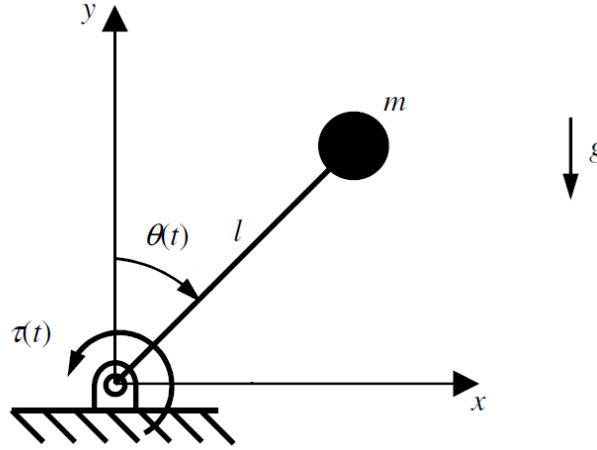


# Controle Fuzzy TS de um Pêndulo Invertido

(Modelagem por Aproximação Local)

**Exemplo:** Considere o modelo do pêndulo invertido apresentado na figura a seguir. A massa do pêndulo que é concentrada em sua ponta é  $m$  e o seu comprimento é  $l$ . Um torque  $\tau(t)$  variável é aplicado no pêndulo na posição de sua articulação.



A dinâmica do pêndulo é dada por:

$$ml^2\ddot{\theta}(t) + mgl\sin(\theta(t)) = \tau(t)$$

O sistema na forma de espaço de estados fica:

$$\begin{cases} \dot{\theta}(t) = \omega(t) & = f_1(\theta, \omega, \tau) \\ \dot{\omega}(t) = \frac{-g}{l}\sin(\theta(t)) + \frac{\tau(t)}{ml^2} & = f_2(\theta, \omega, \tau) \end{cases}$$

Iremos considerar dois modelos locais para a simulação do sistema controlado. Inicialmente calcula-se o modelo local no ponto de equilíbrio (será o **Modelo Local 1**) que neste caso pode ser visto como o principal modelo de projeto, já que o objetivo de controle é manter o pêndulo na posição vertical. Em seguida iremos encontrar um modelo local (**Modelo Local 2a**) que será calculado com base na série de Taylor, entretanto será desprezado o termo afim da função (este caso será considerado apenas como análise, pois não é um modelo local “confiável”, já que a função linearizada, sem o termo afim, não passa pelo ponto de linearização considerado). Em seguida será considerado um modelo local (**Modelo Local 2b**) que é baseado numa escolha mais adequada para o projeto dos controladores.

### – Modelo Local 1:

Considere, neste caso, o ponto de equilíbrio  $\dot{\theta}_0 = 0$  e  $\dot{\omega}_0 = 0$

$$\begin{cases} \dot{\theta}_0 = \omega_0 & = 0 \\ \dot{\omega}_0 = \frac{-g}{l} \text{sen}(\theta_0) + \frac{\tau_0}{ml^2} & = 0 \end{cases}$$

$\Rightarrow$  assim:  $\theta_0 = 0^\circ$  e  $\tau_0 = 0$ .

### **Dinâmica Linearizada**

Utilizando o conceito de linearização baseada na série de Taylor podemos escrever:

$$\left(\frac{\partial f_1}{\partial \theta}\right)_0 = 0 \qquad \left(\frac{\partial f_1}{\partial \omega}\right)_0 = 1 \qquad \left(\frac{\partial f_1}{\partial \tau}\right)_0 = 0$$

$$\left(\frac{\partial f_2}{\partial \theta}\right)_0 = -\frac{g}{l} \cos(0^\circ) = -\frac{g}{l} \qquad \left(\frac{\partial f_2}{\partial \omega}\right)_0 = 0 \qquad \left(\frac{\partial f_2}{\partial \tau}\right)_0 = \frac{1}{ml^2}$$

Portanto:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} \qquad \mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$$

$$\begin{bmatrix} \delta \dot{\theta}(t) \\ \delta \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} \begin{bmatrix} \delta \theta(t) \\ \delta \omega(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} \delta \tau(t)$$

### **Modelo Local 2a:**

Neste caso, foi considerada (escolhido) uma inclinação de  $60^\circ$  em relação à vertical. Como calculado anteriormente, considerando agora que a condição de linearização não é um ponto de equilíbrio, ou seja,  $\ddot{\theta}_0 \neq 0 \Rightarrow$  assim:

$$\begin{cases} \dot{\theta}_0 = \omega_0 \\ \dot{\omega}_0 = \frac{-g}{l} \text{sen}(60^\circ) + \frac{\tau_0}{ml^2} \end{cases}$$

Neste caso, para definir a condição de linearização, deve-se adotar os parâmetros  $\omega_0$  e  $\dot{\omega}_0$ , ou  $\omega_0$  e  $\tau_0$ .

$$\Rightarrow \text{Adotando } \omega_0 = 0 \quad \tau_0 = 0 \quad \Rightarrow \dot{\omega}_0 = -\frac{g}{l} \text{sen}(\theta_0) = -\frac{\sqrt{3}g}{2l}.$$

$\Rightarrow$  Observa-se que nessa condição de linearização o pêndulo está parado, mas iniciando o movimento de queda devido à aceleração da gravidade.

**Dinâmica Linearizada em torno de  $60^\circ$  (Desconsiderando o termo afim):**

$$\left( \frac{\partial f_1}{\partial \theta} \right)_0 = 0 \qquad \left( \frac{\partial f_1}{\partial \omega} \right)_0 = 1 \qquad \left( \frac{\partial f_1}{\partial \tau} \right)_0 = 0$$

$$\left( \frac{\partial f_2}{\partial \theta} \right)_0 = -\frac{g}{l} \cos(60^\circ) \qquad \left( \frac{\partial f_2}{\partial \omega} \right)_0 = 0 \qquad \left( \frac{\partial f_2}{\partial \tau} \right)_0 = \frac{1}{ml^2}$$

Portanto, desconsiderando o termo afim neste caso tem-se:

$$\mathbf{A}_{2a} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{2l} & 0 \end{bmatrix} \qquad \mathbf{B}_{2a} = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{ml^2} \end{bmatrix}$$
$$\begin{bmatrix} \delta \dot{\theta}(t) \\ \delta \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{2l} & 0 \end{bmatrix} \begin{bmatrix} \delta \theta(t) \\ \delta \omega(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \frac{1}{ml^2} \end{bmatrix} \delta \tau(t)$$

## Observação

Considere a função

$$h(\theta) = \frac{-g}{l} \text{sen}(\theta(t))$$

Utilizando os primeiros termos da série de Taylor tem-se,

$$h(\theta) \cong h(\theta_0) + \left( \frac{\partial h(\theta)}{\partial \theta} \right)_{\theta_0} (\theta - \theta_0) + \dots$$

No ponto de interesse ( $\theta_0 = 60^\circ = \frac{\pi}{3}$  rad) obtém-se:

$$h(\theta) \cong h(\theta_0) + \frac{\partial h(\theta)}{\partial \theta} \Big|_{\theta_0} (\theta - \theta_0) + \dots = \frac{-g}{l} \text{sen} \left( \frac{\pi}{3} \right) + \frac{-g}{l} \cos \left( \frac{\pi}{3} \right) \left( \theta - \frac{\pi}{3} \right) + \dots$$

Na figura a seguir pode-se observar a função original  $h(\theta)$ , a função com o termo afim e a função linear sem o termo afim.

Foi considerado  $g = 9,8 \text{ m/s}^2$  e  $l = 1 \text{ m}$ .

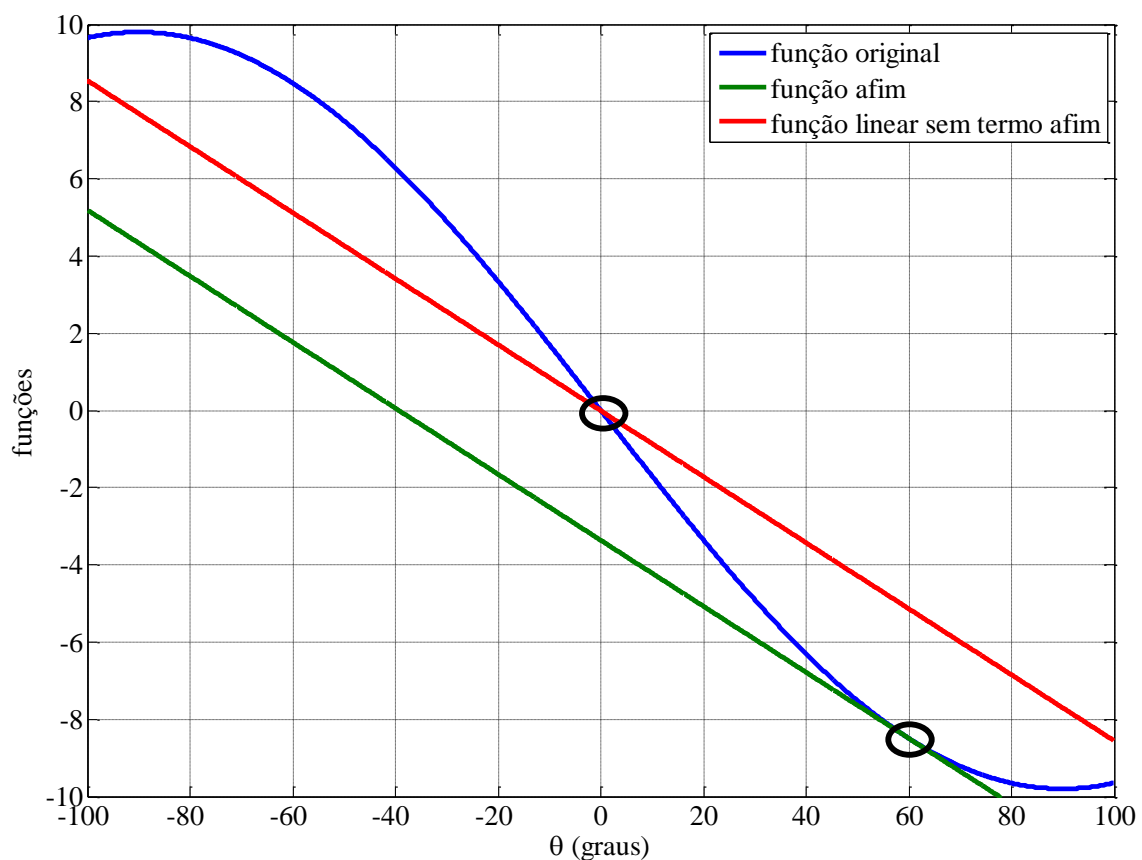


Figura: Comparação entre as funções.

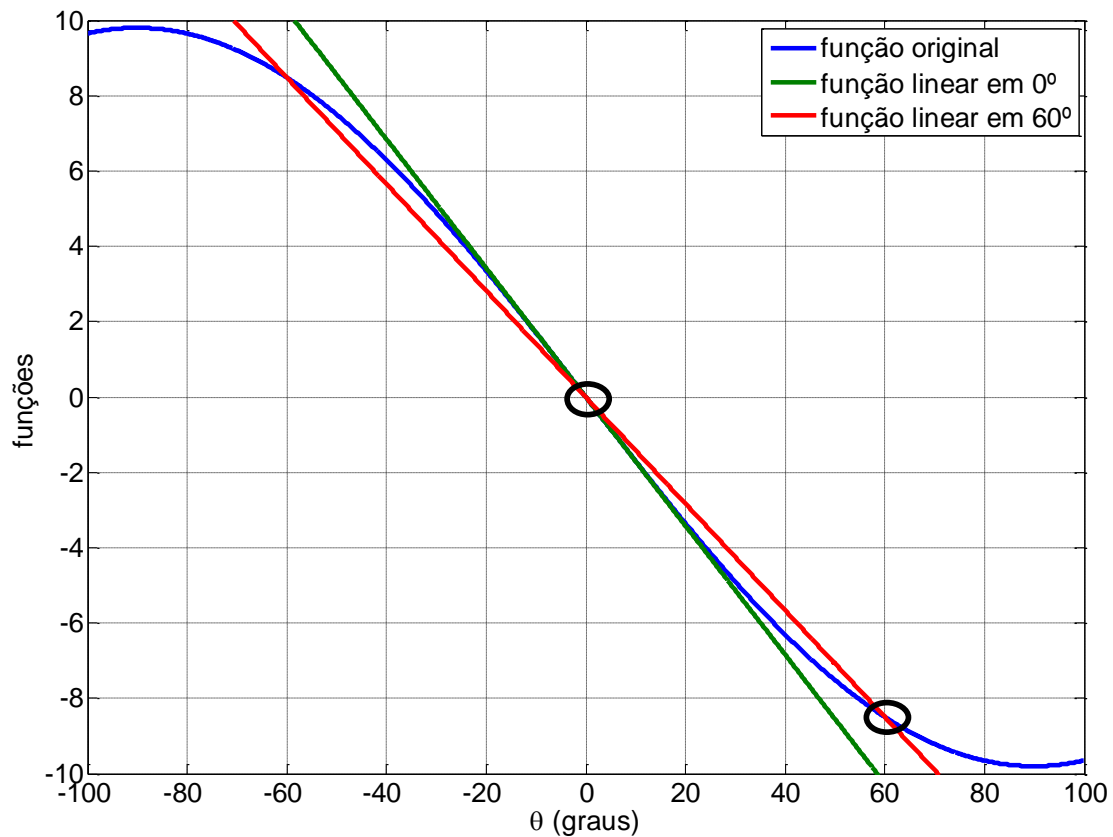
### Modelo Local 2b:

Neste caso iremos considerar o Modelo Local,

$$\mathbf{A}_{2b} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l} \sin(\frac{\pi}{3}) & 0 \end{bmatrix} \quad \mathbf{B}_{2b} = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$$

$$\begin{bmatrix} \delta \dot{\theta}(t) \\ \delta \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l} \sin(\frac{\pi}{3}) & 0 \end{bmatrix} \begin{bmatrix} \delta \theta(t) \\ \delta \omega(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} \delta \tau(t)$$

Na figura a seguir pode-se observar que neste caso a função  $h(\theta)$  linearizada em 60 graus, passa pelo ponto considerado (60 graus). Desta forma, o Modelo Local 2b é considerado mais apropriado para a modelagem fuzzy Takagi-Sugeno.



## Função de Pertinência

Considere, por exemplo, as seguintes funções para o sistema Fuzzy TS.

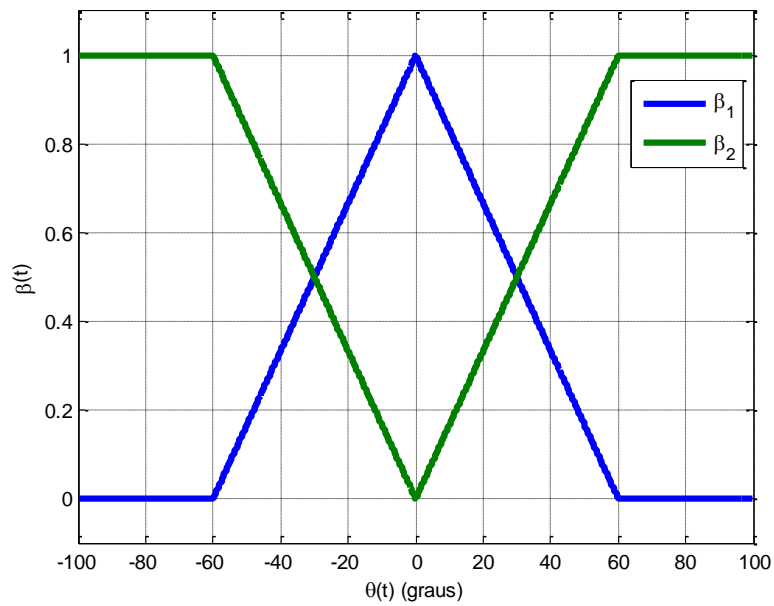


Figura: Funções de pertinência.

Na figura a seguir são apresentadas as aproximações com a função original.

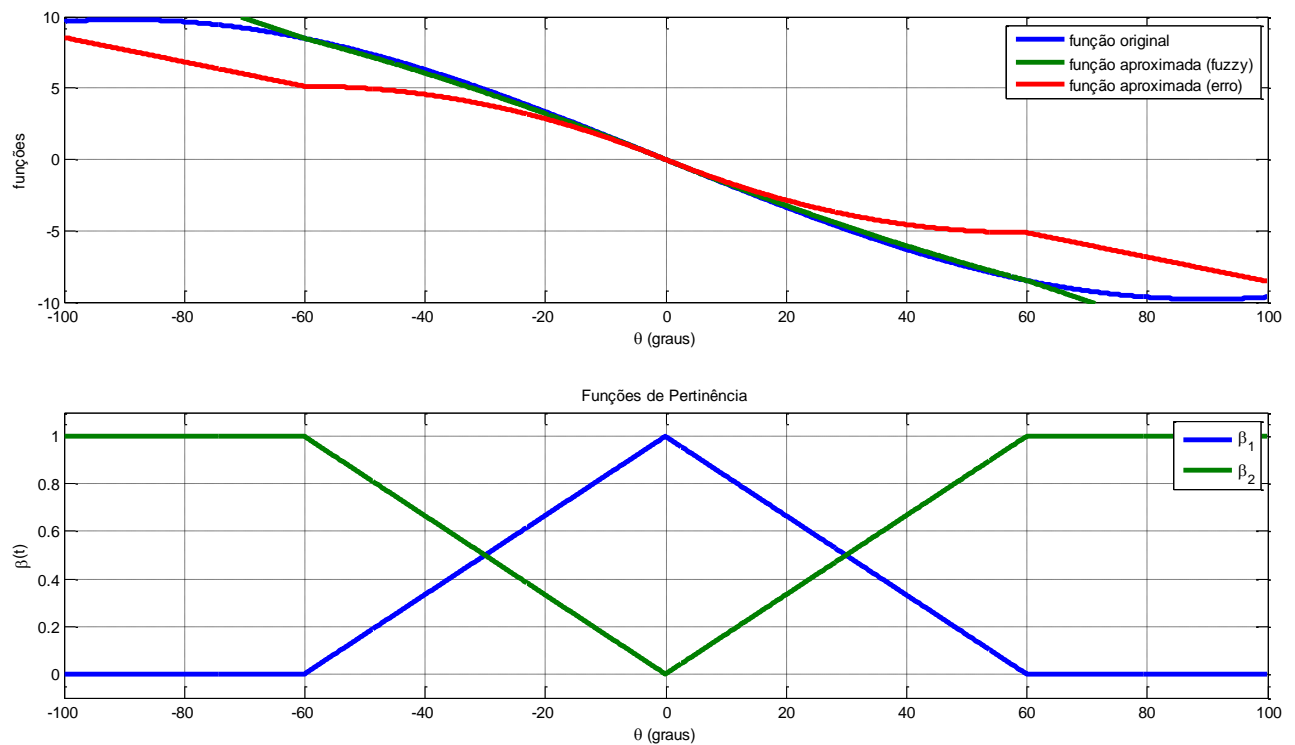


Figura: Funções aproximadas pela combinação convexa.

Pela escolha anterior pode-se verificar uma boa aproximação com a função original. Entretanto, existem outras opções que poderiam apresentar ainda melhores combinações entre os modelos locais. Por exemplo, observe o resultado para as funções a seguir (que serão utilizadas nas simulações).

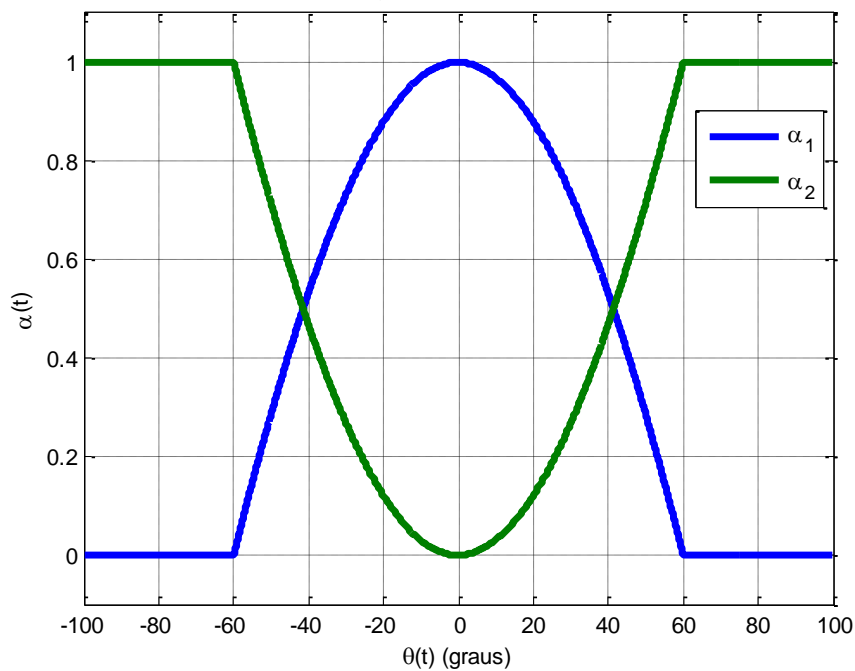


Figura: Funções de pertinência.

A figura a seguir mostra as aproximações com a função original.

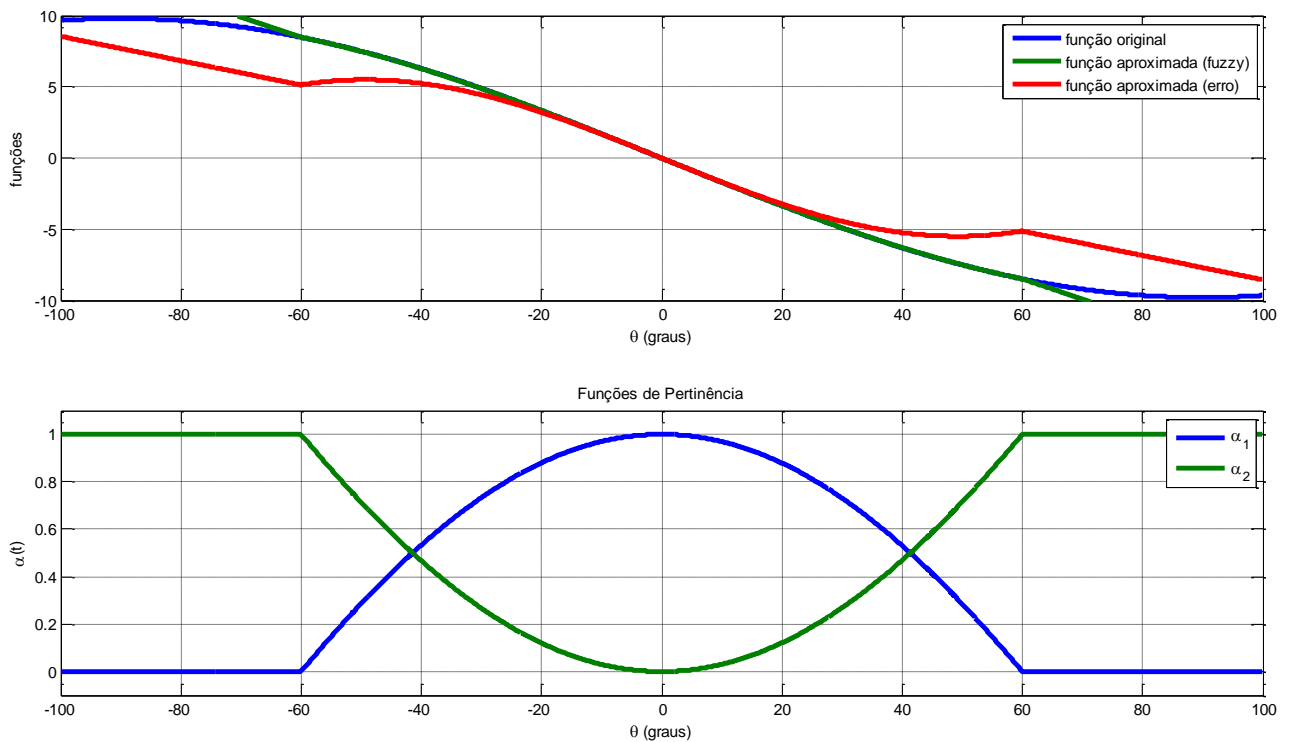


Figura: Funções aproximadas pela combinação convexa.

## Simulações

Parâmetros para o sistema:  $m = 0,5 \text{ kg}$ ,  $l = 1 \text{ m}$ ,  $g = 9,8 \text{ m/s}^2$ .

Linearização - uso do comando **Linmod** no Matlab.

```
>> help linmod
linmod Obtains linear models from systems of ord. diff. equations (ODEs).
[A,B,C,D]=linmod('SYS') obtains the state-space linear model of the
system of ordinary differential equations described in the
block diagram 'SYS' when the state variables and inputs are set
to the defaults specified in the block diagram.

[A,B,C,D]=linmod('SYS',X,U) allows the state vector, X, and
input, U, to be specified. A linear model will then be obtained
at this operating point.
```

Para obter o sistema linearizado com o comando **linmod**, pode-se utilizar a planta (não linear) no Simulink, conforme a figura a seguir. Entretanto, é importante verificar que neste caso o termo afim é desconsiderado.

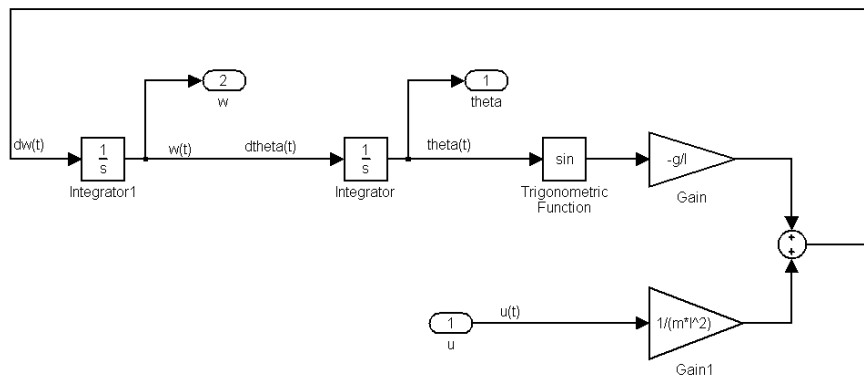


Figura: Planta – Sistema Não Linear.

O sistema realimentado fica da seguinte forma:

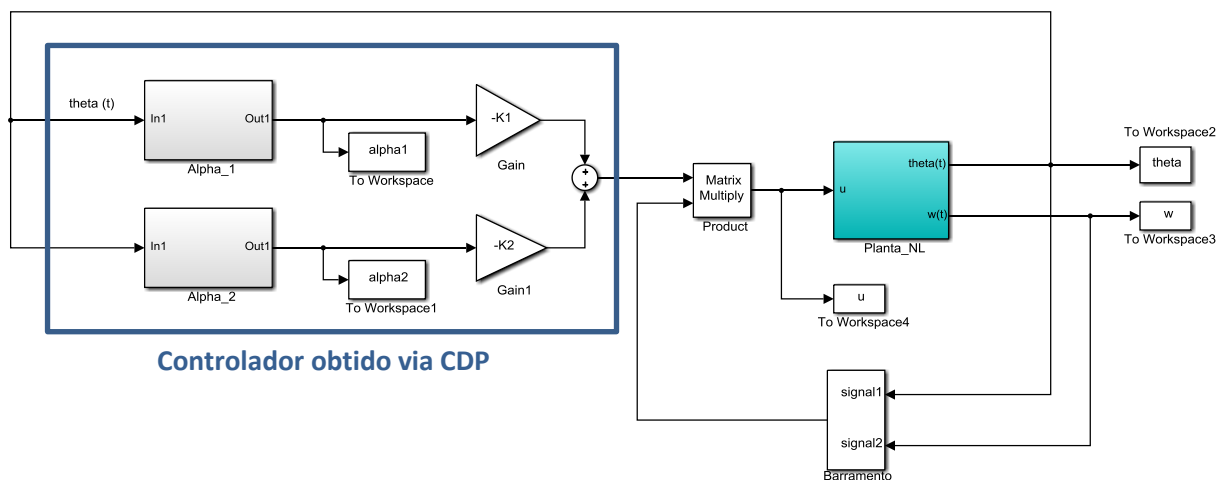


Figura: Sistema realimentado – Controle fuzzy TS.



- **Simulação 1 (Modelos Locais 1 e 2b):**

Controladores obtidos pelas LMIs (restrição de estabilidade):

$$K_1 = [-4,2911 \quad 0,3540] \quad K_{2b} = [-3,4433 \quad 0,3540].$$

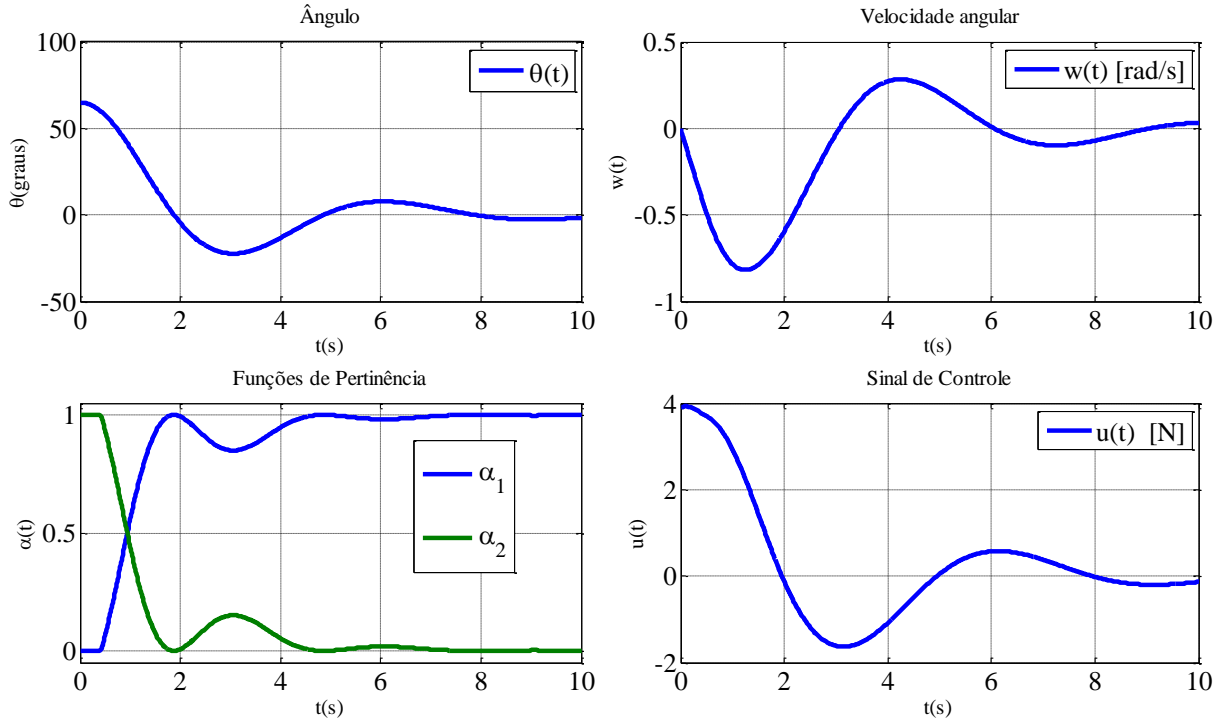


Figura: Simulação 1 – Controle Fuzzy TS do sistema Pêndulo Invertido, com  $x_0 = [65^\circ \quad 0]'$ .

Observe que na simulação, a função de pertinência  $\alpha_2(t)$  inicialmente é  $\alpha_2(t) = 1$ , e a função  $\alpha_1(t) = 0$ , pois o ângulo inicial é maior que  $60^\circ$ . Desta forma, de acordo com a função de pertinência escolhida anteriormente, o controlador ativado inicialmente é o controlador  $K_2$ , projetado através do modelo local composto por  $A_2, B_2$ . Note que no decorrer do tempo a função muda, ponderando os controladores  $K_1$  e  $K_2$  através da combinação convexa entre  $\alpha_1(t)$  e  $\alpha_2(t)$ . No regime, o sistema atinge o ponto de equilíbrio (origem), que linearmente é representado pelo modelo local composto por  $A_1, B_1$ .

- **Simulação 2 (Modelos Locais 1 e 2b):**

Controladores obtidos pelas LMIs (restrição de estabilidade):

$$K_1 = [-4,2911 \quad 0,3540] \quad K_{2b} = [-3,4433 \quad 0,3540].$$

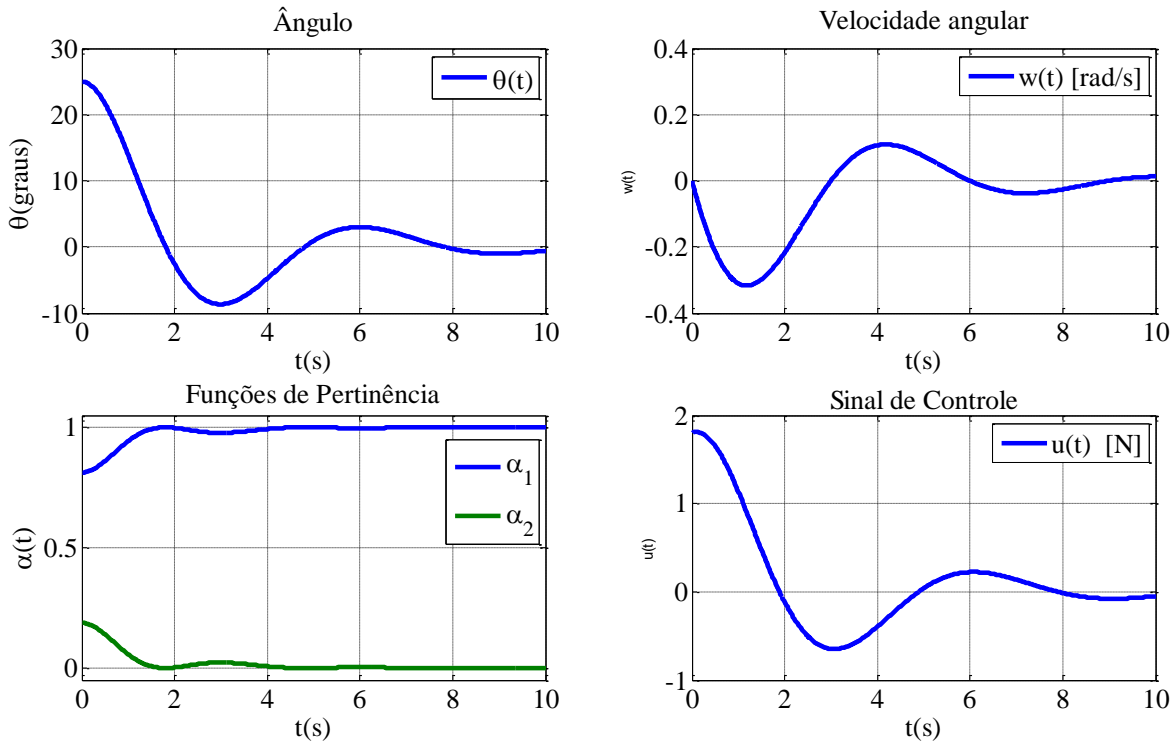


Figura: Simulação 2 – Controle Fuzzy TS do sistema Pêndulo Invertido, com  $x_0 = [25^\circ \quad 0]'$ .

Uma observação interessante é que não é necessário que todas as combinações  $(A_i - B_i K_j)$  sejam *Hurwitz*. Neste caso, por exemplo, temos:

$$>> \text{eig}(A_1 - B_1 K_1) = [-0.3540 + 1.0452i \quad -0.3540 - 1.0452i]'$$

$$>> \text{eig}(A_1 - B_1 K_2) = [-0.3540 + 1.6697i \quad -0.3540 - 1.6697i]'$$

$$>> \text{eig}(A_2 - B_2 K_2) = [-0.3540 + 1.0452i \quad -0.3540 - 1.0452i]'$$

$$>> \text{eig}(A_2 - B_2 K_1) = [0.4225 \quad -1.1305]' \text{ (Não Hurwitz)}$$

Neste exemplo, se fosse utilizado um controlador linear (único) projetado através do modelo linearizado na origem, o sistema poderia ficar instável para condições iniciais mais afastadas da origem. Veja a seguir uma simulação para este caso, com inicial  $x_0 = [51^\circ \quad 0]^T$ . Condição em que foi possível controlar o sistema com controle fuzzy TS, mas não foi possível com um ganho único.

- **Simulação 3** (ganho único):

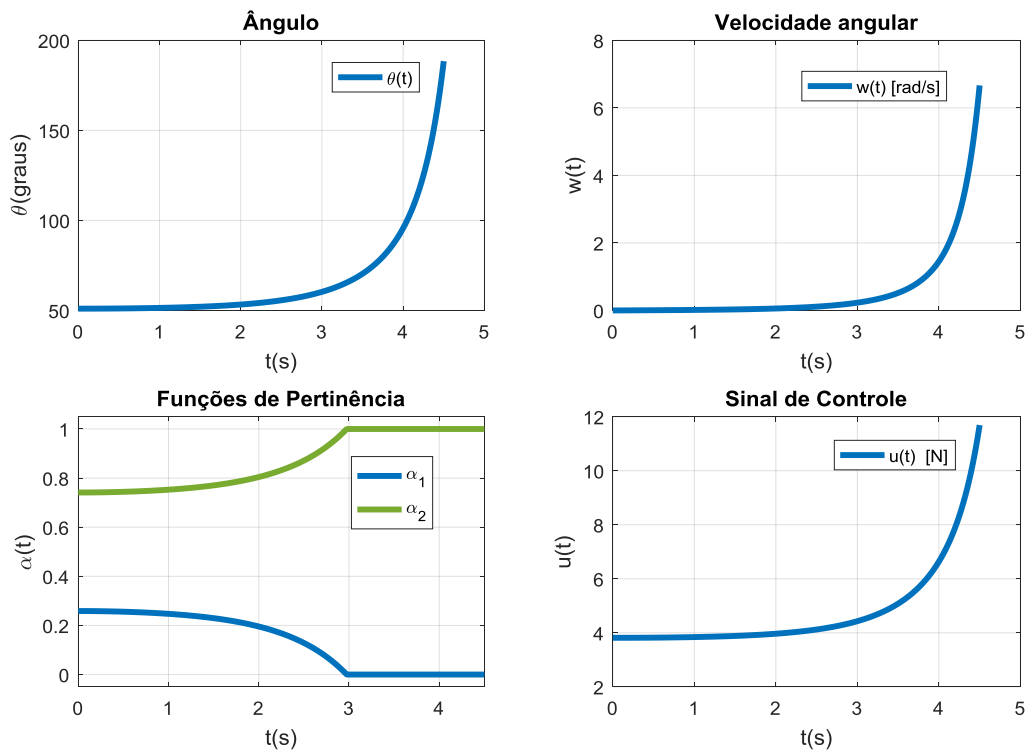


Figura: Simulação 3 – Com ganho único  $u(t) = -K_1 x(t)$ ,  $K_1 = [-4,2911 \quad 0,3540]$  e  $x_0 = [51^\circ \quad 0]'$ .

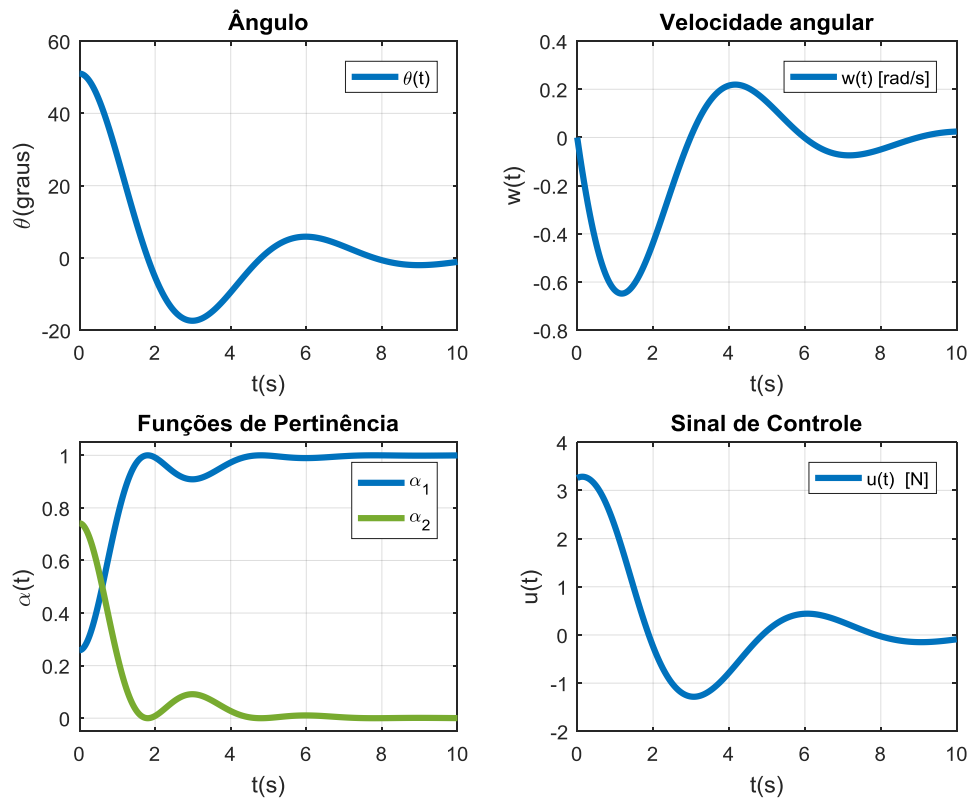


Figura: Simulação 4 – Controle Fuzzy TS do sistema Pêndulo Invertido, com  $x_0 = [51^\circ \quad 0]'$ .

## Código Matlab utilizado:

```
% Simulação de Sistema de Controle Fuzzy TS para um Pêndulo Invertido
% Nesta simulação a linearização para obter o Modelo Local 2a foi obtida
% desprezando o termo afim, por considerar o ponto de linearização como
% sendo a origem (o eixo deslocado). Neste caso, o modelo Local 2b obtido é mais
% adequado para o projeto dos controladores.

%Parâmetros
g=9.8;
m=0.5;
l=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Modelo Local 1 (em torno da origem)
A1=[0 1; -g/l 0];
B1=[0; 1/(m*l^2)];
%[A1, B1, C1, D1]=linmod('plantaNL')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Modelo Local 2a (em torno de theta0=60° w0=0 e u0=0)
A2a=[0 1; -g/(2*l) 0];
B2a=[0; 1/(m*l^2)];
%[A2a, B2a, C2a, D2a]=linmod('plantaNL',[pi/3 0]',0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Modelo Local 2b (em torno de theta0=60°)
A2b=[0 1; -g/l*sin(pi/3)/(pi/3) 0];
B2b=[0; 1/(m*l^2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Projeto dos Controladores via LMIs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Criando a matriz variável X_2x2 simétrica ( Lembrando que X=inv(P) )
% e as matrizes M1 e M2
X = sdpvar(2,2);
M1= sdpvar(1,2);
M2= sdpvar(1,2);

% Criando as restrições LMI do problema
%-----
LMIs = [X > 0,...;
        -X*A1'+M1'*B1'-A1*X+B1*M1 > 0,...
        -X*A2b'+M2'*B2b'-A2b*X+B2b*M2 > 0,...
        -X*A1'+M2'*B1'-X*A2b'+M1'*B2b'-A1*X+B1*M2-A2b*X+B2b*M1 > 0];
%-----

% Configurando o Solver.
opts=sdpsettings('solver','lmilab');
% Resolvendo as LMIs e encontrando uma solução factível para P
sol = optimize(LMIs,[],opts);
% Verificando se as soluções encontradas pelo solvesdp satisfazem as
% restrições LMI
[p,d]=checkset(LMIs);
checkset(LMIs)
if sum(p < 0) == 0
    disp('Factível')
    X_f = value(X);
    M1_f = value(M1);
    M2_f = value(M2);

    F1=M1_f*inv(X_f);
```

```

F2=M2_f*inv(X_f);

disp('Verificação das Restrições inseridas:')
eig(X_f)
eig(-X_f*A1'+M1_f'*B1'-A1*X_f+B1*M1_f)
eig(-X_f*A2b'+M2_f'*B2b'-A2b*X_f+B2b*M2_f)
eig(-X_f*A1'+M2_f'*B1'-X_f*A2b'+M1_f'*B2b'-A1*X_f+B1*M2_f-A2b*X_f+B2b*M1_f)
else
    disp('Não Encontrou Solução')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Condição Inicial
IC=[25*pi/180 0];

%tempo de simulação (s)
t=0:0.01:10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SIMULAÇÃO - Projeto com Modelos Locais 1 e 2b
K1=F1;
K2=F2;
sim('sistema',t)

figure(3)
subplot(2,2,1);
plot(t,180/pi*theta(:,1),'-','LineWidth',3)
xlabel('t(s)')
ylabel('\theta(graus)')
title('Ângulo')
legend('\theta(t)'); grid;

subplot(2,2,2);
plot(t,w(:,1),'-','LineWidth',3)
xlabel('t(s)')
ylabel('w(t)')
title('Velocidade angular')
legend('w(t) [rad/s]'); grid;

subplot(2,2,3);
plot(t,alpha1,t,alpha2,'-','LineWidth',3)
xlabel('t(s)')
ylabel('\alpha(t)')
title('Funções de Pertinência')
legend('\alpha_1','\alpha_2'); grid;
axis([0 max(t) -0.05 1.05])

subplot(2,2,4);
plot(t,u,'-','LineWidth',3)
xlabel('t(s)')
ylabel('u(t)')
title('Sinal de Controle')
legend('u(t) [N]'); grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

## Resultado - Verificação dos autovalores das LMIs inseridas

```
disp('Verificação das Restrições inseridas:')
eig(X_f)
eig(-X_f*A1'+M1_f'*B1'-A1*X_f+B1*M1_f)
eig(-X_f*A2b'+M2_f'*B2b'-A2b*X_f+B2b*M2_f)
eig(-X_f*A1'+M2_f'*B1'-X_f*A2b'+M1_f'*B2b'-A1*X_f+B1*M2_f-A2b*X_f+B2b*M1_f)
```

Verificação das Restrições inseridas:

ans =

```
35.5625
67.1736
```

ans =

```
31.6111
34.2454
```

ans =

```
31.6111
34.2454
```

ans =

```
63.2222
68.4907
```