Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica e Computação

Controle de um pêndulo invertido

SEL0382 - CONTROLE ROBUSTO

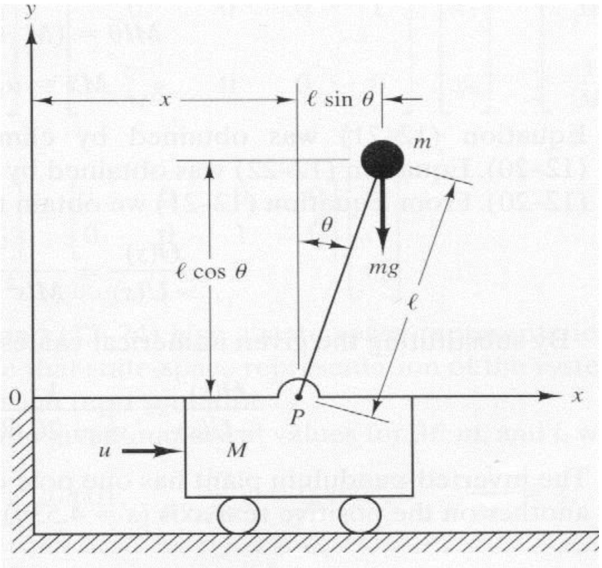
Aluno: Raphael Luiz Vicente Fortulan

Professor: *Vilma Alves de Oliveira*

Sumário

Pêndulo invertido

Equacionamento



Pelas leis de Newton são obtidas as equações abaixo:

1. $\sum x = M\frac{d^2}{dt^2}x + m\frac{d^2}{dt^2}(x + lsin(\theta)) = u$
2. $\sum y = m\frac{d^2}{dt^2}(lcos(\theta)) = mg$
3. $\sum M = (m\frac{d^2}{dt^2}(lcos(\theta)))lsin(\theta) + lcos(\theta)(m\frac{d^2}{dt^2}(x + lsin(\theta))) = lsin(\theta)mg$

São aplicadas as relações:

- $\frac{d^2}{dt^2}(cos(\theta)) = -cos(\theta)\dot{\theta}^2 - sin(\theta)\ddot{\theta}$
- $\frac{d^2}{dt^2}(sin(\theta)) = -sin(\theta)\dot{\theta}^2 + cos(\theta)\ddot{\theta}$

E o sistema é simplificado para o conjunto de equações abaixo:

1. $(M + m)\ddot{x} - mlsin(\theta)\dot{\theta}^2 + mlcos(\theta)\ddot{\theta} = u$
2. $m\ddot{x}cos(\theta) + ml\ddot{\theta} = mgsin(\theta)$

Para acertar a notação das equações é feito o seguinte procedimento:

Aplicando $m\ddot{x}cos(\theta) + ml\ddot{\theta} = mgsin(\theta)$ em $(M+m)\ddot{x} - mlsin(\theta)\dot{\theta}^2 + mlcos(\theta)\ddot{\theta} = u$ é encontrado $(M+m-mcos^2(\theta))\ddot{x} = u+mlsin(\theta)\dot{\theta}^2-mgsin(\theta)cos(\theta)$.

Voltando para a equação $m\ddot{x}cos(\theta) + ml\ddot{\theta} = mgsin(\theta)$ é encontrado que $\ddot{x} = \frac{gsin(\theta)-l\ddot{\theta}}{cos(\theta)}$. Utilizando esse resultado em $(M+m)\ddot{x}-mlsin(\theta)\dot{\theta}^2+mlcos(\theta)\ddot{\theta} = u$ é obtida a equação $(mlcos^2(\theta) - (M+m)l)\ddot{\theta} = ucos(\theta) - (M+m)gsin(\theta) + mlcos(\theta)sin(\theta)\dot{\theta}^2$

As equações finais são:

1. $\ddot{x} = \frac{u+mlsin(\theta)\dot{\theta}^2-mgsin(\theta)cos(\theta)}{(M+m-mcos^2(\theta))}$

2. $\ddot{\theta} = \frac{ucos(\theta)-(M+m)gsin(\theta)+mlcos(\theta)sin(\theta)\dot{\theta}^2}{(mlcos^2(\theta)-(M+m)l)}$

Matricialmente:

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ x\\ \dot{x}\end{bmatrix} = \begin{bmatrix}\dot{\theta}\\ \frac{ucos(\theta)-(M+m)gsin(\theta)+mlcos(\theta)sin(\theta)\dot{\theta}^2}{(mlcos^2(\theta)-(M+m)l)}\\ \dot{x}\\ \frac{u+mlsin(\theta)\dot{\theta}^2-mgsin(\theta)cos(\theta)}{(M+m-mcos^2(\theta))}\end{bmatrix}$$

Linearização

É desejado linearizar o sistema em um ponto de equilíbrio. Analisando as equações diferencias, verifica-se que pontos de equilíbrio são dependentes do valor de θ.Para valores 0,π, 2 π, ... o seno zera e as variáveis zeram também. Assim é escolhido o ponto de equilíbrio $\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$ (instável).

Calcula-se a Jacobiana do espaço de estado:

$$\begin{bmatrix}\frac{\partial f_1}{\partial\theta} & \frac{\partial f_1}{\partial\dot{\theta}} & \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial\dot{x}}\\ \frac{\partial f_2}{\partial\theta} & \frac{\partial f_2}{\partial\dot{\theta}} & \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial\dot{x}}\\ \frac{\partial f_3}{\partial\theta} & \frac{\partial f_3}{\partial\dot{\theta}} & \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial\dot{x}}\\ \frac{\partial f_4}{\partial\theta} & \frac{\partial f_4}{\partial\dot{\theta}} & \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial\dot{x}}\end{bmatrix}_{p.e.} * z + \begin{bmatrix}\frac{\partial f_1}{\partial u}\\ \frac{\partial f_2}{\partial u}\\ \frac{\partial f_3}{\partial u}\\ \frac{\partial f_4}{\partial u}\end{bmatrix}_{p.e.} * u = \begin{bmatrix}0 & 1 & 0 & 0\\ \frac{(M+m)g}{Ml} & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ \frac{-mg}{M} & 0 & 0 & 0\end{bmatrix} * z + \begin{bmatrix}0\\ \frac{-1}{Ml}\\ 0\\ \frac{1}{M}\end{bmatrix} * u.$$

Utilizando que M=2 kg,l=0,5 m,m=0, 1 kg e g=9,81 m/s^2:

A= $\begin{bmatrix}0 & 1 & 0 & 0\\ 20.601 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ -0.4905 & 0 & 0 & 0\end{bmatrix}$

B= $\begin{bmatrix}0\\ -1\\ 0\\ 0.5\end{bmatrix}$

Resposta do sistema linear vs Resposta do sistema não liner para um degrau na entrada
Aplicando um degrau unitário sob a entrada do sistema linearizado e do não linear, são verificadas as respostas para o ângulo e a posição do carrinho.

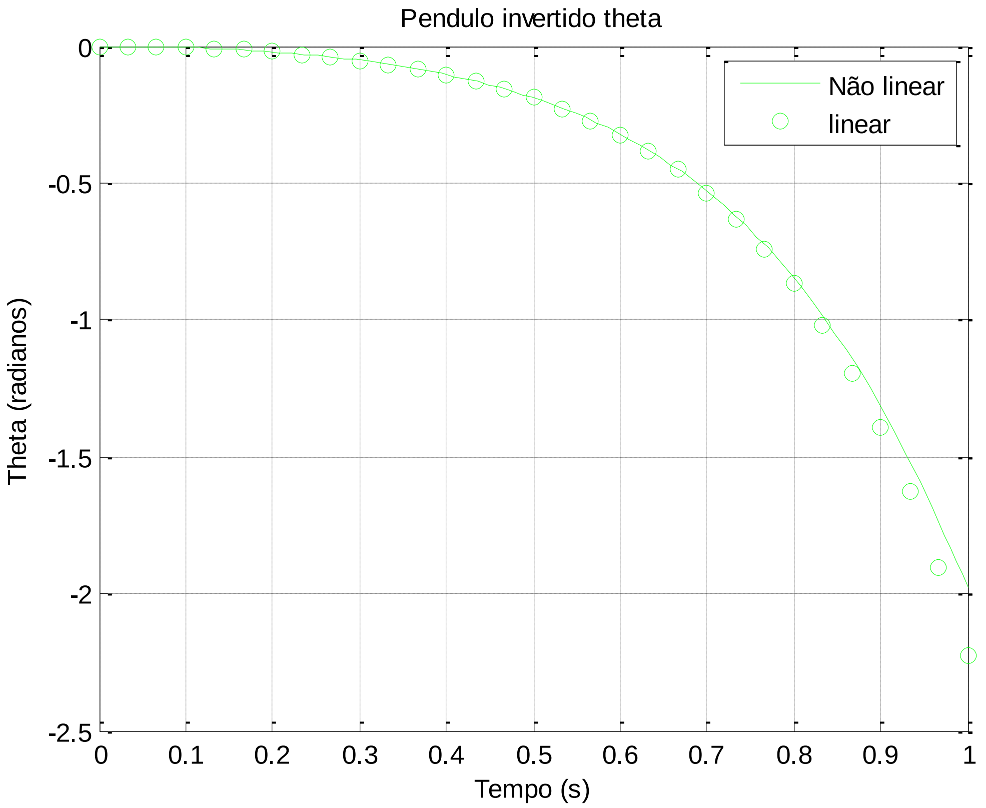


Figura 1 Ângulo vs tempo Linear e Não Linear

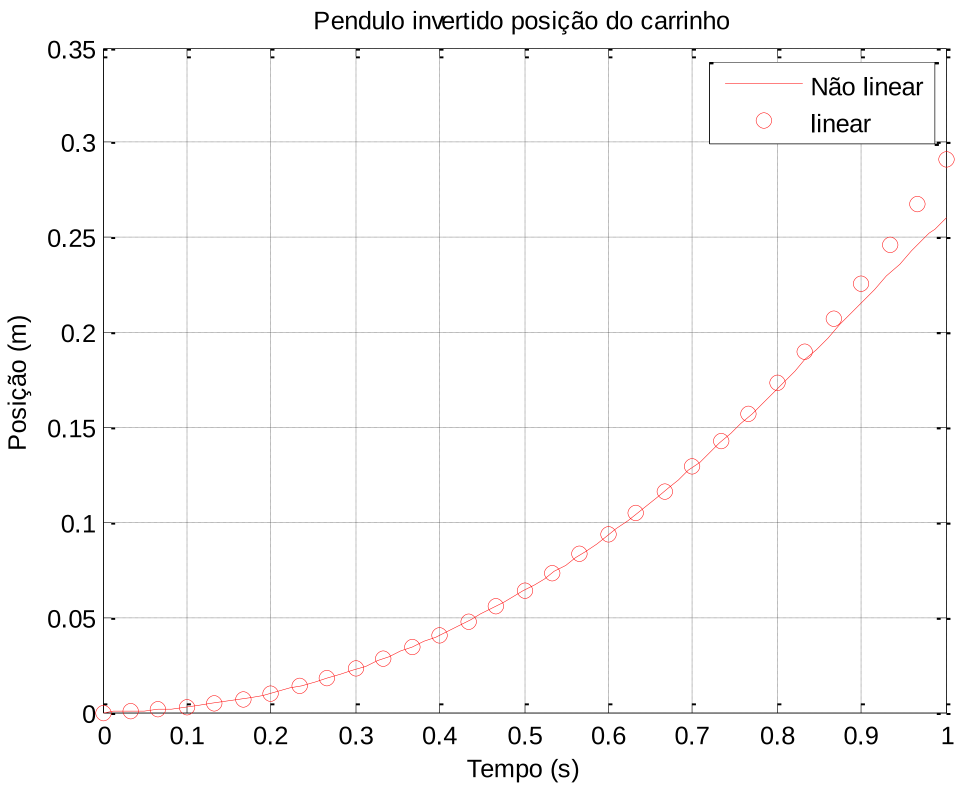


Figura 2 Posição vs tempo Linear e Não Linear

É observado que o sistema linear representa com eficiência o sistema não linear.

Realimentação

É desejado realimentar o sistema. Antes, porém, deve-se verificar se o sistema é controlável. Sabe-se que o sistema linear é controlável se o posto da matriz de controlabilidade for igual à dimensão da matriz A. Então:

$$Cont = [B\ AB\ A^2B\ \dots A^{n-1}\ B] = \begin{bmatrix} 0 & -1 & 0 & -20.16 \\ -1 & 0 & -20.6010 & 0 \\ 0 & 0.5 & 0 & 0.4905 \\ 0.5 & 0 & 0 & 0.4905 \end{bmatrix}.$$

O posto dessa matriz pode ser obtido pela redução de Gauss-Jordan na matriz $Cont$. A matriz reduzida é :

$$Cont_{reduzida} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Assim a matriz possui posto igual a 4 e o sistema é controlável.

Para a realimentação de estados serão utilizadas duas técnicas : LQR e escolha dos polos de malha fechada.

LQR

É necessário encontrar a matriz K de maneira a minimizar a função de custo quadrático $J = \int_0^\infty (x^*Qx + u^*Ru)dt$.

Para resolver esse problema foi utilizado o comando *lqr do* Matlab e as seguintes matrizes de peso:

$$Q_1 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_1 = 100 \text{ e } Q_2 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_2 = 0.1$$

As matrizes utilizadas na realimentação encontradas foram:

$$K_1 = [-44.15 \quad -9.7431 \quad -0.1 \quad -0.7002]$$
$$K_2 = [-66.086 \quad -15.0.313 \quad -3.1623 \quad -6.2709]$$

Escolha dos polos de malha fechada

É escolhida a matriz K de forma que os autovalores de A-B*K fiquem iguais aos polos escolhidos de malha fechada.

Por ser um problema de difícil resolução manual foi utilizado o algoritmo numérico *place* do Matlab.

Os polos de malha fechada escolhidos foram -10,-2,-1+1.5j e -1-1.5j. A matriz utilizada na realimentação encontrada foi:

$$K_3 = [-71.1639 \quad -18.0265 \quad -6.6259 \quad -8.0530]$$

Simulações

O sistema não linear foi realimentado com os controladores baseados no modelo linear, calculados acima. Para verificar a eficiência de cada um, o sistema foi simulado para seis condições iniciais ($\begin{bmatrix} \theta \\ 0 \\ 0 \\ 0 \end{bmatrix}$ com θ=-60, -15, 60, 45,30 e 15 graus), e nenhuma entrada.

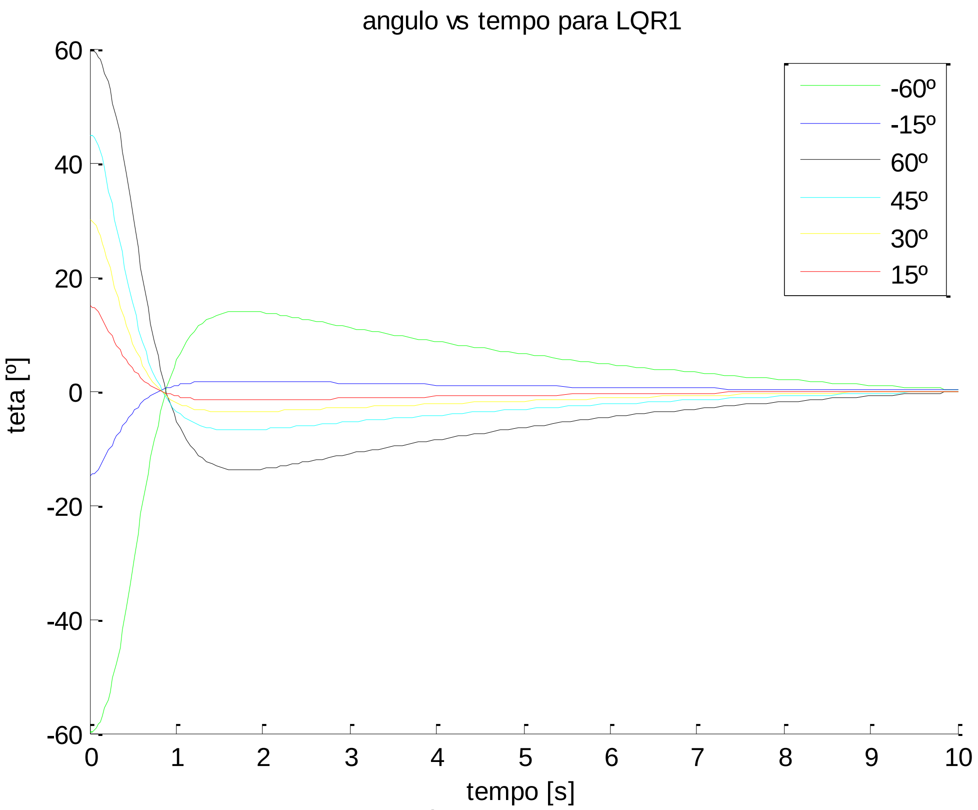


Figura 3- Ângulo vs tempo para LQR1

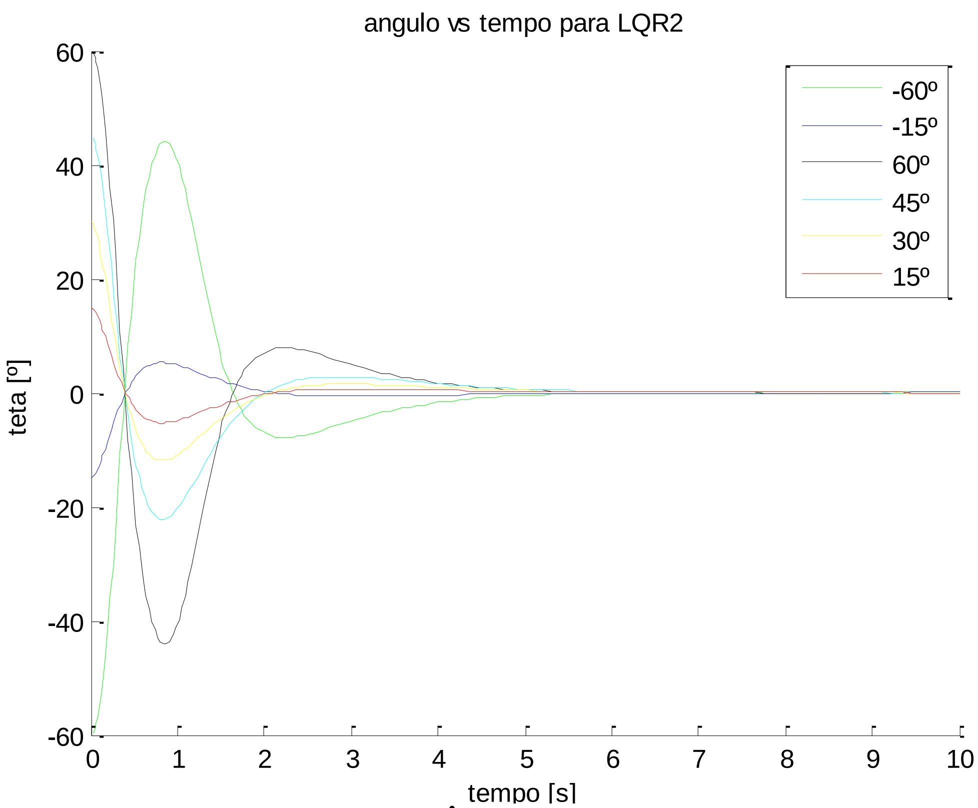


Figura 4- Ângulo vs tempo para LQR2

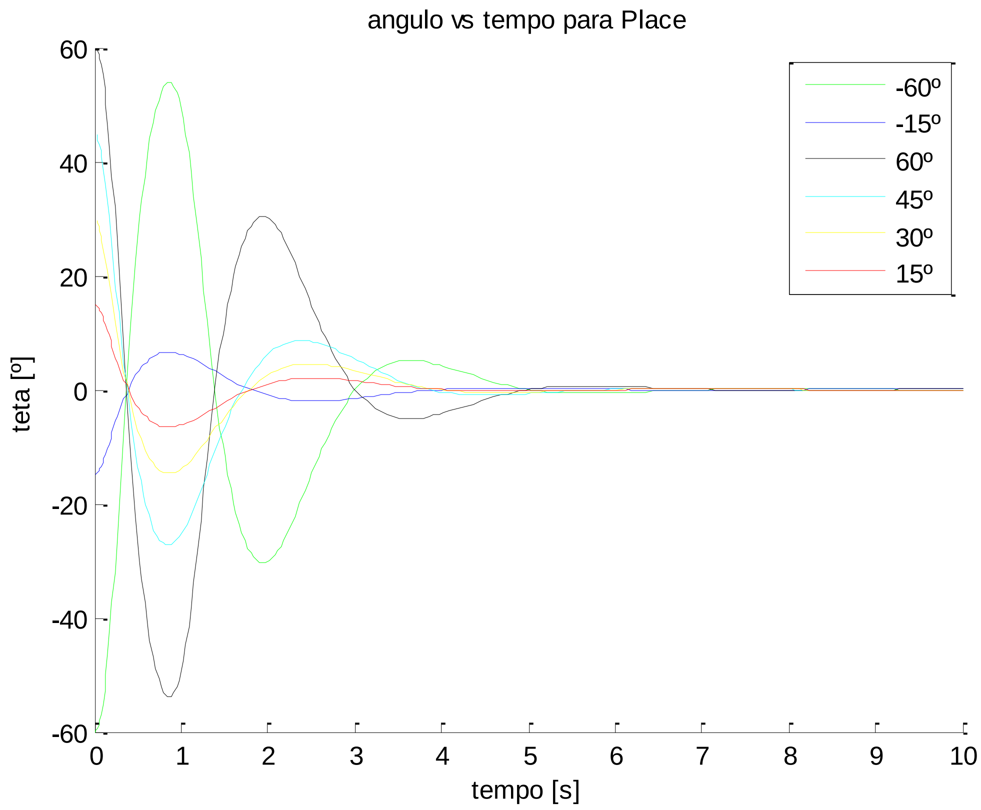


Figura 5- Ângulo vs tempo para Place

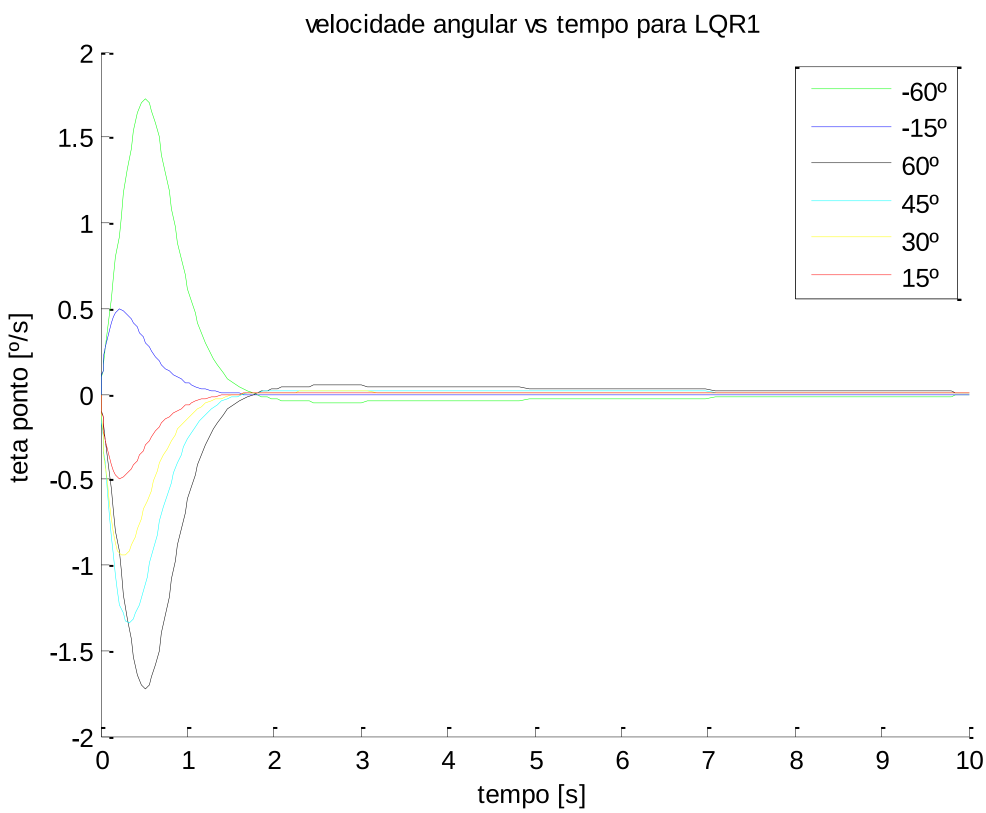


Figura 6- Velocidade Angular vs tempo para LQR1

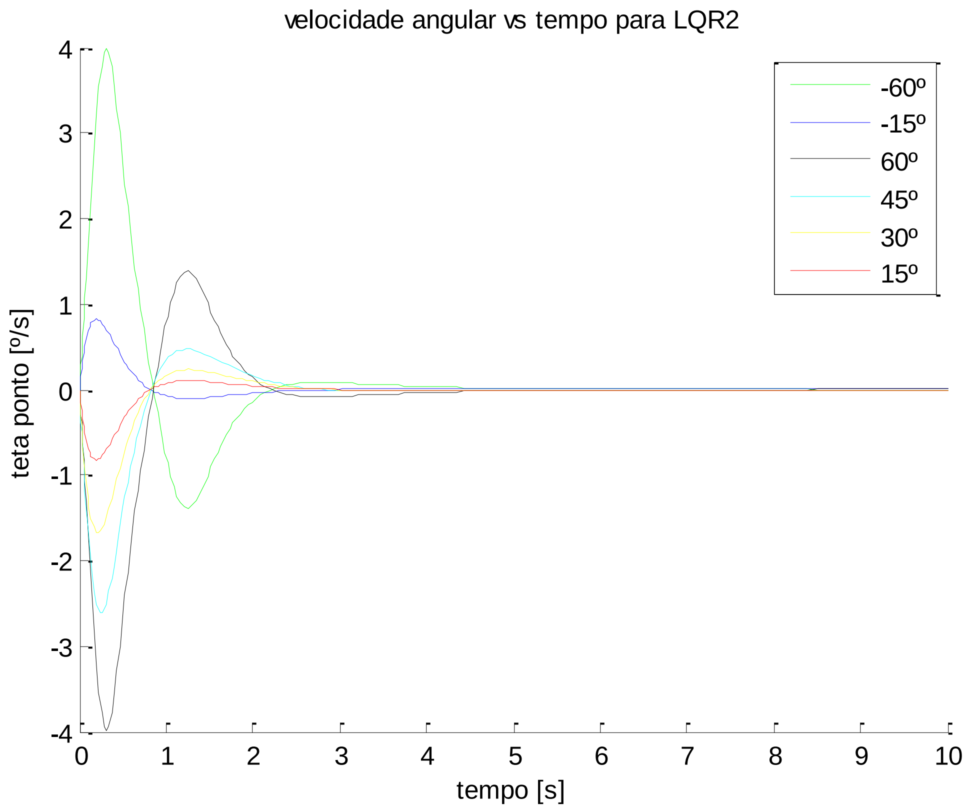


Figura 7- Velocidade Angular vs tempo para LQR2

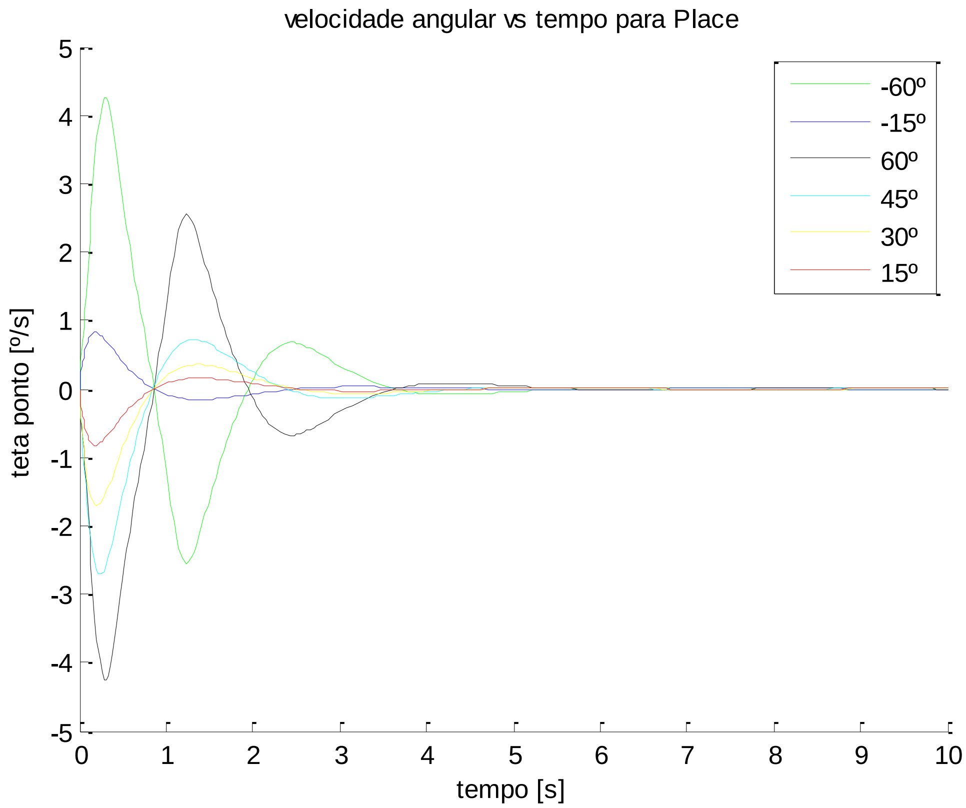


Figura 8- Velocidade Angular vs tempo para Place

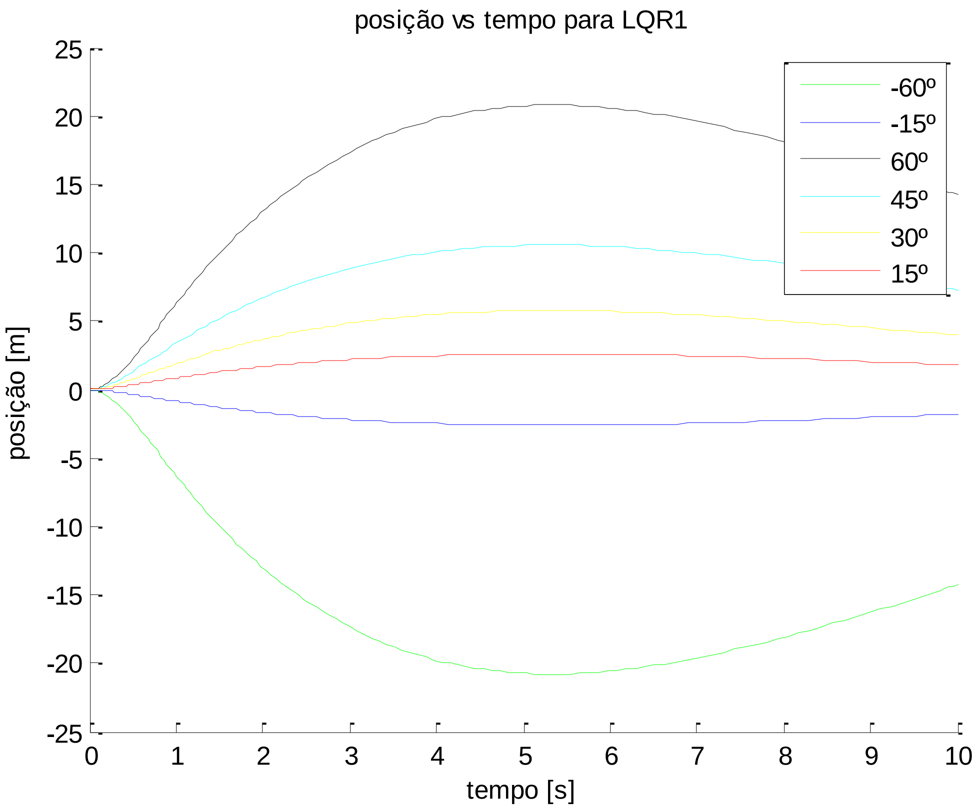


Figura 9- Posição vs tempo para LQR1

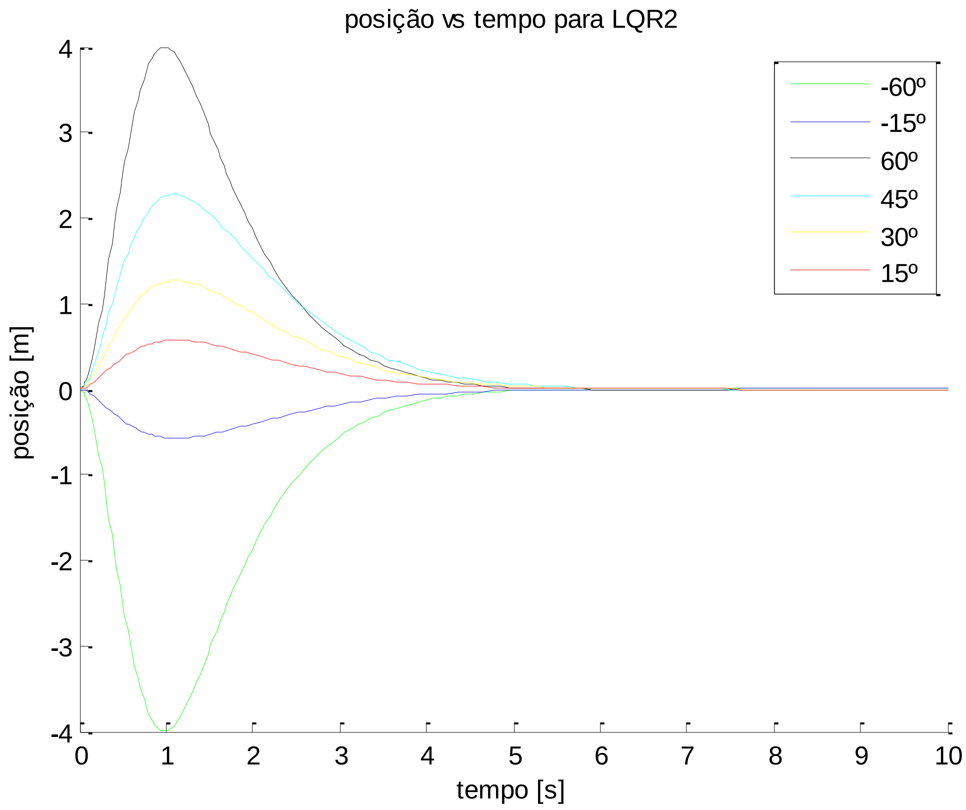


Figura 10- Posição vs tempo para LQR2

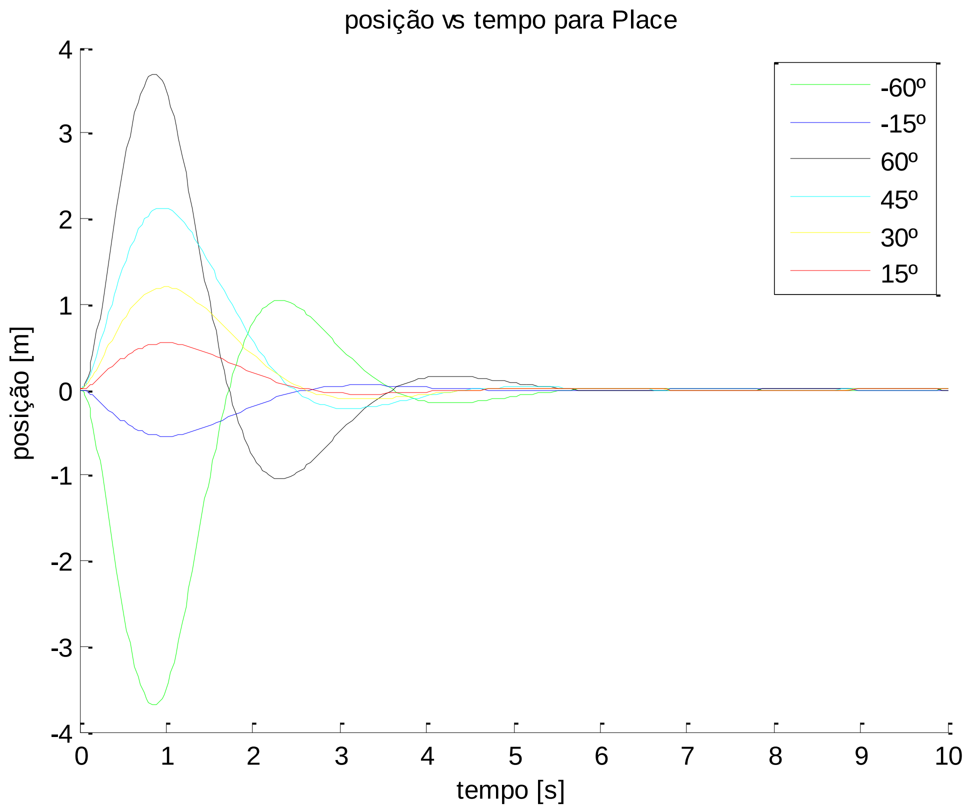


Figura 11 Posição vs tempo para Place

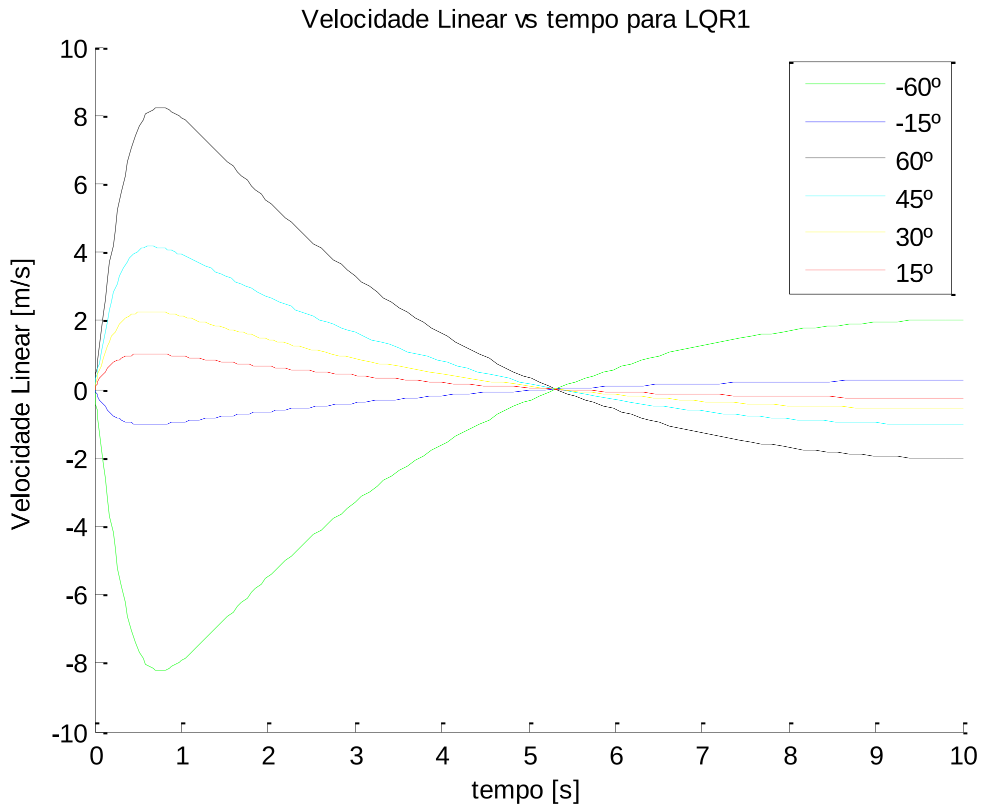


Figura 12- Velocidade Linear vs tempo para LQR1

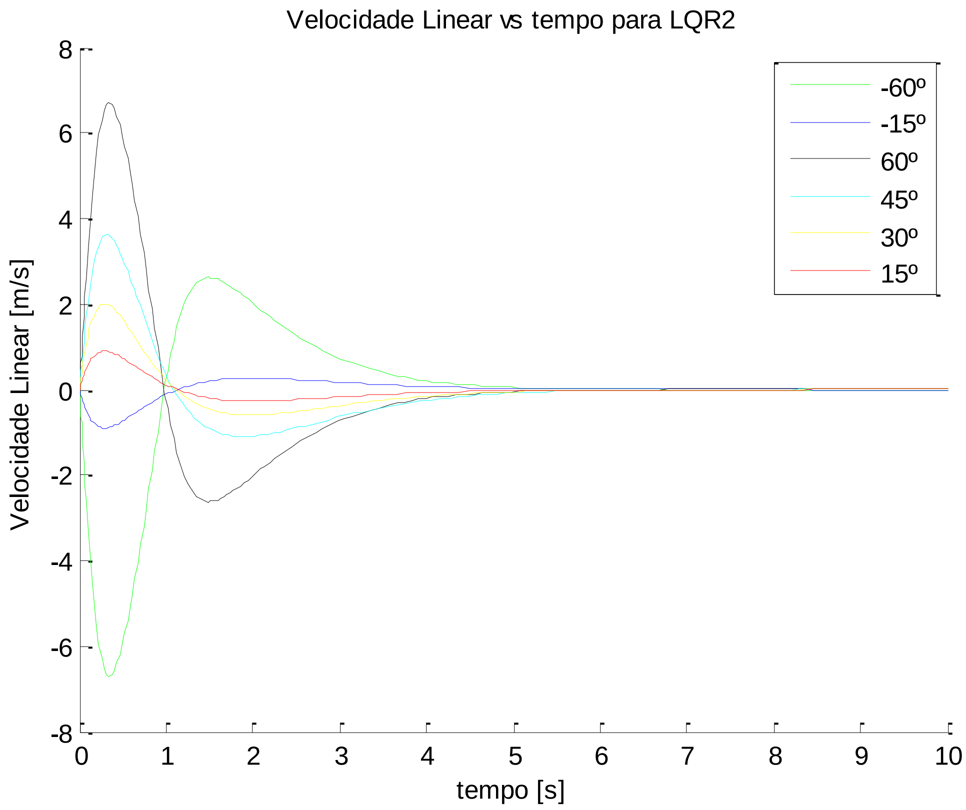


Figura 13- Velocidade Linear vs tempo para LQR2

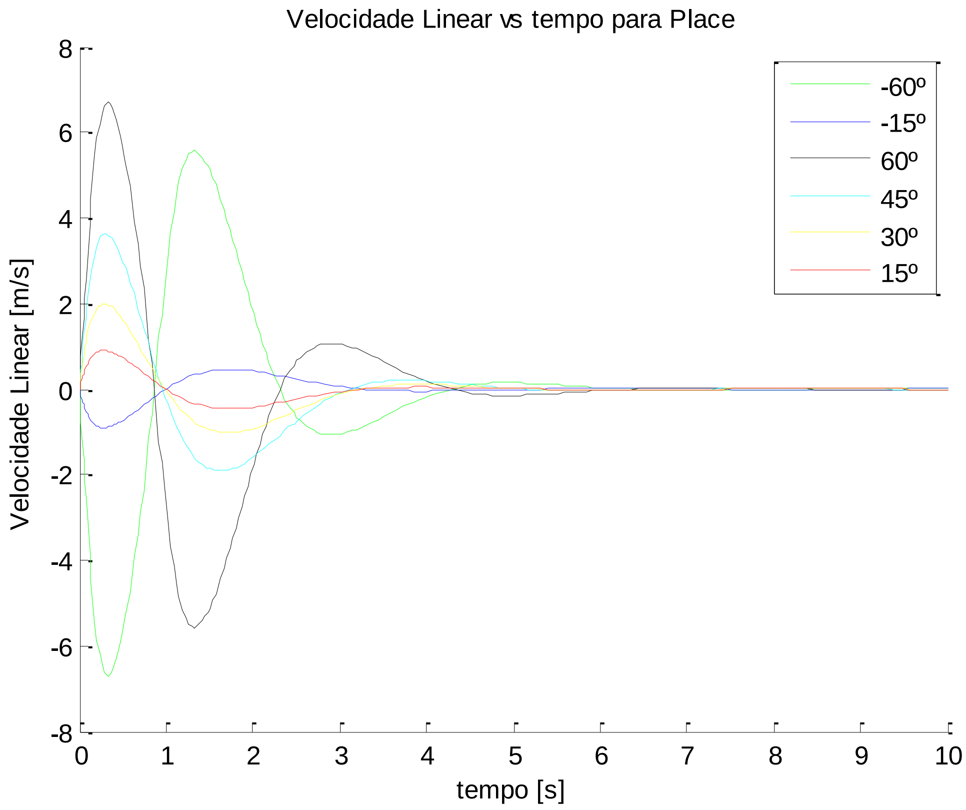


Figura 14- Velocidade Linear vs tempo para Place

Os resultados das Simulações apresentados nas Figuras 3 até 14 verificam que a utilização do LQR1 torna a resposta do sistema muito lenta. Com a escolha dos polos de malha fechada e com a utilização do LQR2, o sistema responde rapidamente e volta para a posição de equilíbrio de maneira eficaz.

Deve-se ressaltar que os princípios físicos de funcionamento do sistema foram mantidos em todas as simulações. Por exemplo, se o ângulo inicial for positivo seria esperado que o carrinho sofresse uma aceleração linear positiva na tentativa de forçar o pêndulo a retornar na posição de repouso. O ângulo iria se tornando cada vez menor e haveria a desaceleração do carrinho, gerando uma velocidade linear negativa sob o ele para compensar a mudança de posição. Porém se a aceleração do carrinho for muito rápida (sistema muito rápido), o pêndulo passaria da posição de equilíbrio e o ângulo ficaria negativo. Para contrabalancear, o carrinho seria acelerado negativamente e o ângulo começaria a ficar mais positivo. O processo ocorreria até que a posição de equilíbrio fosse encontrada. Se observadas as Figuras 3 até 14, verifica-se que realmente as simulações são condizentes com a análise feita.

Com base na análise teórica desenvolvida acima é possível apresentar uma qualidade do controlador baseado no LQR1. Graças à resposta lenta do sistema, durante o transitório, o ângulo não adquire valores muito distantes do ponto de equilíbrio e não há oscilações na posição do carrinho. Já com a escolha dos polos de malha fechada e com a utilização do LQR2, apesar da rapidez da resposta, há oscilações.

Simulink

Foi criado o sistema não linear com o auxílio do Simulink. Para as simulações, foi utilizado o método numérico *ode23* e foi desabilitada a opção de detecção de cruzamento em zero.

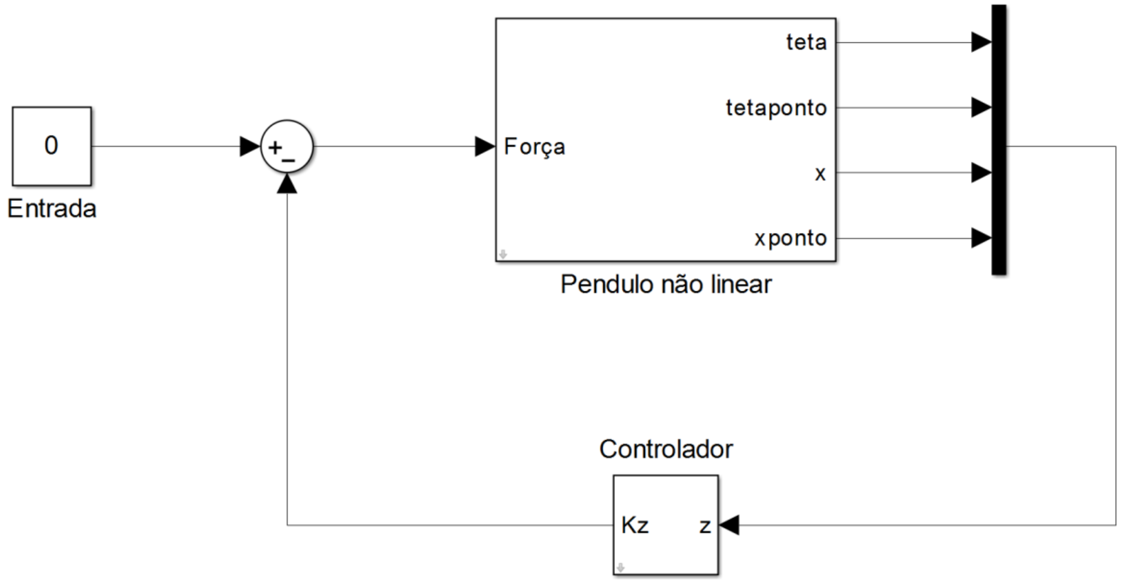


Figura 15- Modelo do Sistema Realimentado

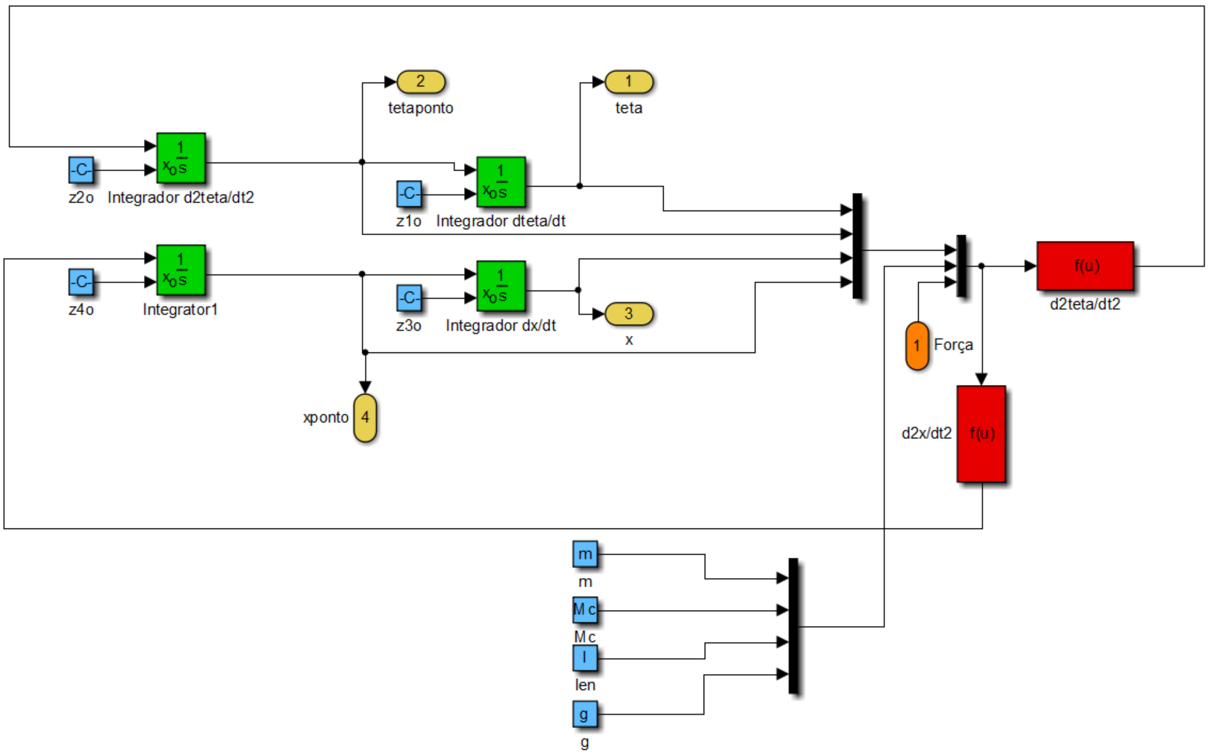


Figura 16- Modelo do Pêndulo Invertido Não Linear

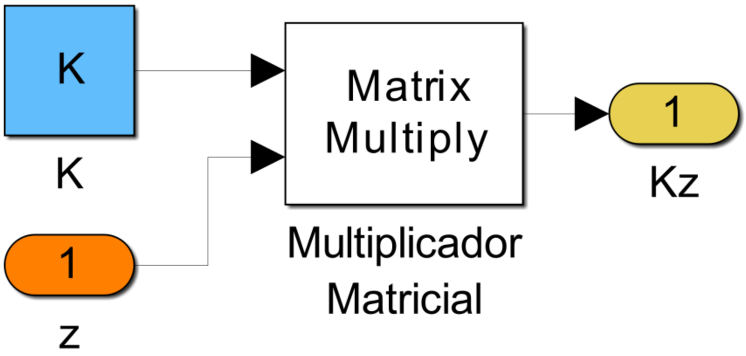


Figura 17- Modelo do Controlador

O resultado das simulações via Simulink ou código de Matlab foram os mesmos. Porém com o Simulink é mais simples de se obter a resposta do controlador. Então, nas Figuras 18,19 e 20 estão apresentadas as respostas do controlador para as mesmas condições inicias das simulações anteriores.

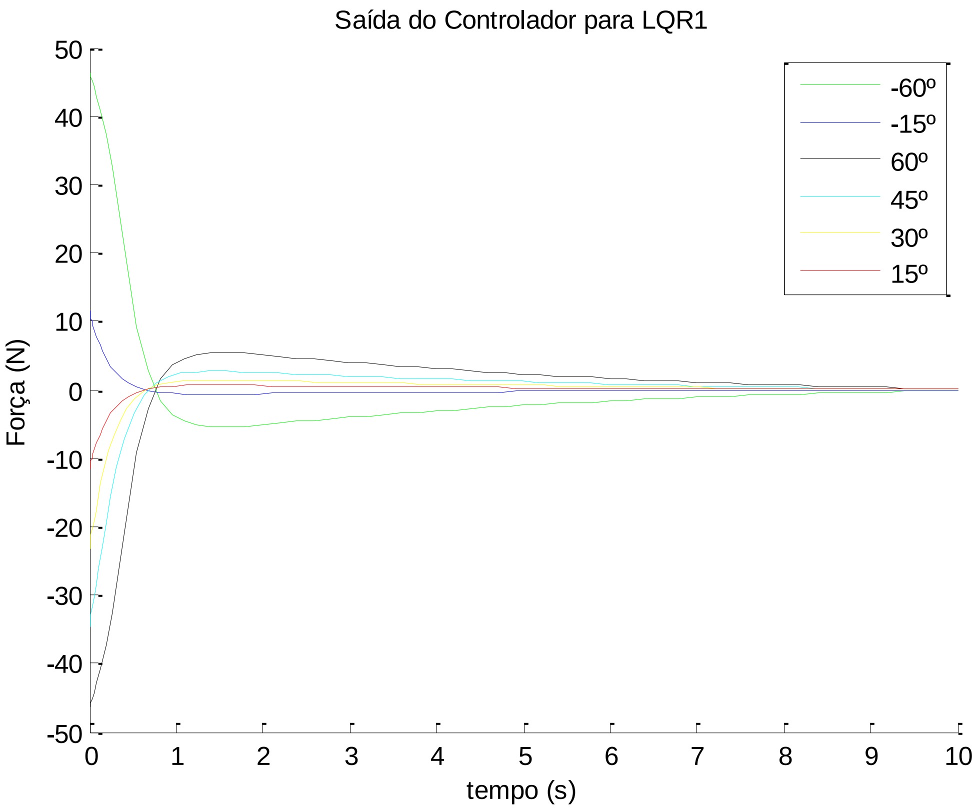


Figura 18- Saída do Controlador para LQR1

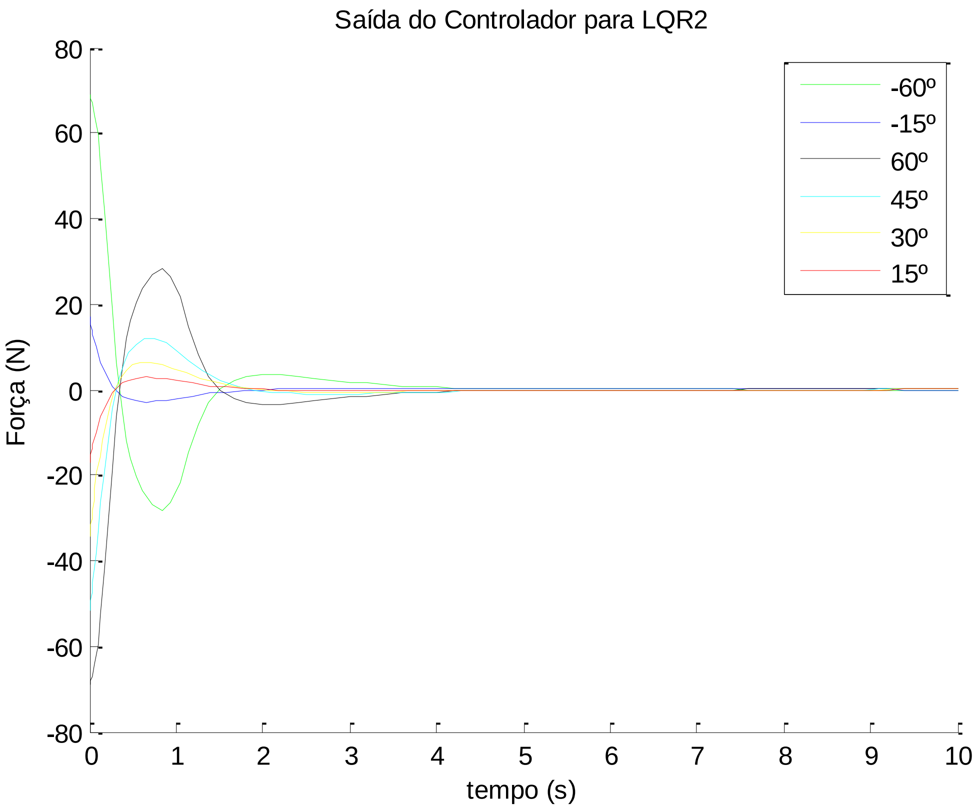


Figura 19- Saída do Controlador para LQR2

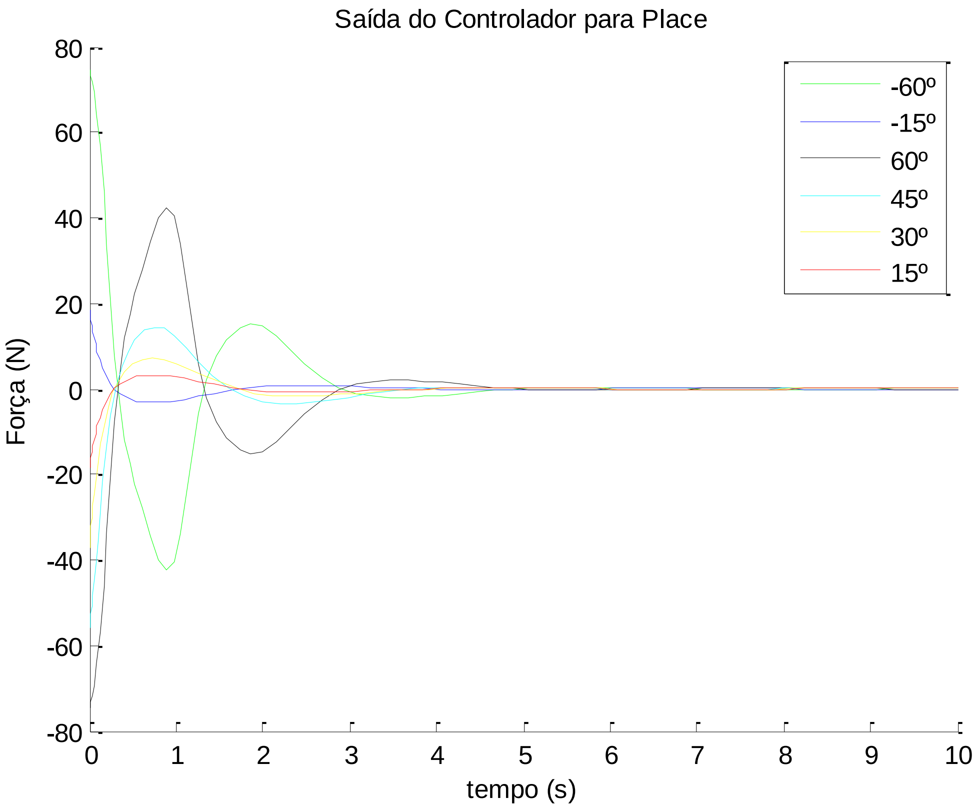


Figura 20- Saída do Controlador para Place

Códigos de Matlab desenvolvidos

EDO’s não lineares com realimentação

```
%função não linear
function zponto = nlinearf(t,z)
      global u M m g len K
      zponto = zeros(size(z));
      u = -K*z;
      c1 = (M+m);  c2 = m*len;  c3 = m*g;  c4 = (M+m)*len;  c5 = (M+m)*g;
      zponto(1) = z(2);
      top2 = u*cos(z(1)) - c5*sin(z(1)) + c2*cos(z(1))*sin(z(1))*z(2)^2;
      zponto(2) = top2/(c2*cos(z(1))^2 - c4);
      zponto(3) = z(4);
      top4 = u + c2*sin(z(1))*z(2)^2 - c3*cos(z(1))*sin(z(1));
      zponto(4) = top4/(c1-m*cos(z(1))^2);
```

EDO’s não lineares sem realimentação

```
%função não linear
function zponto = nlinear(t,z)
      global u M m g len
      zponto = zeros(size(z));
      c1 = (M+m);  c2 = m*len;  c3 = m*g;  c4 = (M+m)*len;  c5 = (M+m)*g;
      zponto(1) = z(2);
      top2 = u*cos(z(1)) - c5*sin(z(1)) + c2*cos(z(1))*sin(z(1))*z(2)^2;
      zponto(2) = top2/(c2*cos(z(1))^2 - c4);
      zponto(3) = z(4);
      top4 = u + c2*sin(z(1))*z(2)^2 - c3*cos(z(1))*sin(z(1));
      zponto(4) = top4/(c1-m*cos(z(1))^2);
```

Código para comparar o sistema linear com o não linear

```
      clear all,   close all,   nfig = 0;

%Sistema Linearizado-Sem realimentação
A=[0 1 0 0;20.601 0 0 0; 0 0 0 1;-0.4905 0 0 0];
B=[0;-1;0;0.5];
C=[1 0 0 0;0 0 1 0];
D=0;
Pl=ss(A,B,C,D);
[yl,tl,zl] = step(Pl,linspace(0,1,31));
%Sistema não linear
global u M m g len          % Constantes
      M = 2.0;   m = 0.1;      % massas
      len = 0.5;               % Comprimento do pêndulo
      g = 9.81;                % aceleração da gravidade
      u = 1;                  % Entrada do sistema
      to = 0;   tf = 1.0;
      zo = [0 0 0 0]';    tol = 1.0e-6;
      options = odeset('RelTol',tol);
      [tnl,znl] = ode45('nlinear',[to tf],zo,options);
```

```
      nfig = nfig+1;    figure(nfig);
      plot(tnl,znl(:,1),'g-',tl,zl(:,1),'go'),grid
      title('Pêndulo invertido theta')
      xlabel('Tempo (s)'),ylabel('Theta (radianos)')
      legend('Não linear','linear')
      nfig = nfig+1;    figure(nfig);
      plot(tnl,znl(:,3),'r-',tl,zl(:,3),'ro'),grid
      title('Pêndulo invertido posição do carrinho')
      xlabel('Tempo (s)'),ylabel('Posição (m)')
      legend('Não linear','linear')
```

Código para o sistema realimentado e cálculo das matrizes K's

```
clear all
close all
% Realimentando o sistema
A=[0 1 0 0;20.601 0 0 0; 0 0 0 1;-0.4905 0 0 0];
B=[0;-1;0;0.5];
%LQR
K1=lqr(A,B,[100 0 0 0;0 1 0 0; 0 0 1 0; 0 0 0 1],100);
Alqr=A-B*K1;
K2=lqr(A,B,[0.1 0 0 0;0 1 0 0; 0 0 1 0; 0 0 0 1],0.1);
Alqr2=A-B*K2;
%Place
p=[-10,-2,-1+1.5*j,-1-1.5*j];
K3=place(A,B,p);
Apl=A-B*K3;
%--------------------------------------------------------------------------
 global u M m g len K  % Constantes
      M = 2.0;   m = 0.1;     % massas
      len = 0.5;              % Comprimento do pêndulo
      g = 9.81;               % aceleração da gravidade
      to = 0;    tf = 10.0;
%       tol = 1.0e-6;
%       options = odeset('RelTol',tol);
%--------------------------------------------------------------------------
%Condições inicias
zo1=[-60*pi/180 0 0 0]';
zo2=[-15*pi/180 0 0 0]';
zo3=[60*pi/180 0 0 0]';
zo4=[45*pi/180 0 0 0]';
zo5=[30*pi/180 0 0 0]';
zo6=[15*pi/180 0 0 0]';
%--------------------------------------------------------------------------
% LQR 1
K=K1;
[tnl1,znl1] = ode45('nlinearf',[to tf],zo1);
[tnl2,znl2] = ode45('nlinearf',[to tf],zo2);
[tnl3,znl3] = ode45('nlinearf',[to tf],zo3);
[tnl4,znl4] = ode45('nlinearf',[to tf],zo4);
[tnl5,znl5] = ode45('nlinearf',[to tf],zo5);
[tnl6,znl6] = ode45('nlinearf',[to tf],zo6);
figure
hold on
plot(tnl1,180/pi*znl1(:,1),'g');
```

```
plot(tnl2,180/pi*znl2(:,1),'b');
plot(tnl3,180/pi*znl3(:,1),'k');
plot(tnl4,180/pi*znl4(:,1),'c');
plot(tnl5,180/pi*znl5(:,1),'y');
plot(tnl6,180/pi*znl6(:,1),'r');
xlabel('tempo [s]')
ylabel('teta [º]')
title('angulo vs tempo para LQR1')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,2),'g');
plot(tnl2,znl2(:,2),'b');
plot(tnl3,znl3(:,2),'k');
plot(tnl4,znl4(:,2),'c');
plot(tnl5,znl5(:,2),'y');
plot(tnl6,znl6(:,2),'r');
xlabel('tempo [s]')
ylabel('teta ponto [º/s]')
title('velocidade angular vs tempo para LQR1')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,3),'g');
plot(tnl2,znl2(:,3),'b');
plot(tnl3,znl3(:,3),'k');
plot(tnl4,znl4(:,3),'c');
plot(tnl5,znl5(:,3),'y');
plot(tnl6,znl6(:,3),'r');
xlabel('tempo [s]')
ylabel('posição [m]')
title('posição vs tempo para LQR1')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,4),'g');
plot(tnl2,znl2(:,4),'b');
plot(tnl3,znl3(:,4),'k');
plot(tnl4,znl4(:,4),'c');
plot(tnl5,znl5(:,4),'y');
plot(tnl6,znl6(:,4),'r');
xlabel('tempo [s]')
ylabel('Velocidade Linear [m/s]')
title('Velocidade Linear vs tempo para LQR1')
legend('-60º','-15º','60º','45º','30º','15º')
%-----------------------------------------------------------------------
%-----------------------------------------------------------------------
% LQR 2
K=K2;
[tnl1,znl1] = ode45('nlinearf',[to tf],zo1);
[tnl2,znl2] = ode45('nlinearf',[to tf],zo2);
[tnl3,znl3] = ode45('nlinearf',[to tf],zo3);
[tnl4,znl4] = ode45('nlinearf',[to tf],zo4);
[tnl5,znl5] = ode45('nlinearf',[to tf],zo5);
[tnl6,znl6] = ode45('nlinearf',[to tf],zo6);
figure
hold on
```

```
plot(tnl1,180/pi*znl1(:,1),'g');
plot(tnl2,180/pi*znl2(:,1),'b');
plot(tnl3,180/pi*znl3(:,1),'k');
plot(tnl4,180/pi*znl4(:,1),'c');
plot(tnl5,180/pi*znl5(:,1),'y');
plot(tnl6,180/pi*znl6(:,1),'r');
xlabel('tempo [s]')
ylabel('teta [º]')
title('angulo vs tempo para LQR2')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,2),'g');
plot(tnl2,znl2(:,2),'b');
plot(tnl3,znl3(:,2),'k');
plot(tnl4,znl4(:,2),'c');
plot(tnl5,znl5(:,2),'y');
plot(tnl6,znl6(:,2),'r');
xlabel('tempo [s]')
ylabel('teta ponto [º/s]')
title('velocidade angular vs tempo para LQR2')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,3),'g');
plot(tnl2,znl2(:,3),'b');
plot(tnl3,znl3(:,3),'k');
plot(tnl4,znl4(:,3),'c');
plot(tnl5,znl5(:,3),'y');
plot(tnl6,znl6(:,3),'r');
xlabel('tempo [s]')
ylabel('posição [m]')
title('posição vs tempo para LQR2')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,4),'g');
plot(tnl2,znl2(:,4),'b');
plot(tnl3,znl3(:,4),'k');
plot(tnl4,znl4(:,4),'c');
plot(tnl5,znl5(:,4),'y');
plot(tnl6,znl6(:,4),'r');
xlabel('tempo [s]')
ylabel('Velocidade Linear [m/s]')
title('Velocidade Linear vs tempo para LQR2')
legend('-60º','-15º','60º','45º','30º','15º')
%-----------------------------------------------------------------------
% Place
K=K3;
[tnl1,znl1] = ode45('nlinearf',[to tf],zo1);
[tnl2,znl2] = ode45('nlinearf',[to tf],zo2);
[tnl3,znl3] = ode45('nlinearf',[to tf],zo3);
[tnl4,znl4] = ode45('nlinearf',[to tf],zo4);
[tnl5,znl5] = ode45('nlinearf',[to tf],zo5);
[tnl6,znl6] = ode45('nlinearf',[to tf],zo6);
figure
hold on
```

```
plot(tnl1,180/pi*znl1(:,1),'g');
plot(tnl2,180/pi*znl2(:,1),'b');
plot(tnl3,180/pi*znl3(:,1),'k');
plot(tnl4,180/pi*znl4(:,1),'c');
plot(tnl5,180/pi*znl5(:,1),'y');
plot(tnl6,180/pi*znl6(:,1),'r');
xlabel('tempo [s]')
ylabel('teta [º]')
title('angulo vs tempo para Place')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,2),'g');
plot(tnl2,znl2(:,2),'b');
plot(tnl3,znl3(:,2),'k');
plot(tnl4,znl4(:,2),'c');
plot(tnl5,znl5(:,2),'y');
plot(tnl6,znl6(:,2),'r');
xlabel('tempo [s]')
ylabel('teta ponto [º/s]')
title('velocidade angular vs tempo para Place')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,3),'g');
plot(tnl2,znl2(:,3),'b');
plot(tnl3,znl3(:,3),'k');
plot(tnl4,znl4(:,3),'c');
plot(tnl5,znl5(:,3),'y');
plot(tnl6,znl6(:,3),'r');
xlabel('tempo [s]')
ylabel('posição [m]')
title('posição vs tempo para Place')
legend('-60º','-15º','60º','45º','30º','15º')
figure
hold on
plot(tnl1,znl1(:,4),'g');
plot(tnl2,znl2(:,4),'b');
plot(tnl3,znl3(:,4),'k');
plot(tnl4,znl4(:,4),'c');
plot(tnl5,znl5(:,4),'y');
plot(tnl6,znl6(:,4),'r');
xlabel('tempo [s]')
ylabel('Velocidade Linear [m/s]')
title('Velocidade Linear vs tempo para Place')
legend('-60º','-15º','60º','45º','30º','15º')
```