macrφ@ufmg
MECHATRONICS, CONTROL, AND ROBOTICS

MACRO/RAFFO–2021/JMC/1+Versão-1

Universidade Federal de Minas Gerais - UFMG

Programa de Pós-Graduação em Engenharia Elétrica

Parallel Distributed Compensator(PDC) de um Pendulo Invertido Via Representação Takagi-Sugeno

Número de matrícula: 2016086496
Estudante: Jonatan Mota Campos
Orientador: Guilherme Vianna Raffo
Co-orientador:

23 de março de 2021

Sumário

1 Modelagem Dinâmica

1.1 Formulação Newton Euler e Representação Via Sistema Descritor

Pela modelagem de Newton Euler, que não será demonstrada aqui por ja ter sido demonstrada no arquivo fornecido para o trabalho obtêm-se o modelo dinâmico abaixo:

$$\begin{bmatrix} M+m & -mlcos(\theta) \\ -mlcos(\theta) & ml^2 \end{bmatrix} \begin{bmatrix} \ddot{s} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & mlsin(\theta) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} (M+m)gsin(\alpha) \\ -mglsin(\alpha+\theta) \end{bmatrix} = \begin{bmatrix} u - k_v \dot{s} \\ 0 \end{bmatrix} \tag{1}$$

Onde, M é a massa do carrinho, m é a massa do pendulo, g é a aceleração da gravidade, l é a distancia entre o centro de gravidade do pendulo e seu eixo de rotação, s é a distância percorrida pelo carrinho, θ é a incliinação do pendulo, e k_v é uma constante de atrito viscoso.

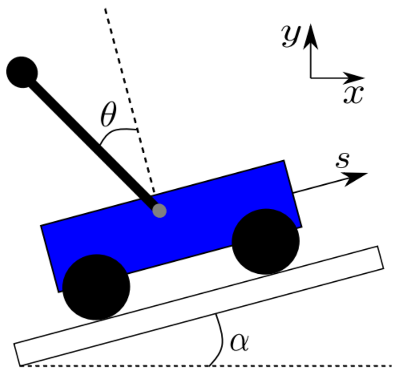


Figura 1: Pêndulo Invertido

Definindo ainda $v = \dot{s}$ e $\omega = \dot{\theta}$ escreve-se a dinâmica em malha aberta como,

$$\begin{aligned} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & M+m & -mlcos(\theta) \\ 0 & 0 & -mlcos(\theta) & ml^2 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -k_v & -mlsin(\theta)\omega \\ 0 & mglcos(\alpha)sinc(\theta) & 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ \theta \\ v \\ \omega \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ -(M+m)gsinc(\alpha) \\ mglcos(\theta)sinc(\alpha) \end{bmatrix} \alpha \end{aligned} \tag{2}$$

Fazendo $\bar{\alpha} = sinc(\alpha)\alpha$ escreve-se o sistema descritor quasi-LPV como,

$$\boldsymbol{E}(x)\dot{\boldsymbol{x}} = \boldsymbol{A}(x)\boldsymbol{x} + \boldsymbol{B_u}u + \boldsymbol{B_\alpha}(x)\bar{\alpha} \tag{3}$$

$$\begin{aligned} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & M+m & -mlcos(\theta) \\ 0 & 0 & -mlcos(\theta) & ml^2 \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -k_v & -mlsin(\theta)\omega \\ 0 & mglcos(\alpha)sinc(\theta) & 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ \theta \\ v \\ \omega \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ -(M+m)g \\ mglcos(\theta) \end{bmatrix} \bar{\alpha} \end{aligned} \tag{4}$$

1.2 Representação Takagi-Sugeno Via Não Linearidade de Setor

Nessa seção será utilizada a técnica de não linearidade de setor para buscar um modelo Takagi-Sugeno exato dentro de um domínio compacto. São escolhidas as seguintes não linearidades:

$$\begin{cases} z_1 = cos(\theta) \\ z_2 = sin(\theta)\omega \\ z_3 = cos(\alpha)sinc(\theta) \end{cases} \tag{5}$$

Dessa forma obtemos um modelo dinâmico na forma,

$$\dot{\boldsymbol{x}} = \left\{ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & M+m & -mlz_1(x) \\ 0 & 0 & -mlz_1(x) & ml^2 \end{bmatrix} \right\}^{-1} \left\{ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -k_v & -mlz_2(x) \\ 0 & mglz_3(x,\alpha) & 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ \theta \\ v \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ -(M+m)g \\ mglz_1(x) \end{bmatrix} \bar{\alpha} \right\} \tag{6}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \hat{\boldsymbol{\alpha}}) \tag{7}$$

Como $\dot{\boldsymbol{x}} = \boldsymbol{f}(\mathbf{0},\mathbf{0},\mathbf{0}) = 0$ podemos utilizar a técnica da não linearidade de setor. Tomando a condição de que estamos projetando um controlador para atuar em um terreno com inclinação máxima e minima conhecidas de forma que $\alpha \in$ [-10°,10°], além de que da geometria do problema $\theta \in$ [-80°,80°], e assumindo que a velocidade angular do pendulo varie como $\omega \in$ [-$\frac{pi}{9}(\frac{rad}{s})$,$\frac{pi}{9}(\frac{rad}{s})$].

$$a_1^{max} = \max z_1 = 1 \tag{8}$$
$$a_1^{min} = \min z_1 = 0.1736 \tag{9}$$
$$\tag{10}$$
$$a_2^{max} = \max z_2 = 0.3438 \tag{11}$$
$$a_2^{min} = \min z_2 = -0.3431 \tag{12}$$
$$\tag{13}$$
$$a_3^{max} = \max z_3 = 1 \tag{14}$$
$$a_3^{min} = \max z_3 = 0.6946 \tag{15}$$
$$\tag{16}$$

Podemos escrever as funções de pertinência utilizando a ideia de pertencimento a uma reta ligando os extremos das não linearidades de forma que $z_i \in [a_i^{\min}, a_i^{\max}]$,

$$z_i(x) = \omega_i^{\min}(x)a_i^{\min} + \omega_i^{\max}(x)a_i^{\max} \tag{17}$$
$$z_i(x) = (1-\omega_i^{\max}(x))a_i^{\min} + \omega_i^{\max}(x)a_i^{\max} \tag{18}$$
$$\boxed{\omega_i^{\max}(x) = \frac{z_i(x) - a_i^{\min}}{a_i^{\max} - a_i^{\min}}, \forall i = 1,2,3} \tag{19}$$

Onde $\omega_i^{\min}(x) = 1 - \omega_i^{\max}(x)$ de forma que definimos as funções de pertencimento das não linearidades como,

$$\omega_1^{\max}(x) = \frac{cos(\theta)-1}{1-0.1736} \tag{20}$$
$$\omega_1^{\min}(x) = 1 - \frac{cos(\theta)-1}{1-0.1736} \tag{21}$$
$$\tag{22}$$
$$\omega_2^{max}(x) = \frac{sin(\theta)\omega + 0.3438}{0.3438+0.3438} \tag{23}$$
$$\omega_2^{min}(x) = 1 - \frac{sin(\theta)\omega + 0.3438}{0.3438+0.3438} \tag{24}$$
$$\tag{25}$$
$$\omega_3^{max}(x) = \frac{cos(\alpha)sinc(\theta) - 0.6946}{1-0.6946} \tag{26}$$
$$\omega_3^{min}(x) = 1 - \frac{cos(\alpha)sinc(\theta) - 0.6946}{1-0.6946} \tag{27}$$
$$\tag{28}$$

É possível notar que com 3 não linearidade teremos 8 vértice/regras (2^3) construir então a pertinência da regra como o produtório das funções de pertencimento de cada vértice,

Regra 1: $z_1 = a_1^{max}, z_2 = a_2^{max}, z_3 = a_3^{max}$, $\eta_1(x) = \omega_1^{\max}(x)\omega_2^{\max}(x)\omega_3^{\max}(x)$

Então,

$$\boldsymbol{E}_1(\boldsymbol{z})\dot{x} = \boldsymbol{A}_1(\boldsymbol{z})\boldsymbol{x} + \boldsymbol{B_u}u + \boldsymbol{B_{\alpha_1}}(\boldsymbol{z})\bar{\alpha} \tag{29}$$

Regra 2: $z_1 = a_1^{\max}, z_2 = a_2^{\max}, z_3 = a_3^{\min}$, $\eta_2(x) = \omega_1^{\max}(x)\omega_2^{\max}(x)\omega_3^{\min}(x)$

Então,

$$\boldsymbol{E}_2(\boldsymbol{z})\dot{x} = \boldsymbol{A}_2(\boldsymbol{z})\boldsymbol{x} + \boldsymbol{B_u}u + \boldsymbol{B_{\alpha_2}}(\boldsymbol{z})\bar{\alpha} \tag{30}$$

$$\vdots \tag{31}$$

Regra 8: $z_1 = a_1^{\min}, z_2 = a_2^{\min}, z_3 = a_3^{\min}$, $\eta_8(x) = \omega_1^{\min}(x)\omega_2^{\min}(x)\omega_3^{\min}(x)$

Então,

$$\boldsymbol{E}_8(\boldsymbol{z})\dot{x} = \boldsymbol{A}_8(\boldsymbol{z})\boldsymbol{x} + \boldsymbol{B_u}u + \boldsymbol{B_{\alpha_8}}(\boldsymbol{z})\bar{\alpha} \tag{32}$$

$$\tag{33}$$

De forma que,

$$\eta(x) = \sum_{i=1}^{8} \eta_i(x) = 1 \tag{34}$$

Note que $\eta(x)$ pertence ao simplex unitário, uma vez que as funções de pertinências podem ser vistas como funções de pertinência já normalizadas. Assim podemos escrever o modelo fuzzy do sistema como,

$$\boldsymbol{E}(\eta(x))\dot{\boldsymbol{x}} = \boldsymbol{A}(\eta(x))\boldsymbol{x} + \boldsymbol{B_u}u + \boldsymbol{B_\alpha}(\eta(x))\bar{\alpha} \tag{35}$$

$$\boxed{\sum_{i=1}^{8} \eta_i(x)\boldsymbol{E}_i\dot{\boldsymbol{x}} = \sum_{i=1}^{8} \eta_i(x)\boldsymbol{A}_i\boldsymbol{x} + \sum_{i=1}^{8} \eta_i(x)\boldsymbol{B_u}u + \sum_{i=1}^{8} \eta_i(x)\boldsymbol{B_{\alpha_i}}\bar{\alpha}} \tag{36}$$

1.3 Código - ScriptTakagiSugenoModel.m

```
clear
clc
disp('++++++++++++++++++++++++++++++++++++')
disp('--> Running File "ScripTakagiSugenoModel.m"')
disp('++++++++++++++++++++++++++++++++++++')
syms M m l g kv real
syms z1 z2 z3 real

Ez = [1 0 0 0;...
0 1 0 0;...
0  0 M+m -m*l*z1;...
0  0 -m*l*z1 m*l^2];

Az = [0 0 1 0;...
0 0 0 1;...
0 0 -kv -m*l*z2;...
0 m*g*l*z3 0 0];

Bz = [0;0;1;0];

Baz = [0;0; -(M+m)*g;m*g*l*z1];

syms s theta v omega alfa sinctheta real



```

```
%% Define Function sinc(x) = sin(x)/x
sinceq = @(x) sin(x)/x;
%% Implementa Funcao para Pegar Pertencimento da Regra
getw = @(gx,max,min) (gx-min)/(max - min);
%% Sector Non Linearities
Z1 = subs(z1,cos(theta));
Z2 = subs(z2,sin(theta)*omega);
Z3 = subs(z3,cos(alfa)*sinctheta);
Z = [Z1;Z2;Z3];
%% Acha o Minimo de Z1
IntervalTheta = [-deg2rad(80) deg2rad(80)];
it =1;
minZ1= 100;
maxZ1 =0;
for th = IntervalTheta(1):0.0175:IntervalTheta(2)
teste(it) = cos(th);
if teste(it) < minZ1
minZ1 = teste(it);
end
if teste(it) > maxZ1
maxZ1 = teste(it);
end
it = it+1;
end


az1max = maxZ1;
az1min = minZ1;
wz1max = getw(Z1,az1max,az1min);
wz1min = 1 - wz1max;


%% Acha o Minimo de Z2
it =1;
minZ2= 100;
maxZ2 =0;
IntervalOmega = [-deg2rad(20) deg2rad(20)];
for th = IntervalTheta(1):0.0175:IntervalTheta(2)
for om = IntervalOmega(1):0.0175:IntervalOmega(2)
teste(it) = sin(th)*om;
if teste(it) < minZ2
minZ2 = teste(it);
end
if teste(it) > maxZ2
maxZ2 = teste(it);
end
it = it+1;
end
end


az2max = maxZ2;
az2min = minZ2;
wz2max = getw(Z2,az2max,az2min);
wz2min = 1 - wz2max;


%% Acha o Minimo de Z3
it =1;
minZ3= 100;
maxZ3 =0;
%   IntervalAlfa = [-0.1745 0.1745];
IntervalAlfa = [-deg2rad(10) deg2rad(10)];
for th = IntervalTheta(1):0.0175:IntervalTheta(2)
for al = IntervalAlfa(1):0.0175:IntervalAlfa(2)
teste(it) = cos(al)*sinceq(th);
if th == 0
teste(it) = cos(al);
end
if teste(it) < minZ3
```

```
minZ3 = teste(it);
end
if teste(it) > maxZ3
maxZ3 = teste(it);
end
it = it+1;
end
end


az3max = maxZ3;
az3min = minZ3;
wz3max = getw(Z3,az3max,az3min);
wz3min = 1 - wz3max ;

%% Define Vertices
vertices = 2^(length(Z))
E_ = cell(vertices,1);
A_ = cell(vertices,1);
Bu_ = cell(vertices,1);
Ba_ = cell(vertices,1);
%% Regra 1 max max max
E_{1} = subs(Ez,z1,az1max);
A_{1} = subs(Az,[z2 z3],[az2max az3max ]);
Bu_{1} = Bz;
Ba_{1} = subs(Baz,z1,az1max);
h(1) = wz1max*wz2max*wz3max;
%% Regra 2 max max min
E_{2} = subs(Ez,z1,az1max);
A_{2} = subs(Az,[z2 z3],[az2max az3min ]);
Bu_{2} = Bz;
Ba_{2} = subs(Baz,z1,az1max);
h(2) = wz1max*wz2max*wz3min;
%% Regra 3 max min max
E_{3} = subs(Ez,z1,az1max);
A_{3} = subs(Az,[z2 z3],[az2min az3max ]);
Bu_{3} = Bz;
Ba_{3} = subs(Baz,z1,az1max);
h(3) = wz1max*wz2min*wz3max;
%% Regra 4 max min min
E_{4} = subs(Ez,z1,az1max);
A_{4} = subs(Az,[z2 z3],[az2min az3min ]);
Bu_{4} = Bz;
Ba_{4} = subs(Baz,z1,az1max);
h(4) = wz1max*wz2min*wz3min;
%% Regra 5 min max max
E_{5} = subs(Ez,z1,az1min);
A_{5} = subs(Az,[z2 z3],[az2max az3max ]);
Bu_{5} = Bz;
Ba_{5} = subs(Baz,z1,az1min);
h(5) = wz1min*wz2max*wz3max;
%% Regra 6 min max min
E_{6} = subs(Ez,z1,az1min);
A_{6} = subs(Az,[z2 z3],[az2max az3min ]);
Bu_{6} = Bz;
Ba_{6} = subs(Baz,z1,az1min);
h(6) = wz1min*wz2max*wz3min;
%% Regra 7 min min max
E_{7} = subs(Ez,z1,az1min);
A_{7} = subs(Az,[z2 z3],[az2min az3max ]);
Bu_{7} = Bz;
Ba_{7} = subs(Baz,z1,az1min);
h(7) = wz1min*wz2min*wz3max;
%% Regra 8 min min min
E_{8} = subs(Ez,z1,az1min);
A_{8} = subs(Az,[z2 z3],[az2min az3min ]);
Bu_{8} = Bz;
Ba_{8} = subs(Baz,z1,az1min);
h(8) = wz1min*wz2min*wz3min;

```

```
%% check if h belongs to unitary simplex
test = simplify(sum(h))
if test == 1
disp('--> h belogs to the unitary simplex')
else
error('h doesnt belong to the unitary simplex')
end


%% Global Descriptor System
vertices = 8;
subvecParam = [M m l g kv];
subvecValues = [1.5 0.3 0.3 9.78 1];

for i =1:vertices
E_{i} = double(subs(E_{i},subvecParam,subvecValues));
A_{i} = double(subs(A_{i},subvecParam,subvecValues));
Bu_{i} = double(subs(Bu_{i},subvecParam,subvecValues));
Ba_{i} = double(subs(Ba_{i},subvecParam,subvecValues));
% C_{i} = [0 0 1 0; 0 0 0 1];
C_{i} = eye(4);
end

% Define a pertinencia da regra como variavel global
%para ser usada pelo controlador
global PertinenciaNormalizada
PertinenciaNormalizada.h = h;

% Define vertices do sistema usados para controle
global DescriptorSystemDynamics

DescriptorSystemDynamics.E = E_;
DescriptorSystemDynamics.A = A_;
DescriptorSystemDynamics.Bu = Bu_;
DescriptorSystemDynamics.Ba = Ba_;
DescriptorSystemDynamics.C = C_;

disp('--> Takagi Sugeno Model Obtained')
disp('+++++++++++++++++++++++++++++++++++++++++++')
```

Listing 1: Script Constrói Modelo TS

2 Controle

2.1 Lei de Controle PDC

A lei de controle PDC toma a forma,

$$\boxed{u = \sum_{j=1}^{8} \eta_j(x)\boldsymbol{K_j x}} \tag{37}$$

Além disso, as seguintes afirmações do Lemma de Finsler serão utilizadas,

$$\begin{cases} \hat{\boldsymbol{x}}'\boldsymbol{\mathcal{Q}}\hat{\boldsymbol{x}} < 0, \forall \boldsymbol{\mathfrak{B}}\hat{\boldsymbol{x}} = 0 \\ \boldsymbol{\mathcal{Q}} + \boldsymbol{\mathcal{X}}\boldsymbol{\mathfrak{B}} + \boldsymbol{\mathfrak{B}}'\boldsymbol{\mathcal{X}} < 0 \end{cases} \tag{38}$$

Definindo uma função de Lyapunov quadrática com os estados tal que $\boldsymbol{P} > 0$,

$$\boldsymbol{V} = \boldsymbol{x}'\boldsymbol{P}\boldsymbol{x} \tag{39}$$

$$\dot{\boldsymbol{V}} = \dot{\boldsymbol{x}}'\boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}'\boldsymbol{P}\dot{\boldsymbol{x}}' < 0 \tag{40}$$

$$\dot{\boldsymbol{V}} = \begin{bmatrix} \boldsymbol{x}' & \dot{\boldsymbol{x}}' & \hat{\alpha}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\emptyset} & \boldsymbol{P} & \boldsymbol{\emptyset} \\ * & \boldsymbol{\emptyset} & \boldsymbol{\emptyset} \\ * & * & \boldsymbol{\emptyset} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \\ \hat{\alpha} \end{bmatrix} < 0, \forall \boldsymbol{\mathfrak{B}}\hat{\boldsymbol{x}} = 0 \tag{41}$$

Para obter $\mathfrak{B}$ vamos tomar a equação da dinâmica em malha fechada,

$$\sum_{i=1}^{8}\eta_i(x)\boldsymbol{E}_i\dot{\boldsymbol{x}} = \sum_{i=1}^{8}\sum_{j=1}^{8}\eta_i(x)\eta_j(x)\big(\boldsymbol{A}_i + \boldsymbol{B}_{\boldsymbol{u_i}}\boldsymbol{K}_j\big)\boldsymbol{x} + \sum_{i=1}^{8}\eta_i(x)\boldsymbol{B}_{\boldsymbol{\alpha_i}}\bar{\alpha} \tag{42}$$

$$\boldsymbol{E}(\eta(x))\dot{\boldsymbol{x}} = \big(\boldsymbol{A}(\eta(x)) + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{K}(\eta(x))\big)\boldsymbol{x} + \boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))\bar{\alpha} \tag{43}$$

$$\begin{bmatrix}\boldsymbol{A}(\eta(x)) + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{K}(\eta(x)) & -\boldsymbol{E}(\eta(x)) & \boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \dot{\boldsymbol{x}}\\ \hat{\alpha}\end{bmatrix} = 0 \tag{44}$$

De forma que $\mathfrak{B} = \begin{bmatrix}\boldsymbol{A}(\eta(x)) + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{K}(\eta(x)) & -\boldsymbol{E}(\eta(x)) & \boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))\end{bmatrix}$ e tomando as variveis de folga, $\boldsymbol{\mathcal{X}} = \begin{bmatrix}\boldsymbol{\mathcal{X}}_1\\ \boldsymbol{\mathcal{X}}_2\\ \boldsymbol{\mathcal{X}}_3\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mathcal{X}}^{-T}\\ \mu\boldsymbol{\mathcal{X}}^{-T}\\ \boldsymbol{\emptyset}\end{bmatrix}$ onde μ é um escalar positivo. Tomando então a segunda relação de Finsler mostrada em (38),

$$\begin{aligned}\begin{bmatrix}\boldsymbol{\emptyset} & \boldsymbol{P} & \boldsymbol{\emptyset}\\ * & \boldsymbol{\emptyset} & \boldsymbol{\emptyset}\\ * & * & \boldsymbol{\emptyset}\end{bmatrix} + \begin{bmatrix}\boldsymbol{\mathcal{X}}^{-T}\\ \mu\boldsymbol{\mathcal{X}}^{-T}\\ \boldsymbol{\emptyset}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}(\eta(x)) + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{K}(\eta(x)) & -\boldsymbol{E}(\eta(x)) & \boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))\end{bmatrix}\\ + \begin{bmatrix}\boldsymbol{A}(\eta(x))' + \boldsymbol{K}(\eta(x))'\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))'\\ -\boldsymbol{E}(\eta(x))'\\ \boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))'\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{X}}^{-1} & \mu\boldsymbol{\mathcal{X}}^{-1} & \boldsymbol{\emptyset}\end{bmatrix} < 0\end{aligned} \tag{45}$$

$$\begin{bmatrix}\boldsymbol{\mathcal{X}}^{-T}\left\{\boldsymbol{A}(\eta(x)) + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{K}(\eta(x))\right\} + \left\{\boldsymbol{A}(\eta(x))' + \boldsymbol{K}(\eta(x))'\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))'\right\}\boldsymbol{\mathcal{X}}^{-1} & \boldsymbol{P} - \boldsymbol{\mathcal{X}}^{-T}\boldsymbol{E}(\eta(x)) + \mu\left\{\boldsymbol{A}(\eta(x))' + \boldsymbol{K}(\eta(x))'\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))'\right\}\boldsymbol{\mathcal{X}}^{-1} & \boldsymbol{\mathcal{X}}^{-T}\boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))\\ * & -\mu\boldsymbol{\mathcal{X}}^{-T}\boldsymbol{E}(\eta(x)) - \mu\boldsymbol{E}(\eta(x))'\boldsymbol{\mathcal{X}}^{-1} & \mu\boldsymbol{\mathcal{X}}^{-T}\boldsymbol{B}_{\boldsymbol{\alpha}}(\eta(x))\\ * & * & \boldsymbol{\emptyset}\end{bmatrix} < 0 \tag{46}$$

Fazendo uma transformação de congruência multiplicando pelo lado direito por $\begin{bmatrix}\boldsymbol{\mathcal{X}} & \boldsymbol{\emptyset} & \boldsymbol{\emptyset}\\ * & \boldsymbol{\mathcal{X}} & \boldsymbol{\emptyset}\\ * & * & \boldsymbol{\emptyset}\end{bmatrix}$ e por sua transposta no lado esquerdo,

$$\begin{bmatrix}\boldsymbol{A}(\eta(x))\boldsymbol{\mathcal{X}} + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{K}(\eta(x))\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{X}}'\boldsymbol{A}(\eta(x))' + \boldsymbol{\mathcal{X}}'\boldsymbol{K}(\eta(x))\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))' & \boldsymbol{\mathcal{X}}'\boldsymbol{P}\boldsymbol{\mathcal{X}} - \boldsymbol{E}(\eta(x))\boldsymbol{\mathcal{X}} + \mu\boldsymbol{\mathcal{X}}'\boldsymbol{A}(\eta(x))' + \mu\boldsymbol{\mathcal{X}}'\boldsymbol{K}(\eta(x))'\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))' & \boldsymbol{\emptyset}\\ * & -\mu\boldsymbol{E}(\eta(x))\boldsymbol{\mathcal{X}} - \mu\boldsymbol{\mathcal{X}}'\boldsymbol{E}(\eta(x))' & \boldsymbol{\emptyset}\\ * & * & \boldsymbol{\emptyset}\end{bmatrix} < 0 \tag{47}$$

Fazendo uma transformação do tipo $\bar{\boldsymbol{P}} = \boldsymbol{\mathcal{X}}'\boldsymbol{P}\boldsymbol{\mathcal{X}}$ e $\boldsymbol{\mathcal{Y}}(\eta(x)) = \boldsymbol{K}(\eta(x))\boldsymbol{\mathcal{X}}$ e eliminando as colunas com zeros para ajudar na implementação, obtemos a seguinte desigualdade matricial.

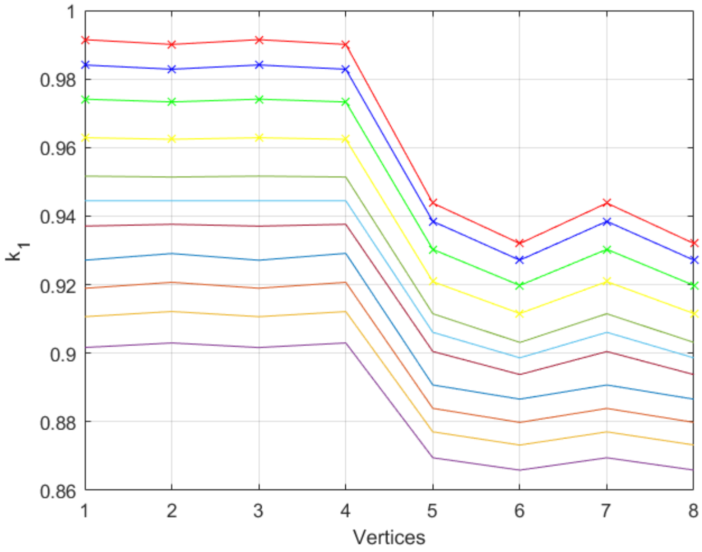
$$\begin{bmatrix}\boldsymbol{A}(\eta(x))\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{X}}'\boldsymbol{A}(\eta(x))' + \boldsymbol{B}_{\boldsymbol{u}}(\eta(x))\boldsymbol{\mathcal{Y}}(\eta(x)) + \boldsymbol{\mathcal{Y}}(\eta(x))'\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))' & \bar{\boldsymbol{P}} - \boldsymbol{E}(\eta(x))\boldsymbol{\mathcal{X}} + \mu\big(\boldsymbol{\mathcal{X}}'\boldsymbol{A}(\eta(x))' + \boldsymbol{\mathcal{Y}}(\eta(x))'\boldsymbol{B}_{\boldsymbol{u}}(\eta(x))'\big)\\ * & -\mu\big(\boldsymbol{E}(\eta(x))\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{X}}'\boldsymbol{E}(\eta(x))'\big)\end{bmatrix} < 0 \tag{48}$$

Chamando essa desigualdade matricial acima de $\boldsymbol{\Phi}$, homogenizando os termos, e lembrando que $(.)(\eta(x)) = \sum_{i=1}^{8}\eta_i(.)_i$ podemos achar as condições de LMI suficientes dadas abaixo,

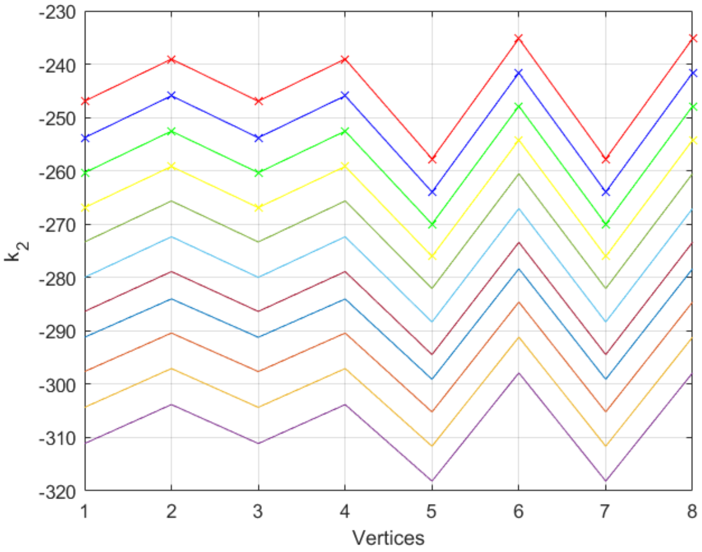
$$\begin{cases}\bar{\boldsymbol{P}} > 0\\ \boldsymbol{\Phi}_{ii} < 0, \forall i\\ \boldsymbol{\Phi}_{ij} + \boldsymbol{\Phi}_{ji} < 0, \text{onde } 1 \leq i \leq 7 \text{ e } 2 \leq j \leq 8\end{cases} \tag{49}$$

2.1.1 Resultados PDC Control

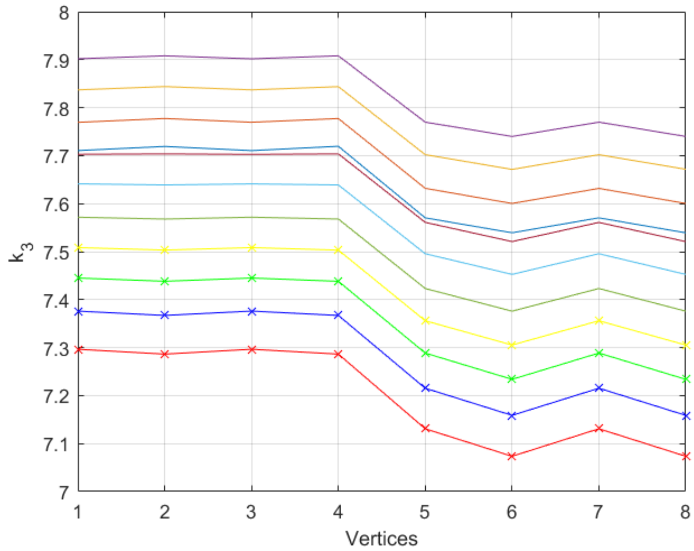
Uma busca foi feita para se obter o escalar μ o variando de 10 até 20 com o passo fixo de 1, e analisando os resultados dos ganhos para avaliar a escolha do escalar. Os resultados de como μ influência nos ganhos do controlador são mostrados abaixo.



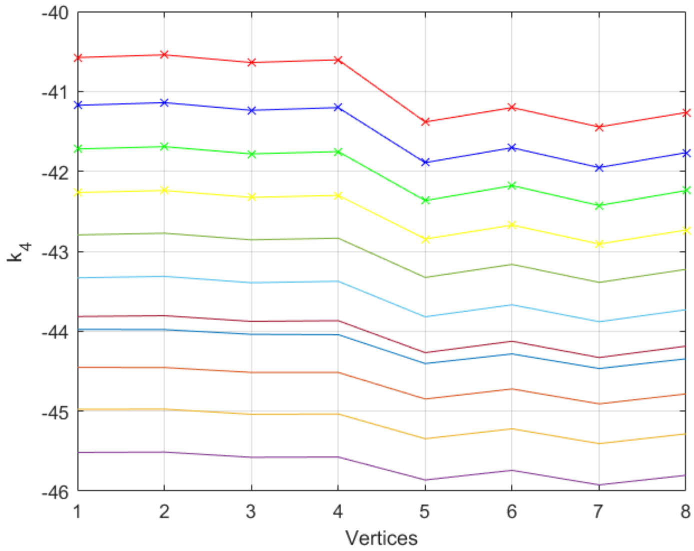
(a) k_1



(b) k_2



(a) k_3

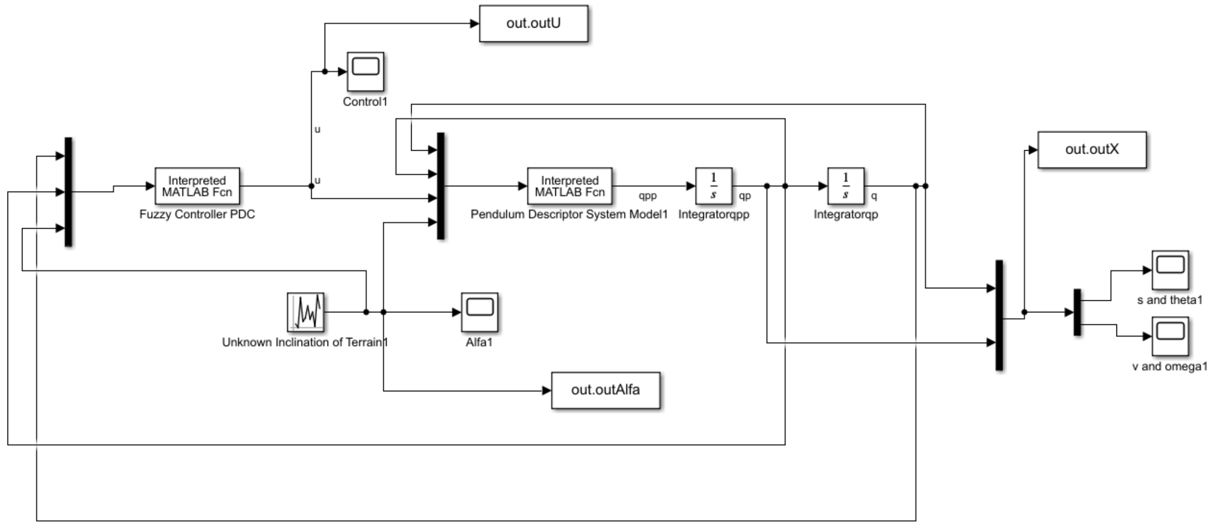


(b) k_4

Selecionando-se então $\mu = 10$ os resultados factíveis da otimização do solver SeDuMi são mostrados abaixo.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|    ID|          Constraint|   Primal residual|   Dual residual|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|    #1|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|    #2|   Matrix inequality|        5.3298e-06|      8.1486e-14|
|    #3|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|    #4|   Matrix inequality|        1.7589e-06|      8.1058e-14|
|    #5|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|    #6|   Matrix inequality|        5.3298e-06|      8.1483e-14|
|    #7|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|    #8|   Matrix inequality|        1.7589e-06|      8.1058e-14|
|    #9|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|   #10|   Matrix inequality|        2.2195e-06|      7.5549e-14|
|   #11|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|   #12|   Matrix inequality|        1.8181e-06|      8.0837e-14|
|   #13|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|   #14|   Matrix inequality|        2.2196e-06|      7.5547e-14|
|   #15|   Matrix inequality|        6.5607e-05|      2.2051e-14|
|   #16|   Matrix inequality|        1.8181e-06|      8.0835e-14|
|   #17|   Matrix inequality|         8.908e-06|      4.2021e-14|
|   #18|   Matrix inequality|         1.066e-05|      4.4286e-14|
|   #19|   Matrix inequality|         8.908e-06|      4.2021e-14|
|   #20|   Matrix inequality|        1.5769e-05|      3.9693e-14|
|   #21|   Matrix inequality|        1.4005e-05|      4.0319e-14|
|   #22|   Matrix inequality|        1.5769e-05|      3.9693e-14|
|   #23|   Matrix inequality|        1.4005e-05|      4.0319e-14|
|   #24|   Matrix inequality|         8.908e-06|       4.202e-14|
|   #25|   Matrix inequality|        3.5179e-06|       3.981e-14|
|   #26|   Matrix inequality|        1.4395e-05|       3.982e-14|
|   #27|   Matrix inequality|        9.2935e-06|      4.0339e-14|
|   #28|   Matrix inequality|        1.4395e-05|       3.982e-14|
|   #29|   Matrix inequality|        9.2935e-06|      4.0339e-14|
|   #30|   Matrix inequality|         8.908e-06|       4.202e-14|
|   #31|   Matrix inequality|        1.5769e-05|      3.9692e-14|
|   #32|   Matrix inequality|        1.4005e-05|      4.0319e-14|
|   #33|   Matrix inequality|        1.5769e-05|      3.9692e-14|
|   #34|   Matrix inequality|        1.4005e-05|      4.0318e-14|
|   #35|   Matrix inequality|        1.4395e-05|       3.982e-14|
|   #36|   Matrix inequality|        9.2935e-06|      4.0339e-14|
|   #37|   Matrix inequality|        1.4395e-05|       3.982e-14|
|   #38|   Matrix inequality|        9.2935e-06|      4.0339e-14|
|   #39|   Matrix inequality|        1.3658e-05|      3.9955e-14|
|   #40|   Matrix inequality|        4.4391e-06|      3.7045e-14|
|   #41|   Matrix inequality|        1.3658e-05|      3.9955e-14|
|   #42|   Matrix inequality|        1.3658e-05|      3.9955e-14|
|   #43|   Matrix inequality|        3.6363e-06|      3.8506e-14|
|   #44|   Matrix inequality|        1.3658e-05|      3.9954e-14|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
| A primal-dual optimal solution would show non-negative residuals.|
| In practice, many solvers converge to slightly infeasible       |
| solutions, which may cause some residuals to be negative.       |
| It is up to the user to judge the importance and impact of      |
| slightly negative residuals (i.e. infeasibilities)              |
| https://yalmip.github.io/command/check/                         |
| https://yalmip.github.io/faq/solutionviolated/                  |
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Para simular o comportamento do sistema um Simulink foi criado como mostrado na figura.



OBS: apesar de no simulink constar o nome "Pendulum Descriptor System Model1"o modelo implementado foi o dado por (1).

Uma simulação com 60 segundos de duração foi feita, tendo como condições iniciais $\boldsymbol{q}(0) = \begin{bmatrix} 0 \\ \frac{\pi}{4} \end{bmatrix}$ e $\dot{\boldsymbol{q}}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, e com a inclinação do terreno aleatória variando entre mais ou menos dez graus. Os resultados dos estados e entrada de controle são mostrados abaixo.

2.2 Lei de Controle PDC com $\mathcal{H}_\infty$

A mesma lei de controle PDC será utilizada bem como as afirmações do Lemma de Finsler ,

$$\boxed{u = \sum_{j=1}^{8} \eta_j(x)\boldsymbol{K_j x}} \tag{50}$$

Essa será uma demonstração muito mais rápida e com menos detalhes. Tomando a derivada temporal da função de Lyapunov a mesma assumirá a seguinte forma para o problema do $\mathcal{H}_\infty$. Assumindo $\boldsymbol{z} = \boldsymbol{Cx} + \boldsymbol{D_\alpha}\hat{\alpha}$

$$\dot{\boldsymbol{V}} + \frac{1}{\gamma}\boldsymbol{x}'\boldsymbol{C}'\boldsymbol{C}\boldsymbol{x} - \gamma\bar{\alpha}'\bar{\alpha} < 0 \tag{51}$$

$$\begin{bmatrix} \boldsymbol{x}' & \dot{\boldsymbol{x}}' & \hat{\alpha}' \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{\emptyset} & \boldsymbol{P} & \boldsymbol{\emptyset} \\ * & \boldsymbol{\emptyset} & \boldsymbol{\emptyset} \\ * & * & -\gamma\boldsymbol{I} \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} \boldsymbol{C}' \\ \boldsymbol{\emptyset} \\ \boldsymbol{D}'_{\boldsymbol{\alpha}} \end{bmatrix} \begin{bmatrix} \boldsymbol{C} & \boldsymbol{\emptyset} & \boldsymbol{D_\alpha} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \\ \hat{\alpha} \end{bmatrix} < 0 \tag{52}$$

Tomando os mesmo passos de se multiplicar pelo lado direito e esquerdo pela mesma transformação de congruência utilizada para o controlador PDC, utilizando as mesmas transformações linearizantes, utilizando complemento de Schur e eliminando as colunas de zeros para facilitar a resolução do solver SeDuMi obtemos a seguinte desigualdade matricial,

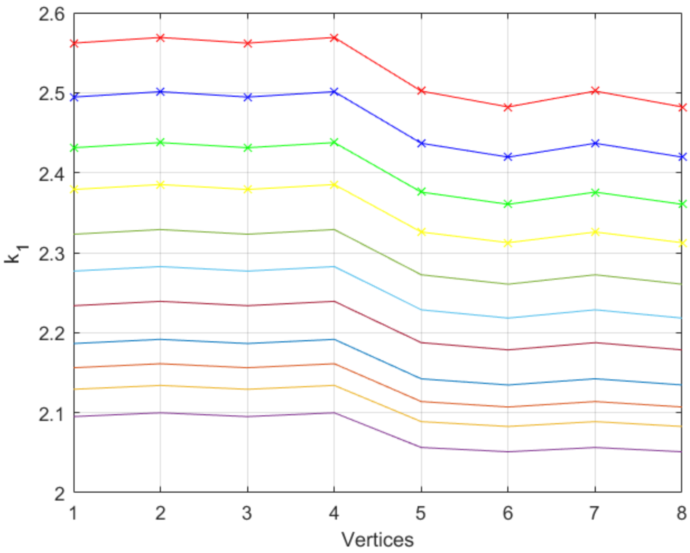
$$\begin{bmatrix} \boldsymbol{A}(\eta(x))\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{X}}'\boldsymbol{A}(\eta(x))' + \boldsymbol{B_u}(\eta(x))\boldsymbol{\mathcal{Y}}(\eta(x)) + \boldsymbol{\mathcal{Y}}(\eta(x))'\boldsymbol{B_u}(\eta(x)) & \bar{\boldsymbol{P}} - \boldsymbol{E}(\eta(x))\boldsymbol{\mathcal{X}} + \mu(\boldsymbol{\mathcal{X}}'\boldsymbol{A}(\eta(x))' + \boldsymbol{\mathcal{Y}}(\eta(x))'\boldsymbol{B_u}(\eta(x))') & \boldsymbol{\mathcal{X}}'\boldsymbol{C}' \\ * & -\mu(\boldsymbol{E}(\eta(x))\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{X}}'\boldsymbol{E}(\eta(x))') & \boldsymbol{\emptyset} \\ * & * & -\gamma\boldsymbol{I} \end{bmatrix} < 0 \tag{53}$$

Chamando a desigualdade de Ψ, homogenizando os termos e lembrando que $\bar{\boldsymbol{P}} = \boldsymbol{\mathcal{X}}'\boldsymbol{P}\boldsymbol{\mathcal{X}}$ e $\boldsymbol{\mathcal{Y}}(\eta(x)) = \boldsymbol{K}(\eta(x))\boldsymbol{\mathcal{X}}$, as seguintes LMIs devem ser implementadas.

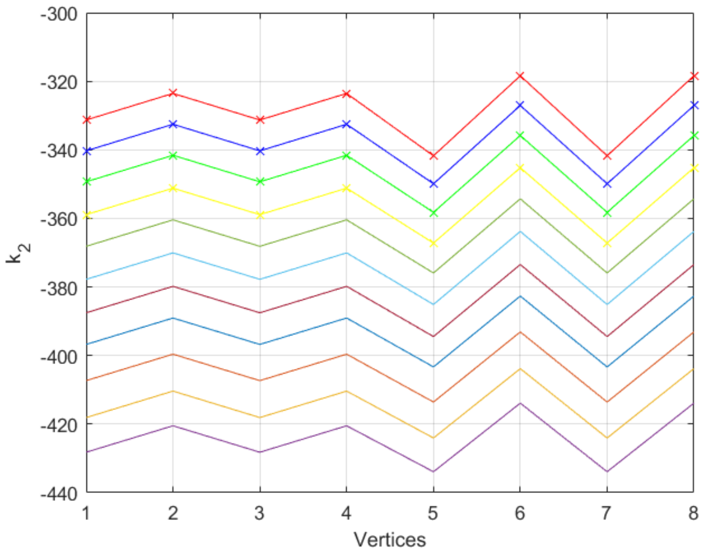
$$\begin{cases} \bar{\boldsymbol{P}} > 0 \\ \boldsymbol{\Psi}_{ii} < 0, \forall i \\ \boldsymbol{\Psi}_{ij} + \boldsymbol{\Psi}_{ji} < 0, \text{onde } 1 \leq i \leq 7 \text{ e } 2 \leq j \leq 8 \end{cases} \tag{54}$$

2.2.1 Resultados PDC com $\mathcal{H}_\infty$

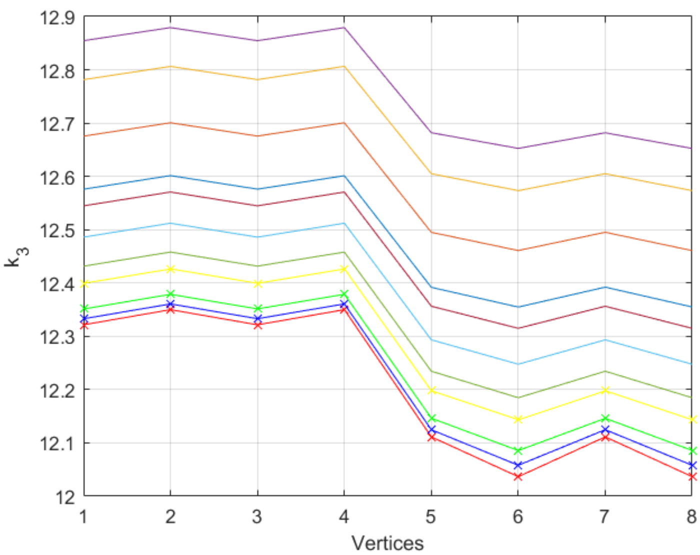
Uma busca foi feita para se obter o escalar μ o variando de 10 até 20 com o passo fixo de 1, e analisando os resultados dos ganhos para avaliar a escolha do escalar. Os resultados de como μ influência nos ganhos do controlador são mostrados abaixo.



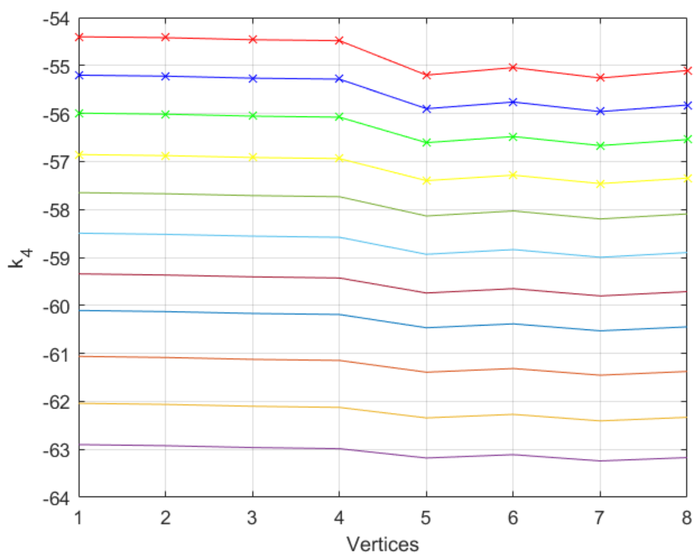
(a) k_1



(b) k_2



(a) k_3



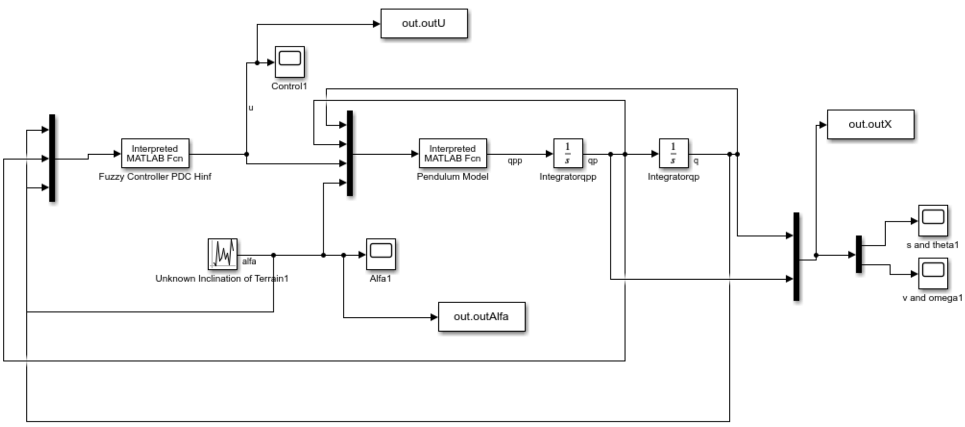
(b) k_4

Selecionando-se então $\mu = 10$ os resultados factíveis da otimização do solver SeDuMi são mostrados abaixo.

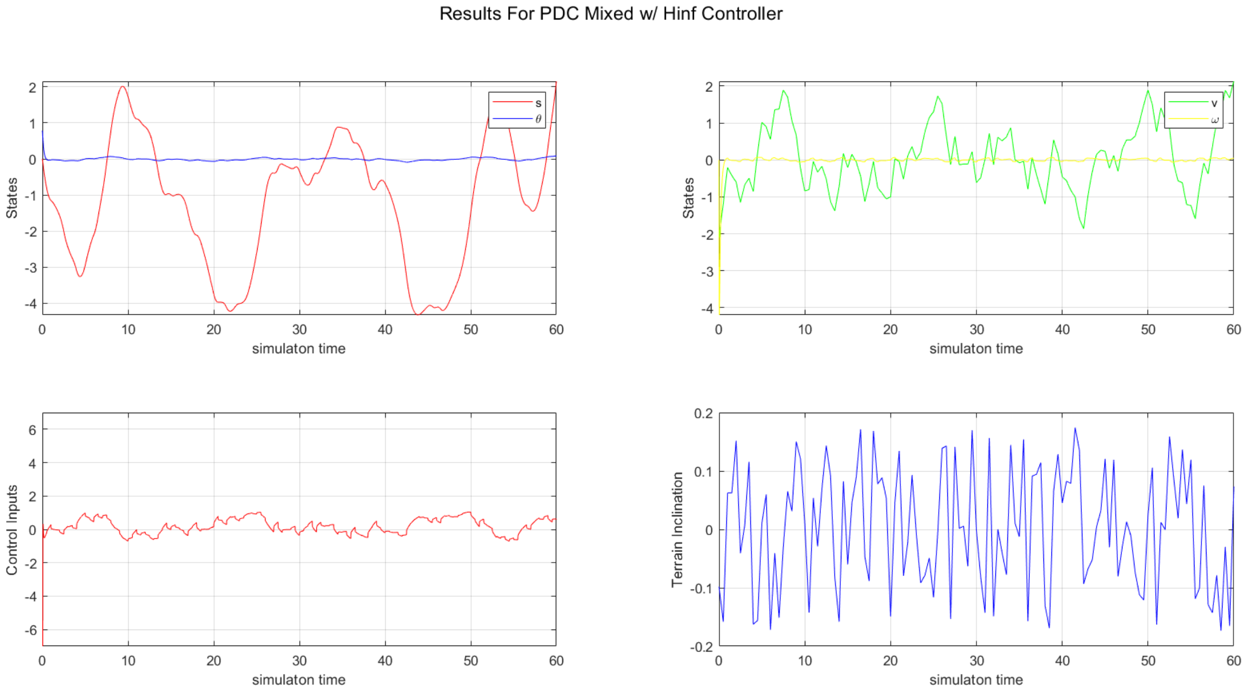
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|     ID|           Constraint|   Primal residual|   Dual residual|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|     #1|    Matrix inequality|        2.9312e-07|       1.245e-14|
|     #2|    Matrix inequality|        1.9837e-08|      5.6459e-14|
|     #3|    Matrix inequality|        2.9312e-07|       1.245e-14|
|     #4|    Matrix inequality|        3.5097e-09|      5.6862e-14|
|     #5|    Matrix inequality|        2.9312e-07|       1.245e-14|
|     #6|    Matrix inequality|        1.9837e-08|      5.6463e-14|
|     #7|    Matrix inequality|        2.9312e-07|       1.245e-14|
|     #8|    Matrix inequality|        3.5097e-09|      5.6862e-14|
|     #9|    Matrix inequality|        2.9312e-07|       1.245e-14|
|    #10|    Matrix inequality|        4.9461e-09|      5.3139e-14|
|    #11|    Matrix inequality|        2.9312e-07|       1.245e-14|
|    #12|    Matrix inequality|        3.6778e-09|       5.743e-14|
|    #13|    Matrix inequality|        2.9312e-07|       1.245e-14|
```

```
|    #14|    Matrix inequality|         4.9461e-09|         5.314e-14|
|    #15|    Matrix inequality|         2.9312e-07|         1.245e-14|
|    #16|    Matrix inequality|         3.6778e-09|        5.7426e-14|
|    #17|    Matrix inequality|         3.1635e-08|        2.8329e-14|
|    #18|    Matrix inequality|         3.9673e-08|        2.9228e-14|
|    #19|    Matrix inequality|         3.1635e-08|        2.8329e-14|
|    #20|    Matrix inequality|          5.002e-08|        2.7924e-14|
|    #21|    Matrix inequality|          4.421e-08|        2.8477e-14|
|    #22|    Matrix inequality|          5.002e-08|        2.7925e-14|
|    #23|    Matrix inequality|          4.421e-08|        2.8477e-14|
|    #24|    Matrix inequality|         3.1635e-08|        2.8329e-14|
|    #25|    Matrix inequality|         7.0194e-09|        2.5329e-14|
|    #26|    Matrix inequality|         4.6352e-08|        2.7937e-14|
|    #27|    Matrix inequality|         2.7814e-08|        2.8446e-14|
|    #28|    Matrix inequality|         4.6352e-08|        2.7937e-14|
|    #29|    Matrix inequality|         2.7814e-08|        2.8446e-14|
|    #30|    Matrix inequality|         3.1635e-08|        2.8329e-14|
|    #31|    Matrix inequality|          5.002e-08|        2.7925e-14|
|    #32|    Matrix inequality|          4.421e-08|        2.8478e-14|
|    #33|    Matrix inequality|          5.002e-08|        2.7926e-14|
|    #34|    Matrix inequality|          4.421e-08|        2.8477e-14|
|    #35|    Matrix inequality|         4.6352e-08|        2.7936e-14|
|    #36|    Matrix inequality|         2.7813e-08|        2.8446e-14|
|    #37|    Matrix inequality|         4.6352e-08|        2.7937e-14|
|    #38|    Matrix inequality|         2.7814e-08|        2.8446e-14|
|    #39|    Matrix inequality|         5.1287e-08|        2.7741e-14|
|    #40|    Matrix inequality|         9.8922e-09|        2.9111e-14|
|    #41|    Matrix inequality|         5.1287e-08|        2.7739e-14|
|    #42|    Matrix inequality|         5.1287e-08|        2.7741e-14|
|    #43|    Matrix inequality|         7.3556e-09|        2.5582e-14|
|    #44|    Matrix inequality|         5.1287e-08|        2.7739e-14|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
| A primal-dual optimal solution would show non-negative residuals.|
| In practice, many solvers converge to slightly infeasible     |
| solutions, which may cause some residuals to be negative.     |
| It is up to the user to judge the importance and impact of    |
| slightly negative residuals (i.e. infeasibilities)            |
| https://yalmip.github.io/command/check/                       |
| https://yalmip.github.io/faq/solutionviolated/                |
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Para simular o comportamento do sistema um Simulink foi criado como mostrado na figura.



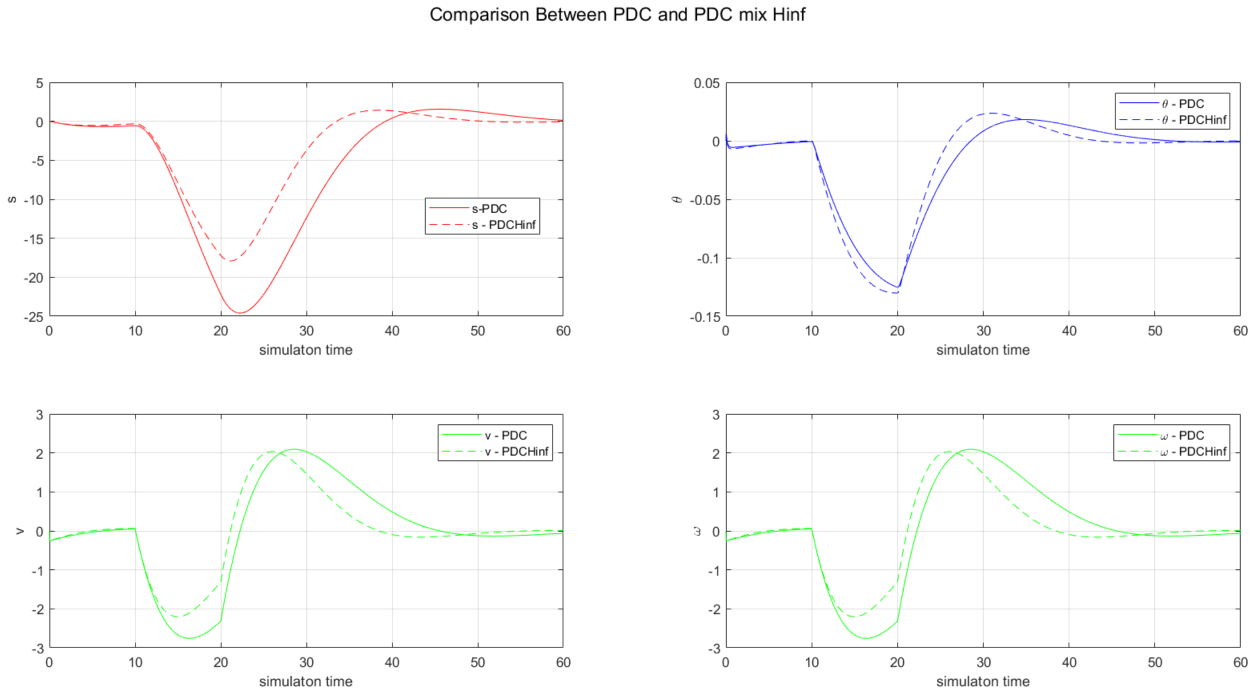
Uma simulação com 60 segundos de duração foi feita, tendo como condições iniciais $\boldsymbol{q}(0) = \begin{bmatrix} 0 \\ \frac{\pi}{4} \end{bmatrix}$ e $\dot{\boldsymbol{q}}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, e com a inclinação do terreno aleatória variando entre mais ou menos dez graus. Os resultados dos estados e entrada de controle são mostrados abaixo.



3 Comparação

3.1 Comparando os Controladores

Para comparar ambos os controladores primeiro modificou-se o simulink para ambos estarem submetidos a uma mesma pertubação, além disso as condições iniciais também foram iguais para ambos os sistemas, e os passo do solver de integração numérica foi definido como 0.001 para ambos os controladores. Os resultados são vistos abaixo.



Como se esperava o controlador com $\mathcal{H}_\infty$ apresenta o melhor comportamento convergindo para zero no menor tempo e com menos *overshoot*.

As simulações podem ser testadas acessando meu repositório no github: https://github.com/Jonatan29/ParallelDistributedCont
tree/master