

## Exemplo

Considere o seguinte modelo para um sistema massa-mola-amortecedor:

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -9 & -2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_B u(t),$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

em que  $x_1$  denota a deformação da mola em metro e  $x_2$  a taxa de deformação da mola em metro por segundo. Considerando que seja possível medir todos os estados do sistema, projetou-se um sistema de regulação empregando o LQR (Regulador Linear Quadrático) com os parâmetros

$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \text{ e } R = 1.$$

Para isso, considerou-se a lei de controle

$$u(t) = -Kx(t)$$

e o problema de otimização

$$\begin{array}{ll} \text{minimizar} & J = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)] dt \\ \text{sujeito a} & \dot{x}(t) = Ax(t) + Bu(t) \end{array}$$

cuja solução é obtida resolvendo-se a Equação Algébrica de Riccati

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

e fazendo  $K = R^{-1}B^T P$ . Utilizando o comando *lqr* do MATLAB®, obteve-se  $K = [4, 4536 \quad 1, 7292]$ .

### Simulação

A simulação foi realizada no MATLAB® (Apêndice) através da discretização exata do sistema (*lsim*) com passo  $\delta t = 0,03$  s.

Figura 1: Representação animada do controle de um sistema massa mola amortecedor utilizando o LQR

Na Figura 1, existem 203 imagens sobrepostas, nomeadas sequencialmente como (*simulacao0.pdf*), (*simulacao1.png*), . . . , (*simulacao201.png*) e (*simulacao202.pdf*). Vale ressaltar que para reduzir o tamanho do arquivo PDF, a primeira e última figura encontram-se no formato **.pdf** e as demais no formato **.png**. Além disso, é possível interagir com a animação através dos comandos

- [◀] Recuo total para o início
- [◀] Recuo manual
- [◀] Recuo automático
- [▶] Avanço automático
- [▶] Avanço manual
- [▶] Avanço total para o final
- [+] Aumenta a velocidade de transição do modo automático
- [\*\*] Velocidade padrão de transição do modo automático
- [−] Diminui a velocidade de transição do modo automático

Finalmente, um exemplo de código para ser utilizado na inserção destas animações no L<sup>A</sup>T<sub>E</sub>X:

```

\documentclass{article}
\usepackage[a4paper, left=2cm, right=2cm, top=2cm, bottom=3cm]{geometry}
\usepackage[utf8]{inputenc}
\usepackage[portuguese]{babel}
\usepackage{graphicx}
\usepackage{animate}
\usepackage{float}

%-----
% Neste exemplo, há 203 figuras, localizadas na pasta "Exemplo" e com o nome de
% "simulacao". A primeira está nomeada como "simulacao0" e a última como
% "simulacao202".
%-----

\newcommand{\iTotal}{203}
\newcommand{\sEndereco}{Exemplo/simulacao}
\newcommand{\iInicial}{0}
\newcommand{\iFinal}{202}

\begin{document}

\begin{figure}[H]
\centering
\animategraphics[controls, width=\textwidth{}]{\iTotal}{\sEndereco}{\iInicial}{\iFinal}
\caption{Representação animada}
\end{figure}

\end{document}

```

## Apêndice

Destaca-se o código MATLAB® que foi utilizado para o desenvolvimento deste exemplo.

```

A = [0 1; -9 -2]; n = length(A);
B = [0; 1]; p = size(B,2);
C = [1 0;0 1]; q = size(C,1);
D = 0;
Q = diag([100,1]);
R = 1;
K = lqr(A,B,Q,R);

Planta = ss(A,B,C,0, 'InputName','u', 'OutputName',{'x1','x2'});
Controlador = tf(-K, 'InputName',{'x1','x2'}, 'OutputName','u');
Gmf = connect(Planta,Controlador,'u',{'x1','x2'});

tFin=6;
deltaT=0.03;
t = 0:deltaT:tFin;
x0 = [-10;0];

% Simulacao em malha aberta
uMA = zeros(1,length(t));
xMA = lsim(Planta,uMA,t,x0); xMA = xMA';

% Simulacao em malha fechada
uMF = zeros(1,length(t));
xMF = lsim(Gmf,uMF,t,x0); xMF = xMF';

```

```

for i=1:length(t)
    uMF(i) = -K*xMF(:,i);
end

for it = 1:length(t)
    plotResultados(1,t,xMF,xMA,uMF,uMA,it)
    set(gcf, 'Color', 'w')
    export_fig(['simulacao',num2str(it)],'-png','-q101','-nocrop')
end

export_fig('simulacao0','-pdf','-q101','-nocrop')
export_fig(['simulacao',num2str(it+1)],'-pdf','-q101','-nocrop')

```

E também a função que foi utilizada para gerar cada uma das 203 figuras.

```

function plotResultados(figura,t,xMF,xMA,uMF,uMA,it)

FIGURA = figure(figura);
FIGURA.Units = 'centimeters'; FIGURA.Position = [0 0 20 16];

subplot(2,2,1)
    MASSAMF = [3 3 -3 -3 3; 0 1 1 0 0];
    MASSAMA = [3 3 -3 -3 3; 0 1 1 0 0];
    MOLAMF(1,:) = linspace(-15,xMF(1,it)-3,10);
    MOLAMF(2,:) = 0.7+0.1.*[1 -1 1 -1 1 -1 1 -1];
    MOLAMA(1,:) = linspace(-15,xMA(1,it)-3,10);
    MOLAMA(2,:) = 0.7+0.1.*[1 -1 1 -1 1 -1 1 -1];
    aux = linspace(-15,xMF(1,it)-3,4);
    AMORTMF(1,:) = [aux(1) aux(2) aux(2) aux(3) aux(3) aux(2) aux(2) ...
                    aux(3) aux(3) aux(4)];
    AMORTMF(2,:) = 0.3+0.1.*[0 0 1 1 -1 -1 1 1 0 0];
    aux = linspace(-15,xMA(1,it)-3,4);
    AMORTMA(1,:) = [aux(1) aux(2) aux(2) aux(3) aux(3) aux(2) aux(2) ...
                    aux(3) aux(3) aux(4)];
    AMORTMA(2,:) = 0.3+0.1.*[0 0 1 1 -1 -1 1 1 0 0];
    plot([0 0],[-0.5 1.5], 'color',0.7.*[1 1 1], 'LineWidth',2); hold on;
    plot([-15 7],[0 0], 'color',0.7.*[1 1 1], 'LineWidth',2);
    plot(MASSAMF(1,:)+xMF(1,it),MASSAMF(2,:),'blue','LineWidth',2);
    plot(MOLAMF(1,:),MOLAMF(2,:),'blue','LineWidth',2);
    plot(AMORTMF(1,:),AMORTMF(2,:),'blue','LineWidth',2);
    plot(MASSAMA(1,:)+xMA(1,it),MASSAMA(2,:),'--red','LineWidth',1);
    plot(MOLAMA(1,:),MOLAMA(2,:),'--red','LineWidth',1);
    plot(AMORTMA(1,:),AMORTMA(2,:),'--red','LineWidth',1); hold off;
    AX=gca; AX.XLim = [-15 7]; AX.YLim = [-0.5 1.5]; AX.YTickLabel = [];
    AX.XTick = [-10 0];
    xlabel('Posição (metro)', 'Interpreter', 'latex');
    AX = gca;
    text(max(AX.XLim), max(AX.YLim), '(a)', ...
          'HorizontalAlignment', 'right', ...
          'VerticalAlignment', 'top', ...
          'Interpreter', 'latex');
    text(min(AX.XLim), max(AX.YLim), ...
          ['Tempo: ', strrep(num2str(round(t(it),2), '%.2f'), '.', ','), ' s'], ...
          'HorizontalAlignment', 'left', ...
          'VerticalAlignment', 'top', ...
          'Interpreter', 'latex');

subplot(2,2,2)

```

```

plot([0 t(end)],[0 0], 'color', 0.7.*[1 1 1], 'LineWidth', 2); hold on;
MF = plot(t(1:it),xMF(1,1:it), 'blue', 'LineWidth', 2);
MA = plot(t(1:it),xMA(1,1:it), '--red', 'LineWidth', 2); hold off;
AX=gca; AX.YLim = [min(min(xMF(1,:))),min(xMA(1,:)))-3, ...
    max(max(xMF(1,:)),max(xMA(1,:)))+3];
xlabel('Tempo (segundo)', 'Interpreter', 'Latex');
ylabel('Posi\c{c}\~ao (metro)', 'Interpreter', 'Latex');
LEGENDA = legend([MF,MA], 'Malha fechada', 'Malha aberta');
LEGENDA.Interpreter = 'Latex';
LEGENDA.Position = [0.4335      0.4779      0.1722      0.0604];
AX = gca;
text(max(AX.XLim), max(AX.YLim), '(b)', ...
    'HorizontalAlignment', 'right', ...
    'VerticalAlignment', 'top', ...
    'Interpreter', 'latex');

subplot(2,2,3)
plot([0 t(end)],[0 0], 'color', 0.7.*[1 1 1], 'LineWidth', 2); hold on;
MF = plot(t(1:it),uMF(1:it), 'blue', 'LineWidth', 2);
MA = plot(t(1:it),uMA(1:it), '--red', 'LineWidth', 2); hold off;
AX=gca; AX.YLim = [min(min(uMF)),min(uMA))-3, ...
    max(max(uMF),max(uMA))+3];
xlabel('Tempo (segundo)', 'Interpreter', 'Latex');
ylabel('Controle (newton)', 'Interpreter', 'Latex');
AX = gca;
text(max(AX.XLim), max(AX.YLim), '(c)', ...
    'HorizontalAlignment', 'right', ...
    'VerticalAlignment', 'top', ...
    'Interpreter', 'latex');

subplot(2,2,4)
plot([0 t(end)],[0 0], 'color', 0.7.*[1 1 1], 'LineWidth', 2); hold on;
MF = plot(t(1:it),xMF(2,1:it), 'blue', 'LineWidth', 2);
MA = plot(t(1:it),xMA(2,1:it), '--red', 'LineWidth', 2); hold off;
AX=gca; AX.YLim = [min(min(xMF(2,:))),min(xMA(2,:)))-3, ...
    max(max(xMF(2,:)),max(xMA(2,:)))+3];
xlabel('Tempo (segundo)', 'Interpreter', 'Latex');
ylabel('Velocidade (metro/segundo)', 'Interpreter', 'Latex');
AX = gca;
text(max(AX.XLim), max(AX.YLim), '(d)', ...
    'HorizontalAlignment', 'right', ...
    'VerticalAlignment', 'top', ...
    'Interpreter', 'latex');
end

```

Vale ressaltar que tambem é necessário instalar o programa **export\_fig**, disponível em

[https://www.mathworks.com/matlabcentral/fileexchange/23629-export\\_fig](https://www.mathworks.com/matlabcentral/fileexchange/23629-export_fig),

o programa **Postscript and PDF interpreter/renderer**, disponível em

<https://ghostscript.com/releases/index.html>

e o programa **Xpdf command line tools**, disponível em

<https://www.xpdfreader.com/download.html>