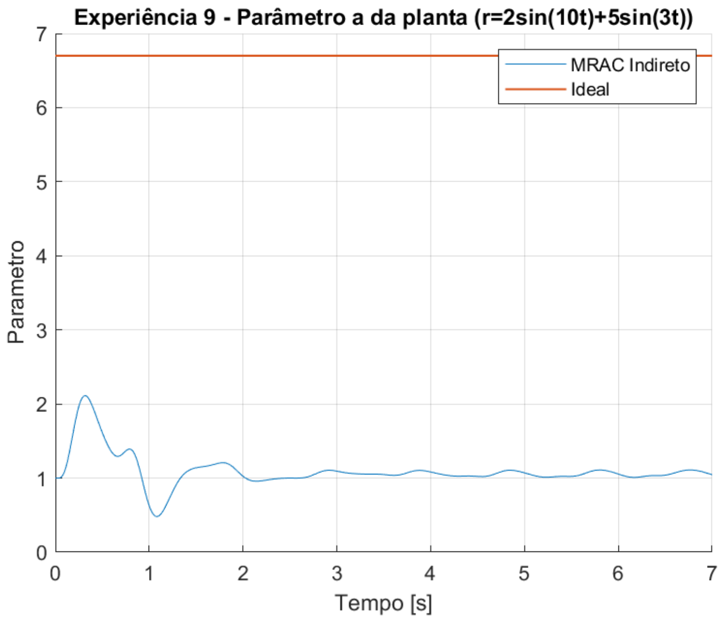
Fonte: Autoria própria

Figura 58: Parâmetro de controle $\hat{l}$ para os controladores adaptativos com $r = 2\sin(10t) + 5\sin(3t)$



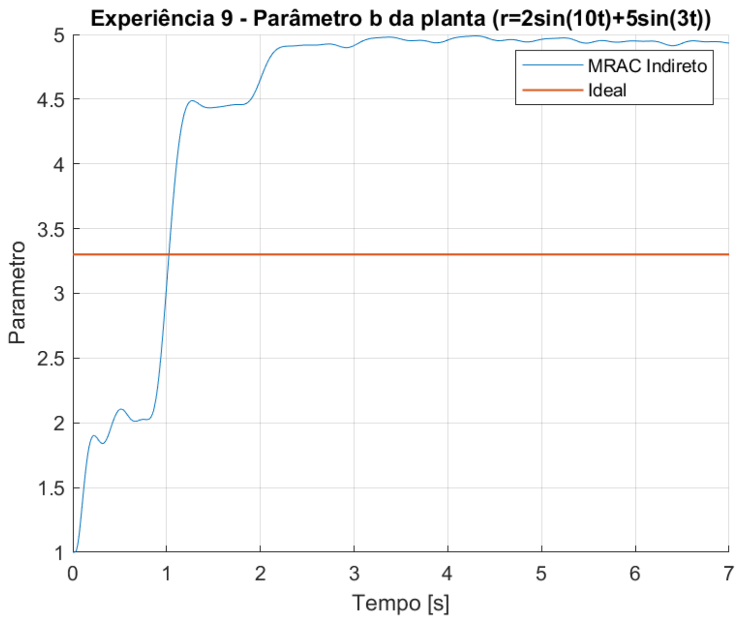
Fonte: Autoria própria

Figura 59: Parâmetro da planta $\hat{a}$ para o controlador indireto com $r = 2\sin(10t) + 5\sin(3t)$



Fonte: Autoria própria

Figura 60: Parâmetro da planta $\hat{b}$ para o controlador indireto com $r = 2\sin(10t) + 5\sin(3t)$



Fonte: Autoria própria

12.1.6. f)

Com relação ao item (d), primeiramente pode-se levantar comparações acerca dos gráficos 45, 46 e 47. Para o controle ideal, percebe-se que o sinal de saída acompanha com praticamente erro zero, acompanhando fielmente o modelo de referência do sistema. Enquanto isso, o MRAC Direto ainda possui um bom desempenho, apesar de oscilatório, em acompanhar o modelo de referência, e o MRAC Indireto possui um desvio maior no transitório.

Já para o gráfico da Figura 48, é notável a presença de um pico para o MRAC Indireto, enquanto os outros mantém a mesma faixa de valores. Isso pode ser levado em consideração na implementação, sabendo que esses picos podem gerar danos ao sistema.

Todos os parâmetros apresentados nesse caso, $\hat{k}$, $\hat{l}$, $\hat{a}$ e $\hat{b}$, com progressões temporais mostradas nas Figuras 49, 50, 51 e 52, respectivamente, não chegam a atingir os valores desejados e considerados como ideais. Entretanto, assumem um bom desempenho ao fazer com que a saída acompanhe x_m.

Outro ponto importante a se ressaltar é que o objetivo da implementação do controlador foi atingido, a planta está se comportando como um filtro passa-baixas com ganho DC de 0.5. Isso é comprovado no valor de regime permanente ao degrau $r = 15$, sendo $\hat{x}_\infty = 7.388$

Tomando como base agora o item (e), as mesmas observações sobre a saída da planta das Figuras 53, 54 e 55 podem ser feitas para este caso também, tendo como ordem de maior para menor desempenho o Ideal, seguido pelo MRAC Direto, e fechando com o MRAC Indireto.

Para esta entrada os parâmetros apresentados, $\hat{k}$, $\hat{l}$, $\hat{a}$ e $\hat{b}$, com progressões temporais mostradas nas Figuras 49, 50, 51 e 52, respectivamente, também não chegam a atingir os valores desejados e considerados como ideais. Entretanto, assumem um bom desempenho ao fazer com que a saída acompanhe x_m.

Tendo em vista o comportamento do sistema como o filtro passa-baixas de 1 Hz, é possível perceber que a frequência de 3 rad/s (menor que 1 Hz) é muito presente no sinal, enquanto a frequência maior de 10 rad/s (maior que 3 Hz) é muito atenuada, gerando pequenos desvios na onda de saída, mostrando que o objetivo do sistema para frequência também foi atingido.

O algoritmo utilizado para gerar os gráficos e resultados dessa questão pode ser visto na sequência. Esse código seleciona qual entrada o usuário deseja e pode salvar os gráficos automaticamente a partir da flag saveFig.

```
clc
```

```
clear
close all

index_fig = 0;

%% Flags

saveFig = false;
%% Entrada

disp("Entrada 1: r=15");
disp("Entrada 2: r=2*sin(10*t) + 5*sin(3*t)")

entrada = input("Qual voce deseja? ");
clc;

%% Experiencia 9 - (d)

index_fig = 0;

% Parametros da planta
a = 1;
b = 5;
% Parametros do modelo de referencia
am = 6.7;
bm = 3.3;

% Valores ideais de controle
k_star = (a+am)/b;
l_star = bm/b;

% Vetor de tempo
dt = 0.01;
t = 0:dt:7;

% Sinal de entrada
if entrada==1
    r = 15;
    r = r*ones(1,length(t));
else
    r = 2*sin(10*t) + 5*sin(3*t);
end

```

```
% Valores de x para o controlador ideal
x0 = 0;
x = zeros(1,length(t));
x(1) = x0;

% Funcao de transferencia modelo de referencia
s = tf('s');
G = bm/(s+am);
xm = lsim(G, r, t);

% Valores iniciais de adaptacao
k0 = 1;
l0 = 1;
% Sinal de entrada
if entrada==1
    gamma1 = 1;
    gamma2 = 1;
else
    gamma1 = 7;
    gamma2 = 7;
end

% Valores de x para o controlador adaptativo
x_hat_direto = zeros(1,length(t));
x0_hat = 0;
x_hat_direto(1) = x0_hat;

% Vetores para progressao dos ganhos e do erro
k_direto = zeros(1,length(t));
k_direto(1) = k0;
l_direto = zeros(1,length(t));
l_direto(1) = l0;
e_direto = zeros(1,length(t));
e_direto(1) = x_hat_direto(1)-xm(1);

u_direto = zeros(1,length(t));

e = zeros(1,length(t));
e(1) = x(1)-xm(1);

u_ideal = zeros(1,length(t));

for i=2:length(t)
```

```
    % X com controlador adaptativo
    e_direto(i-1) = x_hat_direto(i-1) - xm(i-1);

    k_direto_dot = gamma1*e_direto(i-1)*x_hat_direto(i-1)*sign(b);
    k_direto(i) = k_direto_dot*dt + k_direto(i-1);

    l_direto_dot = -gamma2*e_direto(i-1)*r(i-1)*sign(b);
    l_direto(i) = l_direto_dot*dt + l_direto(i-1);

    u_hat_direto = -k_direto(i)*x_hat_direto(i-1) + l_direto(i)*r(i-1);
    u_direto(i) = u_hat_direto;

    x_hat_direto_dot = a*x_hat_direto(i-1) + b*u_hat_direto;
    x_hat_direto(i) = x_hat_direto_dot * dt + x_hat_direto(i-1);

    % X como controlador ideal
    e(i-1) = x(i-1) - xm(i-1);
    u = -k_star*x(i-1) + l_star*r(i-1);
    u_ideal(i) = u;
    x_dot = a*x(i-1) + b*u;
    x_i = x(i-1) + x_dot*dt;
    x(i) = x_i;
end

%% Experiencia 9 - MRAC Indireto

% Valores iniciais de adaptacao
a0 = 1;
b0 = 1;
b_limite=1;
if entrada==1
    gamma1 = 1;
    gamma2 = 1;
else
    gamma1 = 7;
    gamma2 = 7;
end

% Valores de x para o controlador adaptativo
x_hat_indireto = zeros(1,length(t));
x0_hat = 0;
x_hat_indireto(1) = x0_hat;

```

```
% Vetores para progressao dos ganhos e do erro
k_indireto = zeros(1,length(t));
k_indireto(1) = k0;
l_indireto = zeros(1,length(t));
l_indireto(1) = l0;
e_indireto = zeros(1,length(t));
e_indireto(1) = x_hat_indireto(1)-xm(1);

a_indireto = zeros(1,length(t));
a_indireto(1) = a0;
b_indireto = zeros(1,length(t));
b_indireto(1) = b0;

u_indireto = zeros(1,length(t));

for i=2:length(t)
    % X com controlador adaptativo
    e_indireto(i-1) = x_hat_indireto(i-1) - xm(i-1);

    k_indireto_i = (am+a_indireto(i-1))/b_indireto(i-1);
    k_indireto(i) = k_indireto_i;

    l_indireto_i = (bm)/b_indireto(i-1);
    l_indireto(i) = l_indireto_i;

    u_hat_indireto = -k_indireto(i)*x_hat_indireto(i-1) + l_indireto(i)*
       r(i-1);
    u_indireto(i) = u_hat_indireto;

    a_indireto_dot = gamma1*e_indireto(i-1)*x_hat_indireto(i-1);
    a_indireto(i) = a_indireto_dot*dt + a_indireto(i-1);

    b_indireto_dot = gamma2*e_indireto(i-1)*u_hat_indireto;
    b_indireto(i) = b_indireto_dot*dt + b_indireto(i-1);

    if ~ ( (abs(b_indireto(i)) > abs(b_limite)) || ( (abs(b_indireto(i))
        > abs(b_limite)) && (e_indireto(i-1)*u_hat_indireto*sgn(b) > 0)
       ) )
        if b_indireto(i) < 0
            b_indireto(i) = - abs(b_limite);
        else
            b_indireto(i) = abs(b_limite);
        end
```

```
        end

        x_hat_indireto_dot = a*x_hat_indireto(i-1) + b*u_hat_indireto;
        x_hat_indireto(i) = x_hat_indireto_dot * dt + x_hat_indireto(i-1);
end


%% Experiencia 7 (a) - Plots

if entrada == 1
        titulo = "(r=15)";
else
        titulo = "(r=2sin(10t)+5sin(3t))";
end

% Plot do controlador ideal
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, x, "LineWidth", 1)
plot(t, xm, '--', "LineWidth", 2)
plot(t, e, "LineWidth", 1)
legend("Controlador Ideal", "Modelo de Referencia", "Erro")
xlabel("Tempo [s]")
ylabel("Tensao")
title("Experiencia 9 - Controlador ideal " + titulo)
if saveFig
        saveas(gcf,'Exp9_'+string(entrada)'+'_EstadosIdeal.png')
end

% Plot do controlador direto
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, x_hat_direto, "LineWidth", 1)
plot(t, xm, '--', "LineWidth", 2)
plot(t, e_direto, "LineWidth", 1)
legend("MRAC Direto", "Modelo de Referencia", "Erro")
xlabel("Tempo [s]")
ylabel("Tensao")
title("Experiencia 9 - MRAC Direto " + titulo)
```

```
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_EstadosMRACDireto.png')
end

% Plot do controlador indireto
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, x_hat_indireto, "LineWidth", 1)
plot(t, xm, '--', "LineWidth", 2)
plot(t, e_indireto, "LineWidth", 1)
legend("MRAC Indireto", "Modelo de Referencia", "Erro")
xlabel("Tempo [s]")
ylabel("Tensao")
title("Experiencia 9 - MRAC Indireto " + titulo)
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_EstadosMRACIndireto.png')
end

% Sinal de controle
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, u_ideal, "LineWidth", 1)
plot(t, u_direto, "LineWidth", 1)
plot(t, u_indireto, "LineWidth", 1)
legend("Ideal","MRAC Direto","MRAC Indireto")
xlabel("Tempo [s]")
ylabel("Sinal u")
title("Experiencia 9 - Sinal de Controle " + titulo)
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_SinalControle.png')
end

k_star_vector = k_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, k_star_vector, '--')
plot(t, k_direto, "LineWidth", 1)
```

```
plot(t, k_indireto, "LineWidth", 1)
legend("Ideal","MRAC Direto","MRAC Indireto")
xlabel("Tempo [s]")
ylabel("Parametro")
title("Experiencia 9 - Parametro de controle k " + titulo)
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_ParametroK.png')
end

l_star_vector = l_star*ones(1, length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, l_star_vector, '--')
plot(t, l_direto, "LineWidth", 1)
plot(t, l_indireto, "LineWidth", 1)
legend("Ideal","MRAC Direto","MRAC Indireto")
xlabel("Tempo [s]")
ylabel("Parametro")
title("Experiencia 9 - Parametro de controle l " + titulo)
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_ParametroL.png')
end

am_vetor = am*ones(1,length(t));
index_fig = index_fig + 1;
figure(index_fig)
hold on
grid on
plot(t, a_indireto)
plot(t, am_vetor, "LineWidth", 1)
legend("MRAC Indireto","Ideal")
xlabel("Tempo [s]")
ylabel("Parametro")
title("Experiencia 9 - Parametro a da planta " + titulo)
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_ParametroA.png')
end

bm_vetor = bm*ones(1,length(t));
index_fig = index_fig + 1;
figure(index_fig)
```

```
hold on
grid on
plot(t, b_indireto)
plot(t, bm_vetor, "LineWidth", 1)
legend("MRAC Indireto","Ideal")
xlabel("Tempo [s]")
ylabel("Parametro")
title("Experiencia 9 - Parametro b da planta " + titulo)
if saveFig
    saveas(gcf,'Exp9_'+string(entrada)'+'_ParametroB.png')
end
```

13. Experiência 10 – Controle Adaptativo

Analisando os resultados obtidos nas Experiências 7, 8 e 9, responda:

(a) Qual a diferença conceitual entre os controladores adaptativos e os métodos de identificação paramétrica estudados?

(b) Qual o objetivo do "Controlador Adaptativo por Modelo de Referência"?

(c) Qual a diferença conceitual entre o MRAC Direto e o MRAC Indireto?

(d) Os controladores adaptativos estudados podem ser aplicados para estimar dinâmicas com estruturas desconhecidas?

(e) Os métodos implementados são recursivos ou não-recursivos?

(f) Em todos os casos trabalhados, os parâmetros estimados do controlador alcançam os mesmos valores? Comente.

(g) Qual a metodologia aplicada para encontrar a lei de controle u(t) e as leis de adaptação dos parâmetros?

(h) A velocidade de estabilização do sistema controlado é regulada por quais variáveis? Elas são calculadas ou definidas pelo projetista? Comente sobre a atuação dessas variáveis.

(i) As variáveis definidas pelo projetista na implementação dos controladores propostos estão sujeitas a alguma restrição para manter a estabilidade do sistema controlado em malha fechada?

13.1. Resolução

13.1.1. a)

Enquanto os métodos de identificação paramétrica estão preocupados em descobrir os parâmetros da planta analisada, os controladores adaptativos inserem no sistema uma realimentação negativa de controle adaptativa, juntamente com um modelo de referência, com o objetivo de gerar uma saída semelhante ou até igual ao seu modelo proposto, adaptando diretamente os valores de ganho deste controlador, sem necessariamente saber os parâmetros da planta para isso.

13.1.2. b)

O principal objetivo do Controlador Adaptativo por Modelo de Referência (MRAC) é estabelecer um modelo de referência desejado e garantir que a saída da planta seja coincidente com a saída do modelo de referência, independente da entrada do sistema.

13.1.3. c)

O MRAC Direto pode ser caracterizado como a classe em que os parâmetros do controlador são atualizados diretamente, sem a necessidade de cálculos intermediários. Já no MRAC Indireto, os parâmetros do controlador não são atualizados diretamente, mas são recalculados a cada instante de tempo usando as estimativas dos parâmetros do sistema.

13.1.4. d)

Não, pois os controladores implementados necessitam de um modelo referência, esse modelo deve seguir uma dinâmica na qual todas as deduções possam se concentrar na quantidade de polos da planta. Logo modelos com dinâmicas desconhecidas não se enquadram nesse tipo de controle.

13.1.5. e)

Os métodos implementados são recursivos, afinal atualizam os parâmetros a cada instante de tempo, utilizando valores anteriores e atuais dos sinais medidos

13.1.6. f)

Em alguns casos os parâmetros estimados alcançaram o mesmo valor e em outros não. O modelo MRAC não garante que os parâmetros atinjam os valores ideias, ele só garante que o sistema siga o modelo de referência.

13.1.7. g)

Para a lei de controle do sistema, foi adotada uma metodologia na qual eram definidas constantes que agem tanto no estado controlado da planta quanto na entrada do sistema. Essas constantes eram os parâmetros a serem estimados. Já para a lei de controle foi realizado uma metodologia aplicando uma função candidata de Lyapunov, a qual era dependente tanto do sinal de erro de acompanhamento do sistema, quanto os parâmetros a serem estimados. Assim, para que a função fosse assintoticamente estável era garantido que o ponto de equilíbrio fosse zero, pelo teorema de LaSalle.

13.1.8. h)

A velocidade de estabilização é regulada majoritariamente pelas variáveis de adaptação γ presentes no projeto. Elas são definidas pelo projetista com o intuito de gerar a melhor convergência possível para a sua aplicação. Caso elas sejam muito pequenas, a convergência dos parâmetros também é lenta e, por outro lado, caso elas sejam grandes demais, o sistema pode se tornar muito oscilatório ou muito brusco. Logo, essa escolha deve ser feita com certos cuidados e atenção pelo projetista. Outras variáveis que também afetam a velocidade de convergência, mas em menor escala do que as citadas anteriormente, são as condições iniciais dos parâmetros da planta. Caso elas sejam muito destoantes da planta original, pode levar um tempo maior para que o algoritmo de identificação encontre os parâmetros ideais.

13.1.9. i)

Os parâmetros definidos pelo projetista (Passo de adaptação e condições iniciais) não estão sujeitos a restrições para manter a estabilidade. No processo de construção das leis de adaptação, mais especificamente nas funções de Liapunov, são definidos valores γ qualquer para propor a estabilidade do sistema, tanto que os termos que envolvem γ são anulados pela proposição das leis de adaptação.

13.2. Script

Por fim será apresentado o script construído para obter os valores desejados com todos os métodos apresentados neste capítulo. Inicialmente o código recebe do usuário qual conjunto de dados e método desejado e logo após, calcula o os parâmetros de acordo com o método escolhido.

14. Exercício de Sala

Considere o seguinte conjunto de dados:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
u(t)	1	0.8	0.6	0.4	0.2	0	0.2	0.4	0.6	0.8	1	0.8	0.6	0.4	0.2
y(t)	0.9	2.5	2.4	1.3	1.2	0.8	0	0.9	1.4	1.9	2.3	2.4	2.3	1.3	1.2

Obter a função de transferência estimada para um processo representado pelo seguinte modelo matemático:

$$y(t) = b_0 u(t) + b_1 u(t-1) \tag{127}$$

Utilizando os seguintes procedimentos:

(a) Substituição direta;

(b) Método do gradiente;

(c) Método dos mínimos quadrados puro não-recursivo;

(d) Método dos mínimos quadrados puro recursivo;

(e) Método dos mínimos quadrados com fator de esquecimento não-recursivo;

(f) Método dos mínimos quadrados com fator de esquecimento recursivo;

Para cada método utilize:

(i) Apenas o conjunto de dados N = 7, 8, . . . ,14

(ii) Todo o conjunto de dados

14.1. Resolução:

14.1.1. a)

O primeiro método a ser realizado foi através da substituição direta. De forma geral, a estrutura do sistema é apresentado a seguir:

$$y(i) = b_0 u(i) + b_1 u(i-1)$$

$$y\left(i+1\right)=b_0u\left(i+1\right)+b_1u\left(i\right)$$

Manipulando as equações e escrevendo na forma matricial, teremos:

$$\begin{pmatrix} u\left(i\right) & u\left(i-1\right) \\ u\left(i+1\right) & u\left(i\right) \end{pmatrix}^{-1} \cdot \begin{pmatrix} y\left(i\right) \\ y\left(i+1\right) \end{pmatrix} = \begin{pmatrix} b_0\left(i+1\right) \\ b_1\left(i+1\right) \end{pmatrix}$$

Dessa forma, o algoritmo foi implementado no MATLAB e possibilitou a obtenção dos valores de b_0 e b_1 para os casos (i) e (ii). Os dados referentes a ambos os casos constam, respectivamente, nas Tabelas 1 e 2.

Tabela 1: Ganhos para o caso (i) com o método de substituição direta

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
b_0	0.0	-10.0	10.0	-5.0	-2.0	0.0	0.0	2.0	2.0	3.5	1.1	-9.5	8.5	-5.0
b_1	0.0	10.5	-4.5	5.5	4.0	4.0	4.5	0.5	0.5	-1.5	1.6	10.0	-3.5	5.5

Tabela 2: Ganhos para o caso (ii) com o método de substituição direta

	1	**2**	**3**	**4**	**5**	**6**	**7**
b_0	0.0	2.0	3.5	1.1	-9.5	8.5	-5.0
b_1	0.0	0.5	-1.5	1.6	10.0	-3.5	5.5

14.1.2. b)

Para o método do gradiente apresentado anteriormente, utilizando o formato da Equação 1, as seguintes variáveis são definidas:

$$z = y(t)$$

$$\theta = \begin{pmatrix} b_0 & b_1 \end{pmatrix}^T$$

$$\phi = \Big(u(t), u(t-1)\Big)^T$$

Logo, as Tabelas 3 e 4 mostram os resultados obtidos pelo método para o conjunto N de dados (i) e para o conjunto completo (ii).

14.1.3. c)

O terceiro método utilizado é o MMQ puro não recursivo. O MMQ consiste em uma função custo que minimiza quadrado de todos os erros resultantes das diferenças de

Tabela 3: Ganhos para o caso (i) com o método do gradiente

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
b_0	1	1.21	1.41	1.36	1.42	1.42	1.37	1.37	1.34	1.27	1.18	1.16	1.28	1.23
b_1	1	1.26	1.53	1.46	1.57	1.67	1.67	1.67	1.64	1.60	1.52	1.50	1.66	1.58

Tabela 4: Ganhos para o caso (ii) com o método do gradiente

	1	**2**	**3**	**4**	**5**	**6**	**7**
b_0	1	1.16	1.28	1.30	1.35	1.49	1.44
b_1	1	1.11	1.20	1.22	1.27	1.46	1.39

estimativas.

Para o método não recursivo, sem fator de esquecimento, primeiramente foi calculada a matriz de covariância $P(t)$, da forma como apresentado na Equação 128.

$$P(t) = \left[Q_0 + \int_0^t \frac{\phi\ (\tau\ )\, \phi^{\,T}\, (\tau\ )}{m_s^2\, (\tau\ )}\, d\tau \right] \tag{128}$$

onde Q_0 é uma matriz simétrica definida como:

$$Q_0 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

Com isso, a matriz dos parâmetros desconhecidos θ é dada pela Equação 129.

$$\theta\,(t) = P\,(t) \left[Q_0\theta_0 + \int_0^t \frac{z\,(\tau\ )\, \phi\,(\tau\ )}{m_s^2\,(\tau\ )}\, d\tau \right] \tag{129}$$

As Tabelas 5 e 6 mostram os resultados desse método para o conjunto N completo e pela metade.

Tabela 5: Ganhos para o caso (i) com o método do MMQ puro não recursivo

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
b_0	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
b_1	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68

Tabela 6: Ganhos para o caso (ii) com o método do MMQ puro não recursivo

	1	**2**	**3**	**4**	**5**	**6**	**7**
b_0	0.74	0.74	0.74	0.74	0.74	0.74	0.74
b_1	0.74	0.74	0.74	0.74	0.74	0.74	0.74

É possível notar que, como o método não é recursivo, em todos os casos são obtidos os mesmos valores de b_0 e b_1.

14.1.4. d)

Agora é utilizado o método dos Mínimos Quadrados Puro Recursivo, descrito em detalhes no Algoritmo 1 e pelas Equações 19 até a 22. Os resultados para o conjunto N de dados (i) e para o conjunto completo (ii) são apresentados pelas Tabelas 7 e 8.

Tabela 7: Ganhos para o caso (i) com o método do MMQ puro

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
b_0	1.06	1.14	1.15	1.18	1.16	1.12	1.15	1.18	1.19	1.19	1.19	1.21	1.21	1.21
b_1	1.08	1.19	1.21	1.28	1.34	1.35	1.36	1.37	1.37	1.37	1.38	1.41	1.41	1.45

Tabela 8: Ganhos para o caso (ii) com o método do MMQ puro

	1	**2**	**3**	**4**	**5**	**6**	**7**
b_0	1.08	1.13	1.15	1.17	1.20	1.20	1.21
b_1	1.05	1.09	1.11	1.14	1.21	1.22	1.29

14.1.5. e)

A próxima estratégia utilizada foi o método dos mínimos quadrados com fator de esquecimento não-recursivo. Para implementar, foi utilizado um raciocínio com aquele apresentado no item c), porém com uma constante β para representar um fator de esquecimento no sistema. Assim as matrizes $P(t)$ e $\theta(t)$ possuem termos adicionais em relação as equações 128 e 129 que incluem a constante β. Neste caso iremos definir $P(t)$ conforme consta na Equação 130:

$$P\left(t\right)=\left[e^{-\beta t}Q_0+\int_0^t e^{-\beta(t-\tau)}\frac{\phi\left(\tau\right)\phi^T\left(\tau\right)}{m_s^2\left(\tau\right)}\,d\tau\right]^{-1} \tag{130}$$

Q_0 se mantém sendo uma matriz simétrica definida como:

$$Q_0=\begin{pmatrix}1 & 2\\ 2 & 1\end{pmatrix}$$

E a matriz dos parâmetros desconhecidos θ é definida pela Equação 131

$$\theta\left(t\right)=P\left(t\right)\left[e^{-\beta t}Q_0\theta_0+\int_0^t e^{-\beta(t-\tau)}\frac{z\left(\tau\right)\phi\ \left(\tau\right)}{m_s^2\left(\tau\right)}d\tau\right] \tag{131}$$

Após implementação do algoritmo, os resultados obtidos para o conjunto N de dados (i) e para o conjunto completo (ii) são apresentados pelas Tabelas 9 e 10:

Tabela 9: Ganhos para o caso (i) com o MMQ com fator de esquecimento não-recursivo

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
b_0	0.04	0.04	0.05	0.05	0.06	0.07	0.09	0.11	0.14	0.18	0.23	0.30	0.39	0.51
b_1	0.04	0.04	0.05	0.05	0.06	0.07	0.09	0.11	0.14	0.18	0.23	0.30	0.39	0.51

Tabela 10: Ganhos para o caso (ii) com o MMQ com fator de esquecimento não-recursivo

	1	**2**	**3**	**4**	**5**	**6**	**7**
b_0	0.27	0.30	0.33	0.37	0.42	0.50	0.60
b_1	0.27	0.30	0.33	0.37	0.42	0.50	0.60

Nota-se que, diferente do método apresentado na alternativa c), que não possuía fator de esquecimento, este altera os valores de b_0 e b_1 com o passar do tempo.

14.1.6. f)

Para realizar a implementação do método dos mínimos quadrados com fator de esquecimento e recursivo, foi utilizado um algoritmo semelhante ao do item d.

No entanto esse método conta com uma constante β, que é usada no sistema como o fator de esquecimento. Logo o algoritmo implementado para solucionar o problema de identificação dos parâmetros b_0 e b_1, é apresentado a seguir.

Nas Tabela 11 e 12, é possível observar os valores encontrados para esses parâmetros.

Tabela 11: Ganhos para o caso (i) com o método do MMQ com fator de esquecimento

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
b_0	1	1.08	1.23	1.23	1.29	1.21	1.04	1.15	1.16	1.13	1.09	1.10	1.05	1.12
b_1	1	1.10	1.29	1.30	1.47	1.68	1.73	1.74	1.74	1.73	1.69	1.62	1.81	1.64

Tabela 12: Ganhos para o caso (ii) com o método do gradiente

	1	**2**	**3**	**4**	**5**	**6**	**7**
b_0	1.11	1.19	1.22	1.25	1.32	1.31	1.30
b_1	1.07	1.13	1.16	1.21	1.38	1.35	1.58

```

% Exercicio 2 - Identificacao de parametros
```

Algoritmo 4: MMQ com Fator de Esquecimento Recursivo

Require: t, z, ϕ
Ensure: $\hat{\theta}$
  `Definir` P_0
  `Definir` θ_0
  `Definir` β
  for $i \leftarrow 1, \ldots,$ `length`(t) **do**
    $m_s \leftarrow 1 + \phi(i)^T\phi(i)$
    $\dot{P} = \beta P_0 - (P_0\phi(i)\phi(i)^T P_0)/m_s^2$
    $P = \dot{P}.dt + P_0$
    $\hat{z}(i) = \theta_0\phi(i)$
    $\epsilon = (z(i) - \hat{z}(i))/m_s^2$
    $\dot{\theta} = P\epsilon\phi(i)$
    $\theta = \dot{\theta}.dt + \theta_0$
    $P_0 = P$
    $\theta_0 = \theta$
  end for

```

clc;
clear;
close all;

%% Interface

disp("Entrada 1: Vetor inteiro");
disp("Entrada 2: Metade do vetor")

entrada = input("Qual voce deseja? ");
clc;

if entrada==1
    half= false;
else
    half= true;
end

disp("Entrada 1: Substituicao direta");
disp("Entrada 2: Metodo do gradiente");
disp("Entrada 3: MMQ puro nao recursivo");
disp("Entrada 4: MMQ puro recursivo");
disp("Entrada 5: MMQ nao recursivo com fator de esquecimento");
```

```
disp("Entrada 6: MMQ recursivo com fator de esquecimento");

metodo = input("Qual voce deseja? ");

%% Definicao dos vetores

dt = 1;
t = 0:dt:14;
u = [1 0.8 0.6 0.4 0.2 0 0.2 0.4 0.6 0.8 1 0.8 0.6 0.4 0.2];
y = [0.9 2.5 2.4 1.3 1.2 0.8 0 0.9 1.4 1.9 2.3 2.4 2.3 1.3 1.2];

if half
    u = u(8:end);
    y = y(8:end);
    t = t(8:end);
end

%% a) Substituicao direta

if metodo==1

theta = zeros(2,length(t)-1);

for i=2:length(t)-1
    M = [u(i), u(i-1);
         u(i+1), u(i)];
    y_i = [y(i); y(i+1)];

    theta(:, i) = M^(-1)*y_i;
end

end

%% b) Metodo do gradiente
if metodo==2

% Vetores dos parametros
b0 = zeros(1, length(t)-1);
b1 = zeros(1, length(t)-1);
% Valores iniciais
b0(1) = 1;
b1(1) = 1;

```

```
theta = [b0; b1]';
phi = [u(2:end); u(1:(end-1))];

z = y(2:end);

gamma = 1;

z_hat = zeros(1, length(z));
e = zeros(1, length(z));

for i=1:length(z)-1
    z_hat(i) = theta(i,:)*phi(:,i);
    ms_2 = 1 + phi(:,i)'*phi(:,i);
    e(i) = (z(i) - z_hat(i))/ms_2;
    theta_dot = gamma*e(i)*phi(:,i)';
    theta(i+1,:) = theta_dot*dt + theta(i,:);
end

end
%% c) MMQ puro nao recursivo
if metodo==3

% Vetores dos parametros
b0 = zeros(1, length(t)-1);
b1 = zeros(1, length(t)-1);
% Valores iniciais
b0(1) = 1;
b1(1) = 1;

%Parametros iniciais
beta= 0;
Q0= [1 2; 2 1];
theta_0= [b0(1); b1(1)];

theta = zeros(2,length(t)-1);
phi = [u(2:end); u(1:(end-1))];

z = y(2:end);

%Inicializacao
P= zeros(2,2,length(t)-1);
dt= 1;

```

```
for t_i = 1:length(t)-1
    integral= 0;

    for tal = 1:length(t)-1
        phi_i = phi(:,tal); % Valor de phi pro instante tal
        ms = 1+phi_i'*phi_i; % Valor de ms para o instante tal

        derivada = exp(-beta*(t_i-tal)) * (phi_i'*phi_i)/ms^(2);
        integral= integral + derivada*dt;
    end

    P(:,:,t_i) = (exp(-beta*t_i)*Q0 + integral)^(-1);
end


for t_i = 1:length(t)-1
    integral_1= 0;

    for tal = 1:length(t)-1
        phi_i = phi(:,tal); % Valor de phi pro instante tal
        z_i = z(tal); % Valor de z pro instante tal
        ms = 1+phi_i'*phi_i; % Valor de ms para o instante tal

        derivada_1 = exp(-beta*(t_i-tal)) * (z_i*phi_i)/ms^(2);
        integral_1= integral_1 + derivada_1*dt;
    end

    theta(:,t_i) = P(:,:,t_i)*(exp(-beta*t_i)*Q0*theta_0 + integral);
end

end


%% d) MMQ puro recursivo

if metodo==4

% Vetores dos parametros
b0 = zeros(1, length(t)-1);
b1 = zeros(1, length(t)-1);
% Valores iniciais
b0(1) = 1;
b1(1) = 1;
```

```
theta = [b0; b1]';
phi = [u(2:end); u(1:(end-1))];

z = y(2:end);

%Valores iniciais de theta
theta_0= [b0(1); b1(1)];

% Matriz inicial P
P0 = [1 0 ;
      0 1];

% Progressao de theta
theta_v = [theta_0];
P_m= [];

for i=1:length(t)-1
    phi_i = phi(:,i); % Valor de phi pro instante i

    ms = 1+phi_i'*phi_i; % Valor de ms para o instante i

    P_d = -P0*phi_i*phi_i'*P0/(ms^2); % Calculo da derivada de P
    P = P0 + P_d*dt; % Atualizacao de P

    z_i = z(i);  % Valor de z para o instante i
    z_hat = theta_0'*phi_i; % Estimativa de z com os parametros atuais

    e = (z_i - z_hat)/ms^2; % Calculo de erro

    theta_d = P*e*phi_i;  % Calculo da derivada de theta
    theta = theta_0 + theta_d*dt; % Atualizacao de theta

    theta_v(:, end+1) = theta; % Armazenamento da progressao de theta_v

    % Valores para a proxima iteracao
    P0 = P;
    theta_0 = theta;
end
theta= theta_v;
end

%% e) MMQ nao recursivo com fator de esquecimento
```

```
if metodo==5

% Vetores dos parametros
b0 = zeros(1, length(t)-1);
b1 = zeros(1, length(t)-1);
% Valores iniciais
b0(1) = 1;
b1(1) = 1;

%Parametros iniciais
beta= 0.3;
Q0= [1 2; 2 1];
theta_0= [b0(1); b1(1)];

theta = zeros(2,length(t)-1);
phi = [u(2:end); u(1:(end-1))];

z = y(2:end);

%Inicializacao
P= zeros(2,2,length(t)-1);
dt= 1;

for t_i = 1:length(t)-1
    integral= 0;

    for tal = 1:length(t)-1
        phi_i = phi(:,tal); % Valor de phi pro instante tal
        ms = 1+phi_i'*phi_i; % Valor de ms para o instante tal

        derivada = exp(-beta*(t_i-tal)) * (phi_i'*phi_i)/ms^(2);
        integral= integral + derivada*dt;
    end

    P(:,:,t_i) = (exp(-beta*t_i)*Q0 + integral)^(-1);
end


for t_i = 1:length(t)-1
    integral_1= 0;

    for tal = 1:length(t)-1
        phi_i = phi(:,tal); % Valor de phi pro instante tal
```

```
        z_i = z(tal); % Valor de z pro instante tal
        ms = 1+phi_i'*phi_i; % Valor de ms para o instante tal

        derivada_1 = exp(-beta*(t_i-tal)) * (z_i*phi_i)/ms^(2);
        integral_1= integral_1 + derivada_1*dt;
    end

    theta(:,t_i) = P(:,:,t_i)*(exp(-beta*t_i)*Q0*theta_0 + integral);
end

end

%% f) MMQ recursivo com fator de esquecimento

if metodo==6

% Vetores dos parametros
b0 = zeros(1, length(t)-1);
b1 = zeros(1, length(t)-1);
% Valores iniciais
b0(1) = 1;
b1(1) = 1;
beta= 0.3;

theta = [b0; b1]';
phi = [u(2:end); u(1:(end-1))];

z = y(2:end);

%Valores iniciais de theta
theta_0= [b0(1); b1(1)];

% Matriz inicial P
P0 = [1 0 ;
      0 1];

% Progressao de theta
theta_v = [theta_0];
P_m= [];

for i=1:length(t)-1
    phi_i = phi(:,i); % Valor de phi pro instante i

```

```
    ms = 1+phi_i'*phi_i; % Valor de ms para o instante i

    P_d = beta*P0 - P0*phi_i*phi_i'*P0/(ms^2); % Calculo da derivada de
       P
    P = P0 + P_d*dt; % Atualizacao de P

    z_i = z(i);  % Valor de z para o instante i
    z_hat = theta_0'*phi_i; % Estimativa de z com os parametros atuais

    e = (z_i - z_hat)/ms^2; % Calculo de erro

    theta_d = P*e*phi_i;  % Calculo da derivada de theta
    theta = theta_0 + theta_d*dt; % Atualizacao de theta

    theta_v(:, end+1) = theta; % Armazenamento da progressao de theta_v

    % Valores para a proxima iteracao
    P0 = P;
    theta_0 = theta;
end
theta= theta_v;
end
disp("Os parametros encontrados sao:")
disp(theta)
```