

Códigos do Mini-Curso de MATLAB® & SIMULINK®

Reginaldo Cardoso

October 17, 2016

1 Parte I (até a EDO)

```
% Comandos usados no Mini_Curso de MatLAB
%%
%===== SLIDE 10 =====
x = 5
abs(x)          %Valor absoluto de x
acos(x)         %Arco cosseno de x (rad)
asin(x)         %Arco seno de x (rad)
atan(x)         %Arco tangente de x (rad)
cos(x)          %Cosseno de x (rad)
sin(x)          %Seno de x (rad)
tan(x)          %Tangente de x (rad)
exp(x)          %Exponencial (e^x)
log(x)          %Logaritmo natural (base e)
log10(x)        %Logaritmo na base 10
sqrt(x)         %Raiz quadrada
factorial(x)    %Fatorial de x (x!)
%%
%===== SLIDE 12 & 13 =====
distância = 100    %Exemplo de variavel com acento
Distancia = 20     % diferenca maiusculo/minusculo (case sensitive)
distancia = 100    % diferenca maiusculo/minusculo (case sensitive)
tempo = 3          %declando uma variavel
% calculo de velocidade media
velocidade_media = distancia / tempo
% Para suprimir a exibicao da variavel adicionamos um ponto-e-virgula ao
% final do comando.
%%
%===== SLIDE 14 =====
% Formatacao de dados numericos
n = 12.345678901234567
```

```

format short      %4 digitos decimais (formato padrao) 12.3457
n_short = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format long       %14 digitos decimais 12.345678901234567
n_long = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short e    %5 digitos mais expoente 1.2346e+001
n_short_e = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short g    %5 digitos no total com ou sem expoente 12.346
n_short_g = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format long e     %15 digitos mais expoente 1.234567890123457e+001
n_long_e = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format long g     %15 digitos no total com ou sem expoente 12.3456789012346
n_long_g = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format bank       %Formato monetario 12.35
n_bank = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format hex        %Exibicao hexadecimal de bits 4028b0fcd32f707a
n_hex = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format rat        %Razao aproximada entre interios pequenos 1000/81
n_rat = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format compact    %Elimina espacos (+informacao mostrada na tela)
n_compact = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format loose      %Adiciona espacos entre linhas
n_loose = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format +          %Exibe somente o sinal do numero
n_mais = n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%% PADRAO
format short
%%
% clc
%===== SLIDE 15 =====
% Ordem de Procedencia

```

```

exemplo_1 = 6/2*3
exemplo_2 = 12/2 + 3*(2^4)
%%
clc
clear tempo
%===== SLIDE 16 =====
% Variaveis predefinidas
n_pi = pi %3.141592653589793
maior_que_um = eps %Somado a 1, cria um numero maior do que 1
% NaN %Nao numero (not a number)
menor_real = realmin %menor numero real positivo
maior_real = realmax %maior numero real positivo
% inf %Infinito
infinito = realmax/realmin
%%
clear
clc
%===== SLIDE 18 =====
% Declarando Variaveis: Vetor
[1 2 3] %Vetor linha (covetor)
[1,2,3] %Vetor linha (covetor) com virgula
[1;2;3] %Vetor coluna
%===== SLIDE 19 =====
A = 1:1:5 %Outra forma de criar um conjunto
X = linspace(0,pi,4) %Gera um vetor linearmente espacado
V = logspace(0,2,5) %Gera um vetor logaritmicamente espacado
%===== SLIDE 20 =====
B=2*A;
Y=sin(X);
% Qual sera a resposta?
Z_1 =A.^2;
Z_2 =A^2;
%%
clear
clc
%===== SLIDE 22 =====
% Declarando Variaveis: Matriz
M = [1,0,-1;2,3,4;-7,1,3];
M = [1 0 -1; 2 3 4; -7 1 3];
% Identifica o elemento da Segunda linha e Terceira coluna.
M(2,3)
M(8)

```

```

##### Erro
M(10)
%===== SLIDE 24 =====
% Mx foi gerada concatenando-se o vetor b transposto e as colunas
% 2 e 3 da matriz M.
b=[2,-3,1];
Mx=[b',M(:,2:3)]
%===== SLIDE 25 =====
% Algumas matrizes Predefinidas e funcoes
% size(Mx)      Retorna o numero de linhas e de colunas de Mx
% length(Mx)    A maior dimensao da matriz Mx
% inv(Mx)       Calcula a matriz inversa de Mx
% zeros(n,m)    Matriz de zeros com n linhas e m colunas
% eye(n,m)      Matriz identidade com n linhas e m colunas
% ones(n,m)     Matriz com 1, com n linhas e m colunas
% det(Mx)       Calcula o determinante da matriz Mx
#####
% Qual sera a resposta?
inversa_1 = inv([2,-3,1]);
inversa_2 = inv(eye(size(Mx)));
tamano = length(Mx);
identi_2 = eye(2);
zeros_3 = zeros(3);
um_3 = ones(3);
[n,m] = size(Mx);
%===== SLIDE 27 =====
% Algumas funcoes
A1 = [2 0 -1;-3 3 4;1 1 3];
b1 = [9;8;7];
A2 = [2 0 -1;3 3 4];
b2 = [9;8];
x_com_inv = inv(A)*b;
x_com_barra = A\b;
% Pseudo-inversa
x_pseudo = A2'*inv(A2*A2')*b2;
% Algumas funcoes
% who Exibe uma lista de variaveis declaradas/ativas na memoria
% whos Exibe uma lista de variaveis declaradas na memoria, com o
% respectivo tamanho em bytes e a classe de armazenamento
%%
clc
clear

```

```

%===== SLIDE 29 =====
% GRAFICOS
x = 0:1:10;
y = x.^2-10.*x+15;
plot(x,y);
%%
%===== SLIDE 31 =====
grid;
%===== SLIDE 35 =====
x=0:1:10;
y=cos(x);
z=sin(x);
plot(x,y,':rv',x,z,'-.mh');
grid;
%===== SLIDE 37 =====
plot(x,y,':rv',x,z,'-.mh','linewidth',3,'markersize',10,...
      'markeredgecolor','g','markerfacecolor','y');
grid;
%===== SLIDE 39 =====
plot(x,y,':rv',x,z,'-.mh','linewidth',3,'markersize',10,...
      'markeredgecolor','g','markerfacecolor','y');
grid;
xlabel('Eixo x');
ylabel('Eixo y');
title('Exemplo de Formatação')
text(x(6),y(6),'Cos(x)')
text(x(4),z(4),'Sen(x)')
axis([1,9,-0.8,0.8])
legend('Cosseno','Seno','location','northwestoutside')
%===== SLIDE 42 =====
% SUBPLOT
subplot(2,2,1)
t = 0:0.01:2*pi;
prim = sin(2*t).*cos(2*t);
polar(t,prim,'--r');grid
title('Polar')
subplot(2,2,2)
x = 0:0.1:10;
semilogx(10.^x,x);grid
title('Semilogx')
subplot(2,2,3)
x = logspace(-1,2);

```

```

loglog(x,exp(x),'-s');grid
title('Loglog')
subplot(2,2,4)
X = 0:pi/10:pi;
Y = sin(X);
E = std(Y)*ones(size(X));
errorbar(X,Y,E);grid
title('Errorbar')
%===== SLIDE 44 =====
% PLOT 3D
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
xlabel('sin(t)')
ylabel('cos(t)')
zlabel('t')
grid;
axis square
%%
clc
clear
% PLOT INTERATIVO
%===== SLIDE 46 =====
t = 0:pi/50:10*pi;
y = sin(t);
%%
clc
clear
% OPERADORES LOGICOS
%===== SLIDE 57 =====
A= 5; B= -3;
eq(A,B) %A igual a B.
ne(A,B) %A diferente de B. A~=B
lt(A,B) %A menor que B. A<B
gt(A,B) %A maior que B. A>B
le(A,B) %A menor ou igual que B. A<=B
ge(A,B) %A maior ou igual que B. A>=B
%===== SLIDE 58 =====
% A&B
% A|B
% ?A
xor(7,0) %Ou Exclusivo
all([6 2 3 6 7])

```

```

%===== SLIDE 59 =====
any([6 0 3 0 0])
find([0 9 4 3 7])
find([0 9 4 3 7]>4)
%%
clc
clear
%      CONTROLOADORES DE FLUXOS
%===== SLIDE 62 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:10
v(i) = 3*i;
end
%===== SLIDE 63 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% WHILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = 1;
while prod(1:n) < 10e100
n = n +1;
end
% O que e prod??
% Digite: help prod
%===== SLIDE 64 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% IF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii = 1:5
    if ii == 3;
        break;
    end
end
%===== SLIDE 65 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% IF-ELSE - END %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vetor = ['A','B','C','D','E','F','G','H','I','J','L','M','N','O',...
        'P','Q','R','S','T','U','V','W','X','Z'];
n = length(vetor);
for i = 1:n
    if i == 20
        msg(1) = vetor(i);
    elseif i == 6
        msg(2) = vetor(i);
    elseif i == 1
        msg(3) = vetor(i);
    elseif i == 2
        msg(4) = vetor(i);

```

```

elseif i == 3
    msg(5) = vetor(i);
else
    continue;
end
end
%===== SLIDE 67 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SWITCH %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dia = today;           % today retorna o dia na forma serial
switch weekday(dia)    % retorna o dia da semana (1 - 7)
    case 1
        display('Domingo');
    case 2
        display('Segunda');
    case 3
        display('Terça');
    case 4
        display('Quarta');
    case 5
        display('Quinta');
    case 6
        display('Sexta');
    case 7
        display('Sábado');
    otherwise
        display('Valor Inválido');
end
%%
clc
clear
%      Variavel Simbolica
%===== SLIDE 69 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ver symbolic
x = sym('x');           % declarando variavel x como symbolic
A = [sin(x), x; cos(x), x]
%===== SLIDE 70 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Comando sym:
a = sym('alpha');
% Comando syms:
syms x a y;

```



```

%===== SLIDE 71 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x_1 = sym('x','real')      %Assume que a variavel x e real.
x_2 = sym('x','positive')  %Assume que x e real e positivo.
x_3 = sym('x','clear')     %Limpa o que havia assumido.
m=3; n=2;
x_4 = sym('x',[m n])       %Cria uma matriz m por n.
x_5 = sym('x',n)           %Cria uma matriz n por n.
x_6 = sym('x',flag)        %Cria um escalar numerico ou matriz.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sym(4/3,'r')               %'r' (racional)
sym(4/3,'d')               %'d' (decimal)
sym(4/3,'e')               %'e' (erro estimado)
sym(4/3,'f')               %'f' (ponto flutuante)
%===== SLIDE 72 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Integral
x = sym('x');
y = int(exp(x)*sin(x))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Eq. 2 grau
eq2 = sym('a*x^2+b*x+c');
s = solve(eq2,'x')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Visualizacao
visual_s = pretty(s)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Derivada parcial
d = diff(y,x,1)
d2 = diff(y,x,2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Limite
syms h
limite = limit((sin(x+h)-sin(x))/h,h,0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simplifica
simplifi_derivada = simplify(d)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expande
expande_seno = expand(sin(h+x))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Agrupar

```

```

f=4*x*exp(x)+3*exp(x);
agrupar_exp = collect(f,exp(x))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sustituicao
subs(y,x,pi)
%===== SLIDE 73 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms h x
li=limit((sin(x+h)-sin(x))/h,h,0);
ezplot(li,[-2*pi,2*pi])
%===== SLIDE 76 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dh = clock;
str_dh = sprintf('%04d-%02d-%02d %02d:%02d:%02.0f',...
                 dh(1),dh(2),dh(3),dh(4),dh(5),dh(6))
sprintf('Alfabeto \n%s',65:89)
% quebrei em dois somente para caber na folha
f1=('Em caso de emergência:\nPolícia:%1.0f%d%1.0f\nBombeiros:');
f2('%c%c%1.0f\nPizza: Anuncie aqui');
f = [f1 f2];
sprintf(f,1.3*cos(2-2),9,sin(2-2),'1','9',pi)
%===== SLIDE 77 =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = 0:10;
y = sin(x);
xi = 0:.25:10;
yi = interp1(x,y,xi);
plot(x,y,'o',xi,yi);grid

```

2 SUSPENSÃO PASSIVA

2.1 Código Principal (Suspensão Passiva)

```

% Este e o arquivo principal para a resolução do modelo da suspensao
%%
clear all
close all
clc
%%
asf_t = linspace(0,20,900); % gera o tempo da entrada asfalto
u=zeros(1,900/3);

```

```

asf=[u, ones(1,900/3), u];
T_sim = [0 20];
Cond_Ini = [0;0;0;0];
%%
[t,x,dot_x]= ode45(@ (t,x,dot_x) modelo_suspensao_passiva(t,x,asf_t,asf),...
                    T_sim,Cond_Ini);
%%
x(:,5) = interp1(asf_t,asf,t);
%%
figures = plot_EDO_v1(t,x);

```

2.2 Código Modelagem (Suspensão Passiva)

```

function dot_x = modelo_suspensao_passiva(t,x,asf_t,asf)
% Esta funcao descreve a modelagem de uma suspensao passiva, onde:
% x(1) = Z_s - Z_us
% x(2) = dot_Z_s
% x(3) = Z_us - Z_r
% x(4) = dot_Z_us
%%
    B_s=7.5;
    K_s=900;
    K_us=2500;
    M_s=2.45;
    M_us=1;
    dotZ_r = interp1(asf_t,asf,t);
    %%
    dot_x(1) = x(2) - x(4);
    dot_x(2) = -(K_s/M_s)*x(1) - (B_s/M_s)*(x(2)-x(4));
    dot_x(3) = x(4)-dotZ_r;
    dot_x(4) = (K_s/M_us)*x(1)+(B_s/M_us)*(x(2)-x(4))-(K_us/M_us)*x(3);
    dot_x = [dot_x(1);dot_x(2);dot_x(3);dot_x(4)];
end

```

2.3 Código Gráfico (Suspensão Passiva)

```

function figures = plot_EDO_v1(t,x)
% t: vector of x data
% x: vector of y data

```

```

%%
x1 = x(:,1);
x2 = x(:,2);
x3 = x(:,3);
x4 = x(:,4);
zr = x(:,5);
%%
figures = figure;    % Create figure
subplot(2,2,1)
plot(t,x1,'b','LineWidth',3);grid;% Create plot
title('Variável x1')
xlabel('Tempo (s)');
ylabel(sprintf('Deslocamento (m)\n x_%d',1));

subplot(2,2,2)
plot(t,x2,'','LineWidth',3);grid;% Create plot
title('Variável x2')
xlabel('Tempo (s)');
ylabel('Velocidade (m/s)');

subplot(2,2,3)
plot(t,x3,'','LineWidth',3);grid;% Create plot
title('Variável x3')
xlabel('Tempo(s)');
ylabel('Deslocamento (m)');

subplot(2,2,4)
plot(t,x4,'','LineWidth',3);grid;% Create plot
title('Variável x4')
xlabel('Tempo (s)');
ylabel('Velocidade (m/s)');

figures = figure;    % Create figure
plot(t,x2,'',t,x4,'',t,zr,':r','LineWidth',3);grid;
% axis([0 20 -1 2])
xlabel('Tempo (s)');
ylabel('Velocidade (m/s)');
title('Velocidades');
legend('dZ_s','dZ_{us}','Z_r');

```

3 SUSPENSÃO ATIVA

3.1 Código Principal (Suspensão Ativa)

```
% Este e o arquivo principal para a resolucao do modelo da suspensao
%%
clear all
close all
clc
%% SELECAO ASFALTO
% input_asf = 1;      % ENTRADA SENO
input_asf = 2;      % ENTRADA QUADRADA
%% SELECAO SUSPENSAO
input_susp = 1;      % Passiva
% input_susp = 2;      % Ativa
% input_susp = 3;      % Ativa + Passiva
%%
asf_t = linspace(0,20,900); % gera o tempo da entrada asfalto
T_sim = [0 20];
Cond_Ini = [0;0;0;0];
if input_asf ==1
    asf = sin(asf_t);
elseif input_asf == 2
    u=zeros(1,900/3);
    asf=[u, ones(1,900/3), u];
end
%%
if input_susp ==1
    [t,x]=ode45(@(t,x) modelo_suspensao_passiva(t,x,asf_t,asf),...
                T_sim,Cond_Ini);
    x(:,5) = interp1(asf_t,asf,t);
    opts = [input_susp,input_asf];
    F=0;
    figures = plot_EDO(t,x,F,opts);
elseif input_susp == 2
    K = [24.66 48.87 -0.47 3.68];
    [t,x,F]=ode45(@(t,x,F) modelo_suspensao_ativa(t,x,asf_t,asf,K),...
                  T_sim,Cond_Ini);
    x(:,5) = interp1(asf_t,asf,t);
    opts = [input_susp,input_asf];
    figures = plot_EDO(t,x,F,opts);
elseif input_susp == 3
    [t,x]=ode45(@(t,x) modelo_suspensao_passiva(t,x,asf_t,asf),...
```

```

        T_sim,Cond_Ini);
x(:,5) = interp1(asf_t,asf,t);
opts = [input_susp,input_asf];
F=0;
figures = plot_EDO(t,x,F,opts);
%%
K = [24.66 48.87 -0.47 3.68];
[t,x,F]=ode45(@ (t,x,F) modelo_suspensao_ativa(t,x,asf_t,asf,K),...
        T_sim,Cond_Ini);
x(:,5) = interp1(asf_t,asf,t);
opts = [input_susp,input_asf];
figures = plot_EDO(t,x,F,opts);
else
    sprintf('\nErro, variável\n input_susp=%d',input_susp)
end

```

3.2 Código Modelagem (Suspensão Ativa)

```

function dot_xa = modelo_suspensao_ativa(t,xa,asf_t,asf,K,F)
% Esta funcao descreve a modelagem de uma suspensao passiva, onde:
% x(1) = Z_s - Z_us
% x(2) = dot_Z_s
% x(3) = Z_us - Z_r
% x(4) = dot_Z_us
%%
B_s=7.5;
B_us = 5;
K_s=900;
K_us=2500;
M_s=2.45;
M_us=1;
F = -K*xa;
dotZ_r = interp1(asf_t,asf,t);
%%
dot_xa(1) = xa(2) - xa(4);
dot_xa(2) = -(K_s/M_s)*xa(1) - (B_s/M_s)*(xa(2)-xa(4)) + (F/M_s);
dot_xa(3) = xa(4)-dotZ_r;
dot_xa(4) = (K_s/M_us)*xa(1)+(B_s/M_us)*xa(2)-((B_s+B_us)/M_us)*xa(4)...
            -(K_us/M_us)*xa(3)-(F/M_us)+(B_us/M_us)*dotZ_r;
dot_xa = [dot_xa(1);dot_xa(2);dot_xa(3);dot_xa(4)];
end

```

3.3 Código Gráfico (Suspensão Ativa)

```
function figures = plot_EDO(t,x,F,opts)
%CREATEFIGURE(T1,Y1)
% t: vector of x data
% x: vector of y data
% F: vector of force
% opts: [Ativa/Passiva,Asfalto]
%         Passiva == 1
%         Ativa == 2
%         Passiva e Ativa == 3
%         Seno == 1
%         Quadrado == 2

%%
if norm(F,2) == 0 && opts(1) == 2
    if opts(2) == 1
        title_str = ('Suspensão Ativa,Força Nula,Entrada Senoidal');
    elseif opts(2) == 2
        title_str = ('Suspensão Ativa,Força Nula,Entrada Quadrada');
    else
        title_str = ('Suspensão Ativa,Força Nula,Erro');
    end
    resposta = sprintf('\nNorma do vetor força:\n%d',norm(F,2))
elseif norm(F,2) == 0 && opts(1) == 1
    if opts(2) == 1
        title_str = ('Suspensão Passiva,Entrada Senoidal');
    elseif opts(2) == 2
        title_str = ('Suspensão Passiva,Entrada Quadrada');
    else
        title_str = ('Suspensão Passiva,Erro');
    end
    resposta = sprintf('\nNorma do vetor força:\n%d',norm(F,2))
elseif norm(F,2) ~= 0 && opts(1) == 2
    if opts(2) == 1
        title_str = ('Suspensão Ativa,Força Nula,Entrada Senoidal');
    elseif opts(2) == 2
        title_str = ('Suspensão Ativa,Força Nula,Entrada Quadrada');
    else
        title_str = ('Suspensão Ativa,Força Nula,Erro');
    end
    resposta = sprintf('\nNorma do vetor força:\n%d',norm(F,2))
elseif norm(F,2) == 0 && opts(1) == 3
```

```

if opts(2)== 1
    title_str = ('Suspensão Passiva,Entrada Senoidal');
elseif opts(2) == 2
    title_str = ('Suspensão Passiva,Entrada Quadrada');
else
    title_str = ('Suspensão Passiva,Erro');
end
resposta = sprintf('\nNorma do vetor força:\n%d',norm(F,2))
elseif norm(F,2) ~= 0 && opts(1) == 3
if opts(2)== 1
    title_str = ('Suspensão Ativa,Força Nula,Entrada Senoidal');
elseif opts(2) == 2
    title_str = ('Suspensão Ativa,Força Nula,Entrada Quadrada');
else
    title_str = ('Suspensão Ativa,Força Nula,Erro');
end
resposta = sprintf('\nNorma do vetor força:\n%d',norm(F,2))
else
    title_str = ('Erro');
end

%%
x1 = x(:,1);
x2 = x(:,2);
x3 = x(:,3);
x4 = x(:,4);
zr = x(:,5);
%%
figures = figure;    % Create figure
subplot(2,2,1)
plot(t,x1,'b','LineWidth',3);grid;% Create plot
title(title_str)
xlabel('Tempo (s)');
ylabel(sprintf('Deslocamento (m)\n x_%d',1));

subplot(2,2,2)
plot(t,x2,'r','LineWidth',3);grid;% Create plot
title(title_str)
xlabel('Tempo (s)');
ylabel('Velocidade (m/s)');

subplot(2,2,3)

```



```

plot(t,x3,'','LineWidth',3);grid;% Create plot
title(title_str)
xlabel('Tempo(s)');
ylabel('Deslocamento (m)');

subplot(2,2,4)
plot(t,x4,'','LineWidth',3);grid;% Create plot
title(title_str)
xlabel('Tempo (s)');
ylabel('Velocidade (m/s)');

figures = figure; % Create figure
plot(t,x2,'',t,x4,'',t,zr,':r','LineWidth',3);grid;
% axis([0 20 -1 2])
title(title_str)
xlabel('Tempo');
ylabel('Velocidade');
title('Velocidade');
legend('dZ_s','dZ_{us}','Z_r');
%% ou pode-se otimizar o codigo
figures = figure; % Create figure
for i = [1,3]
    subplot(2,2,i)
    plot(t,x(:,i),'b','LineWidth',3);grid;% Create plot
    title(title_str)
    xlabel('Tempo (s)');
    ylabel(sprintf('Deslocamento (m)\nVariável x_%d',i));
    i=i+1;
    subplot(2,2,i)
    plot(t,x(:,i),'','LineWidth',3);grid;% Create plot
    title(title_str)
    xlabel('Tempo (s)');
    ylabel(sprintf('Velocidade(m/s)\nVariável x_%d',i));
end

```

4 SIMULINK®

A parte da suspensão passiva foi realizada ineitavelmente no SIMULINK®, portanto não possui nenhum código no MATLAB®.

4.1 Suspensão Ativa SIMULINK®.

```
% Este e o arquivo DEVE ser simulado primeiro para carregar os
% valores das matrizes no WORKSPACE, so entao simular o
% Suspensao_Ativa.mdl ou Suspensao_Ativa.slx
%%
clear all
close all
clc
%% CONSTANTES
B_s=7.5;
B_us = 5;
K_s=900;
K_us=2500;
M_s=2.45;
M_us=1;
%% MATRIZES
A = [ 0, 1, 0, -1;
      -K_s/M_s, -B_s/M_s, 0, B_s/M_s;
      0, 0, 0, 1;
      K_s/M_us, B_s/M_us, -K_us/M_us, -(B_s+B_us)/M_us];
%=====
B = [ 0, 0;
      0, 1/M_s;
      -1, 0;
      (B_us/M_us), -1/M_us];
%=====
C = [eye(4);
      1, 0, 0, 0;
      -K_s/M_s, -B_s/M_s, 0, B_s/M_s]; % A(2,:)
%=====
D = [zeros(4,2);0,0;0,1/M_s]; %B(2,:)
%% CONTROLE, MATRIZ K
% O controle escolhido foi o LQR, pois ele ja vem com o kit
% Q = [0.01,zeros(1,3);zeros(3,4)];
% R = 0.01;
% K = lqr(A,B(:,2),Q,R);
% obtem-se:
K = [24.66 48.87 -0.47 3.68];
```

5 SIMULINK® & MATLAB®

5.1 Código Principal

```
% Este e o arquivo, carrega os parametros do modelo
% (suspensao ativa, QUANSER), faz a simulacao do arquivo
% (,simulink) e por fim faz os plot
%%
clear all
close all
clc
%% CONSTANTES
B_s=7.5;
B_us = 5;
K_s=900;
K_us=2500;
M_s=2.45;
M_us=1;
%% MATRIZES
A = [ 0, 1, 0, -1;
      -K_s/M_s, -B_s/M_s, 0, B_s/M_s;
      0, 0, 0, 1;
      K_s/M_us, B_s/M_us, -K_us/M_us, -(B_s+B_us)/M_us];
=====
B = [ 0, 0;
      0, 1/M_s;
      -1, 0;
      (B_us/M_us), -1/M_us];
=====
C = [eye(4);
      1, 0, 0, 0;
      -K_s/M_s, -B_s/M_s, 0, B_s/M_s]; % A(2,:)
=====
D = [zeros(4,2);0,0;0,1/M_s]; %B(2,:)
%% CONTROLE, MATRIZ K
```

```

% O controle escolhido foi o LQR, pois ele ja vem com o kit
% Q = [0.01,zeros(1,3);zeros(3,4)];
% R = 0.01;
% K = lqr(A,B(:,2),Q,R);
% obtem-se:
K = [24.66 48.87 -0.47 3.68];
%% SELECAO
Sel_Asfalto = 1;      % Onda quadrada
% Sel_Asfalto = 2;    % Senoidal
%=====
Sel_Forca = 1;        % Controle K
% Sel_Forca = 2;      % Zero
%=====
% Sel_Planta = 1;     % Planta Block State Space
Sel_Planta = 2;       % Planta Interpreted MATLAB Function
%% Simulacao SIMULINK
Simu_MAT_SIM_SA = sim('MAT_SIM_SA.mdl',[0 10]);
%% Graficos
%=====
% Qual e o controle
if Sel_Forca == 1
    title_F = sprintf('Controle, K = [%1.2f %1.2f %1.2f %1.2f] ',K);
elseif Sel_Forca == 2
    title_F = ('Sem Controle ');
else
    title_F = sprintf('Erro na Variável\n Sel_Forca = %d',Sel_Forca);
end
%=====
% Qual e o modelo
if Sel_Planta == 1
    title_P = (' State Space');
elseif Sel_Planta == 2
    title_P = (' Interpreted MATLAB Function');
else
    title_P = sprintf('Erro na Variável\n Sel_Planta = %d',Sel_Planta);
end
%=====
% Juntando os dois
title_str = [title_F title_P];
%=====
%%
F = forca(:,2);

```

```

Z_r = Z(:,2);
Z_s = Z(:,3);
Z_us = Z(:,4);
%=====
figuras(1) = figure;
for i = [1,3]
    subplot(2,2,i)
    plot(tempo,XX(:,i),'b','LineWidth',3);grid;% Create plot
    title(title_str)
    xlabel('Tempo (s)');
    ylabel(sprintf('Deslocamento (m)\nVariável x_%d',i));
    i=i+1;
    subplot(2,2,i)
    plot(tempo,XX(:,i),'','LineWidth',3);grid;% Create plot
    title(title_str)
    xlabel('Tempo (s)');
    ylabel(sprintf('Velocidade(m/s)\nVariável x_%d',i));
end
%=====
figuras(2) = figure; % Create figure
plot(tempo,Z_s','',tempo,Z_us','',tempo,Z_r,':r','LineWidth',3);grid;
title(title_str)
xlabel('Tempo (s)');
ylabel('Velocidade (m/s)');
legend('dZ_s','dZ_{us}','Z_r');
%=====
figuras(3) = figure; % Create figure
plot(tempo,F','', 'LineWidth',3);grid;
title(title_str)
xlabel('Tempo (s)');
ylabel('Força (N)');

```

5.2 Modelo da Suspensão Ativa

```

function output = Modelo_SA(input)
% Modelo da Suspensao Ativa
% input(1)= X_1
% input(2)= X_2
% input(3)= X_3
% input(4)= X_4
% input(5) = Z_r
% input(6) = F

```

```

%
% output(1)= X_1
% output(2)= X_2
% output(3)= X_3
% output(4)= X_4
% output(5)= Y_1
% output(6)= Y_2
%=====
x = [input(1);input(2);input(3);input(4)];
Z_r = input(5);
F = input(6);
%=====
%% CONSTANTES
B_s=7.5;
B_us = 5;
K_s=900;
K_us=2500;
M_s=2.45;
M_us=1;
%=====
%%
output(1) = x(2) - x(4);
output(2) = -(K_s/M_s)*x(1) - (B_s/M_s)*(x(2)-x(4)) + (F/M_s);
output(3) = x(4)-Z_r;
output(4) = (K_s/M_us)*x(1) + (B_s/M_us)*x(2) - ((B_s+B_us)/M_us)*x(4) ...
            - (K_us/M_us)*x(3) - (F/M_us) + (B_us/M_us)*Z_r;
%=====
output(5) = x(1);
output(6) = output(2);
%=====
output = [output(1);output(2);output(3);output(4);output(5);output(6)];
end

```