EA721 - Princípios de Controle e Servomecanismo
Turma A

Trabalho Computacional 01

Thiago Maximo Pavão - 247381
Vinícius Esperança Mantovani - 247395

Experimento 1 (Figuras):

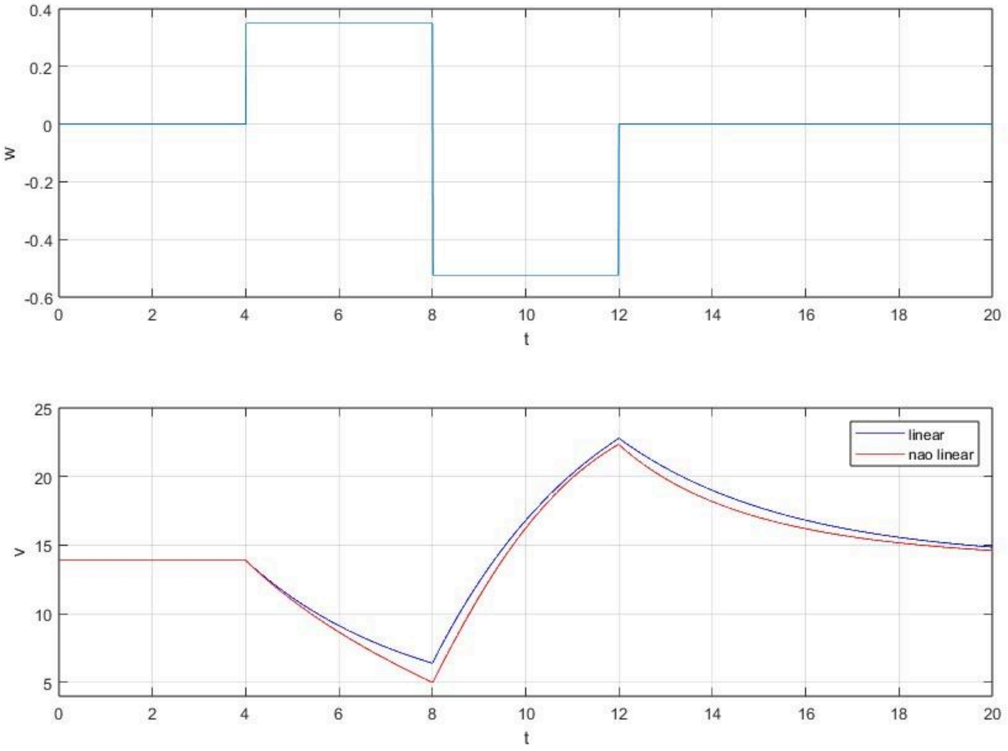


Figura 1: gráficos experimento 1 item a

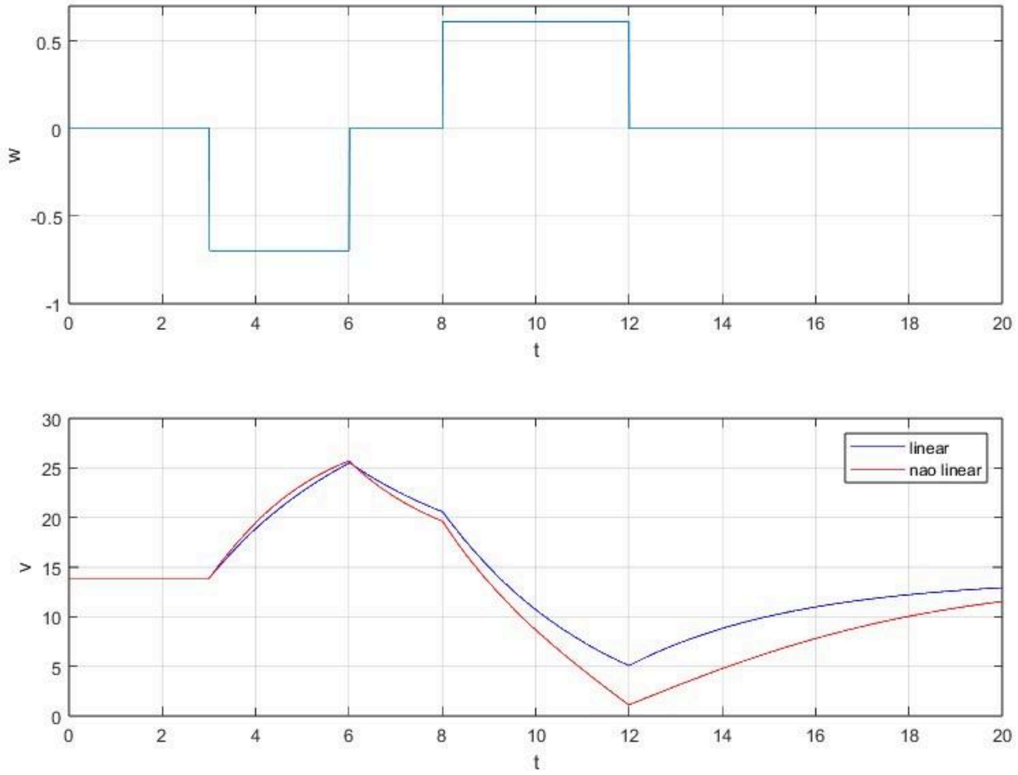


Figura 2: gráficos experimento 1 item b

Experimento 2 (Figuras):

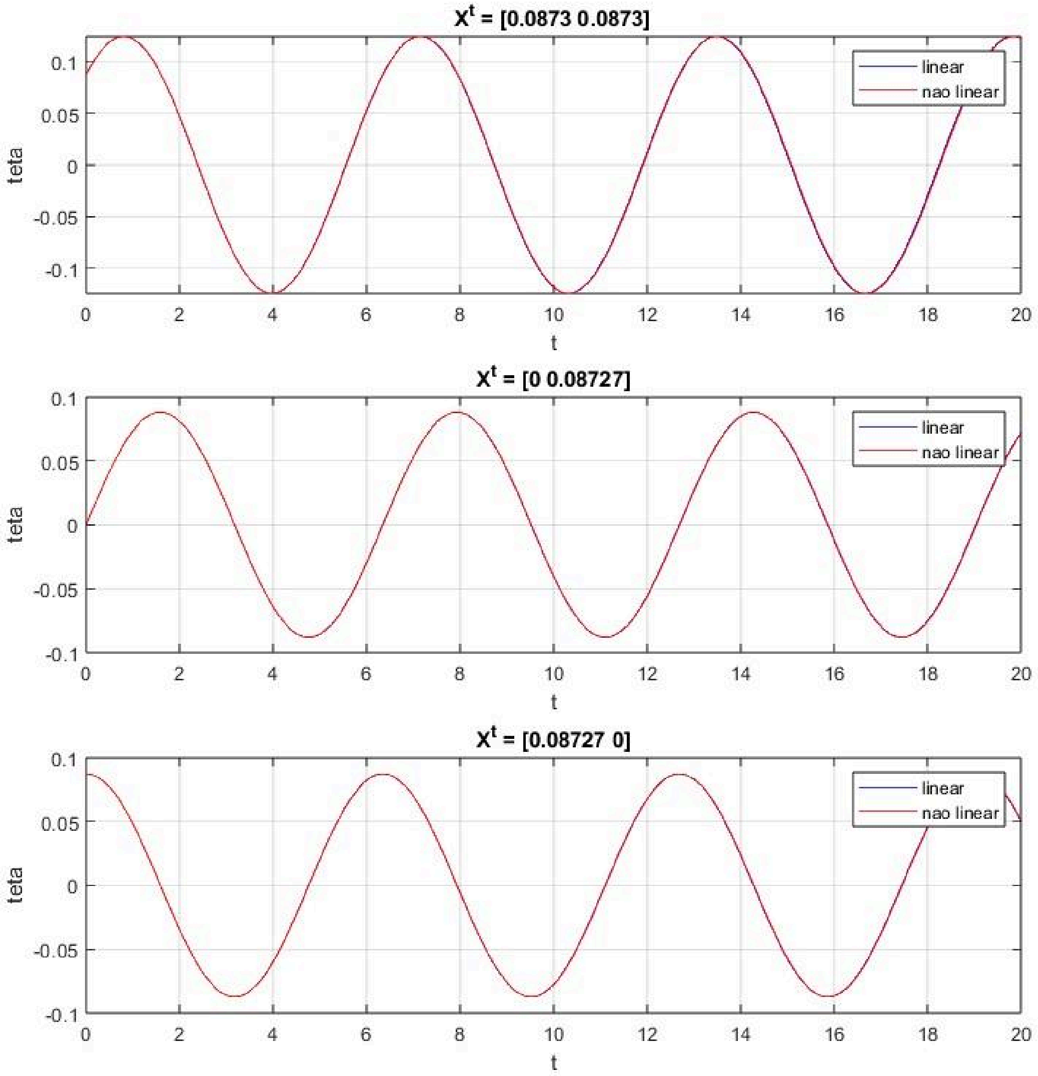


Figura 3: gráficos experimento 2

Experimento 3 (Figuras):

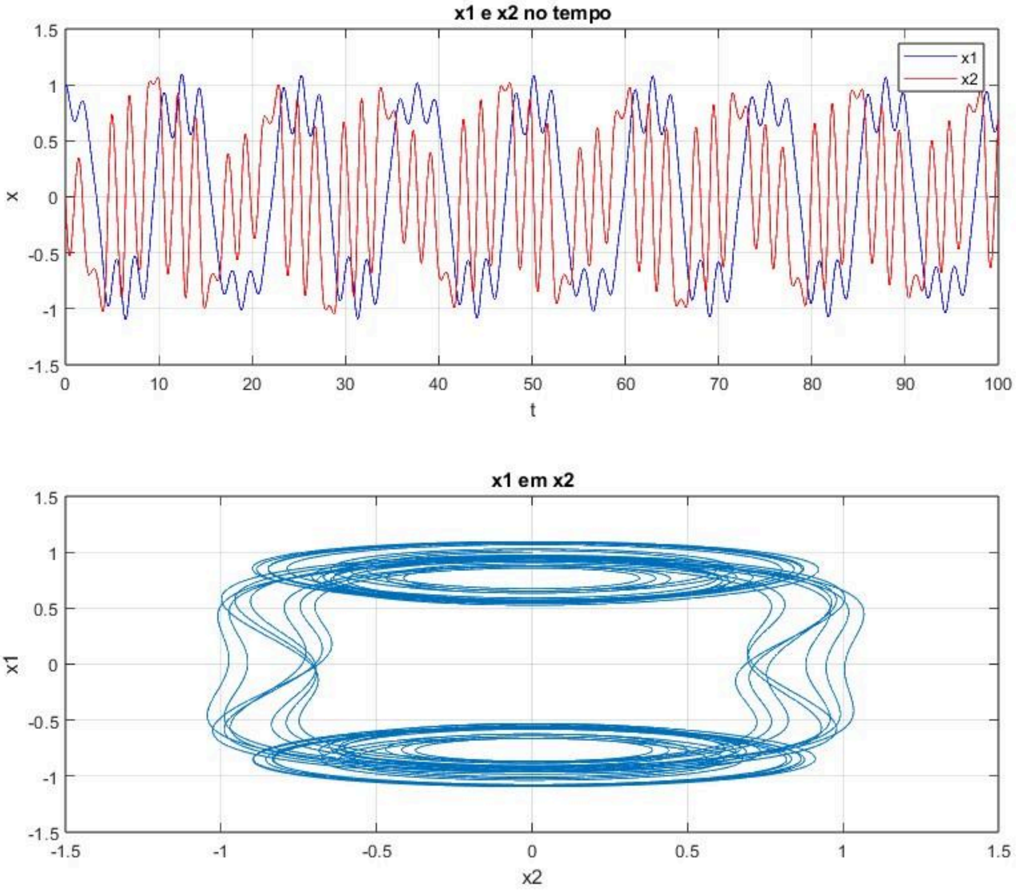


Figura 4: gráficos experimento 3

Experimento 1 (Código fonte):

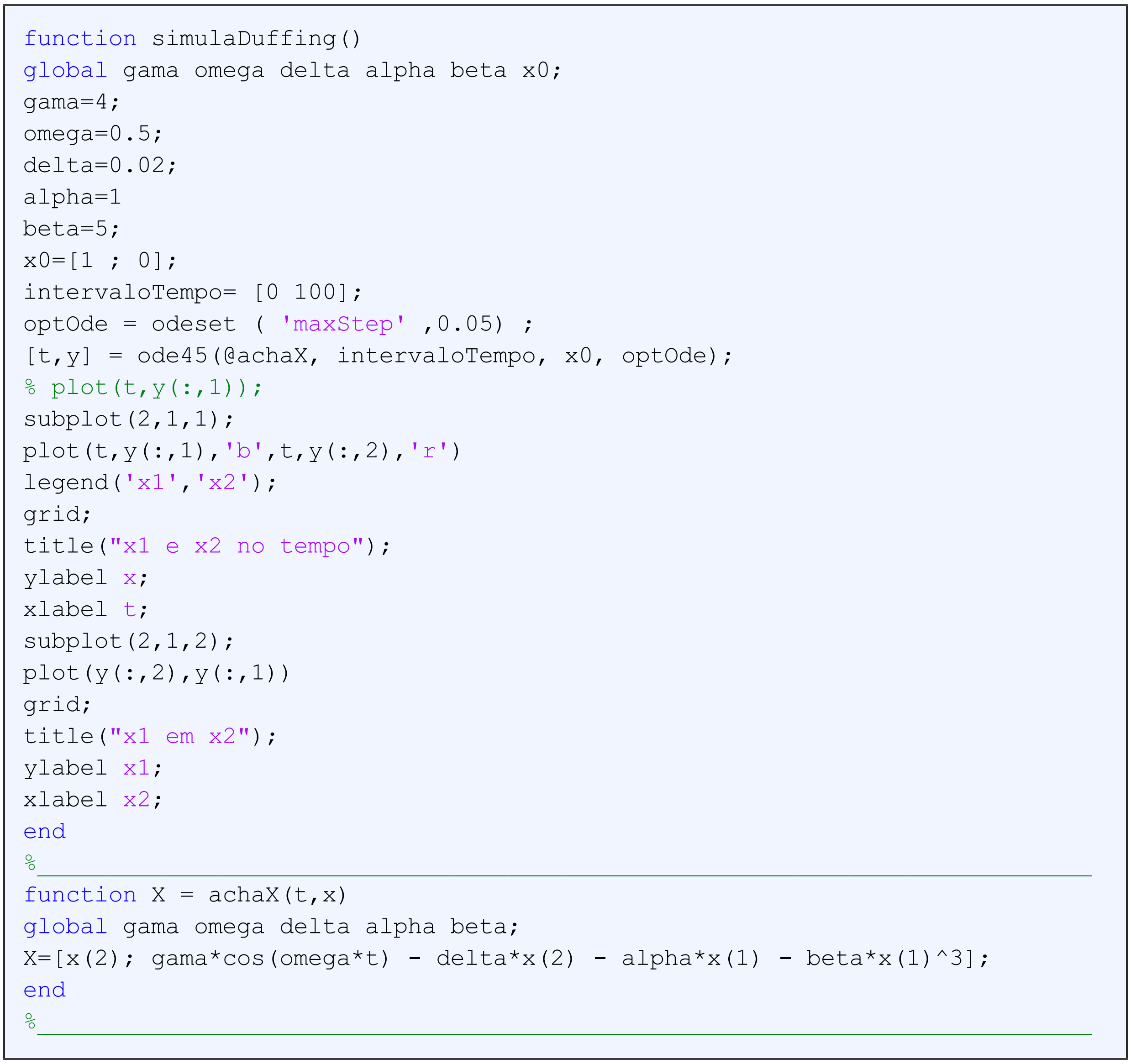
```
function simulaVelocidadeCarro()
global g m k mu u v0 u0 w0;
g=9.81;
m=1;
k=0.01;
mu=0.1;
w0=0;
v0=13.8889;
u0=k*v0*v0+mu*m*g*cos(w0)+m*g*sin(w0);
u=u0;
intervaloTempo= [0 20];
optOde = odeset ( 'maxStep' ,0.05) ;
[t,y] = ode45(@dinamicaCarro, intervaloTempo, v0,optOde);
subplot(2,1,1)
for i=1:length(t)
   ww(i)=sinalW(t(i));
end
plot(t,ww)
grid;
```

```
ylabel w;
xlabel t;
subplot(2,1,2)
[tl,yl] = ode45(@dinamicaCarro_linear, intervaloTempo, 0,optOde);
plot(tl,yl+v0,'b',t,y,'r')
legend('linear','nao linear');
grid;
ylabel v;
xlabel t;
%__________________________________________________________________________
function dotV = dinamicaCarro(t,v)
global g m k mu u v0 u0 w0;
w = sinalW(t);
dotV = (-k/m)*v*v -mu*g*cos(w)-g*sin(w)+u;
%__________________________________________________________________________
function dotDeltaV = dinamicaCarro_linear(t,deltaV)
global g m k mu v0 w0;
w = sinalW(t);
deltaW=(w-w0);
deltaU=0; % implica que u=u0;
dotDeltaV = ((-2*k/m)*v0)*deltaV +(mu*g*sin(w)-g*cos(w))*deltaW
+(1/m)*deltaU;
%__________________________________________________________________________
function w = sinalW(t)
% if t < 4
%     w=0;
% elseif t >= 4 & t < 8
%     w=deg2rad(20);
% elseif t >= 8 & t < 12
%     w=deg2rad(-30);
% else
%     w=0;
if t < 3
   w=0;
elseif t >= 3 & t < 6
   w=deg2rad(-40);
elseif t >= 6 & t < 8
   w=0;
elseif t >= 8 & t < 12
   w=deg2rad(35);
else
   w=0;
end
```

Experimento 2 (Código fonte):

```matlab
function simulaPendulo()
global g l teta a x0;
g=9.81;
l=10;
a=g/l;
x0=[0.0873 ; 0.0873];
intervaloTempo= [0 20];
optOde = odeset ( 'maxStep' ,0.05) ;
[t_a,y_a] = ode45(@achaX, intervaloTempo, x0, optOde);
[t1_a,y1_a] = ode45(@achaXlinear, intervaloTempo, x0, optOde);
% plot(t,y(:,1));
subplot(3,1,1);
plot(t1_a,y1_a(:,1),'b',t_a,y_a(:,1),'r')
legend('linear','nao linear');
grid;
title("X^t = [0.0873 0.0873]");
ylabel teta;
xlabel t;
x0=[0 ; 0.08727];
[t_b,y_b] = ode45(@achaX, intervaloTempo, x0, optOde);
[t1_b,y1_b] = ode45(@achaXlinear, intervaloTempo, x0, optOde);
subplot(3,1,2);
plot(t1_b,y1_b(:,1),'b',t_b,y_b(:,1),'r')
legend('linear','nao linear');
grid;
title("X^t = [0 0.08727]");
ylabel teta;
xlabel t;
x0=[0.08727 ; 0];
[t_c,y_c] = ode45(@achaX, intervaloTempo, x0, optOde);
[t1_c,y1_c] = ode45(@achaXlinear, intervaloTempo, x0, optOde);
subplot(3,1,3);
plot(t1_c,y1_c(:,1),'b',t_c,y_c(:,1),'r')
legend('linear','nao linear');
grid;
title("X^t = [0.08727 0]");
ylabel teta;
xlabel t;
end
%__________________________________________________________________________
function X = achaX(t,x)
global g l x0 a;
X=[x(2); -a*sin(x(1))];
end
%__________________________________________________________________________
function X = achaXlinear(t,x)
global g l x0 a;
X=[x(2); -a*x(1)];
end
%__________________________________________________________________________
```

Experimento 3 (Código fonte):



```
function simulaDuffing()
global gama omega delta alpha beta x0;
gama=4;
omega=0.5;
delta=0.02;
alpha=1
beta=5;
x0=[1 ; 0];
intervaloTempo= [0 100];
optOde = odeset ( 'maxStep' ,0.05) ;
[t,y] = ode45(@achaX, intervaloTempo, x0, optOde);
% plot(t,y(:,1));
subplot(2,1,1);
plot(t,y(:,1),'b',t,y(:,2),'r')
legend('x1','x2');
grid;
title("x1 e x2 no tempo");
ylabel x;
xlabel t;
subplot(2,1,2);
plot(y(:,2),y(:,1))
grid;
title("x1 em x2");
ylabel x1;
xlabel x2;
end
%
function X = achaX(t,x)
global gama omega delta alpha beta;
X=[x(2); gama*cos(omega*t) - delta*x(2) - alpha*x(1) - beta*x(1)^3];
end
%
```