Modelado, Identificación & Control Digital de un Péndulo Rotativo

TP Final - Control Automático (22.93)

Integrantes:
Marcelo Regueira - 58300
Valentina Lago - 57249

Experimentos

Simulaciones Pendulum_CAD_Control2293 **B, C y D**

Linealización

Paper "**On the dynamics of the Furuta Pendulum**"

Control LQI

Visto en clase de **Control Automático (22.93)**

Control LQI con observador

Observador utilizado en clases de **Control Automático (22.93)** + **Libro Franklin, Powell y Emami-Naeini** (observador de orden reducido)

Loop Shaping

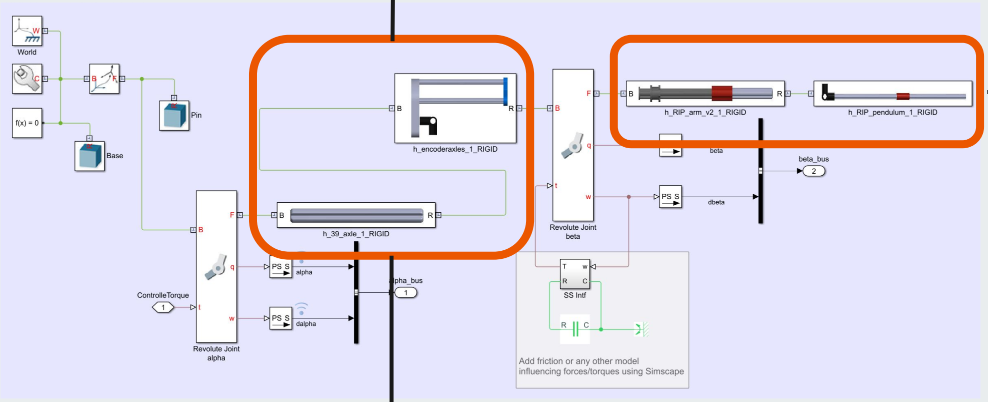
Apunte de "Nyquist" visto en clases de Control Automático (22.93) + **Libro de Goodwin**

Implementación en µC

Libro Modern Control Engineering - Ogata

Datos del sistema

Pendulum_CAD_Control2293



Masa de la base (m1)

$$m_1 = 0,01835kg$$

Masa del péndulo (m2)

$$m_2 = 0,00575kg$$

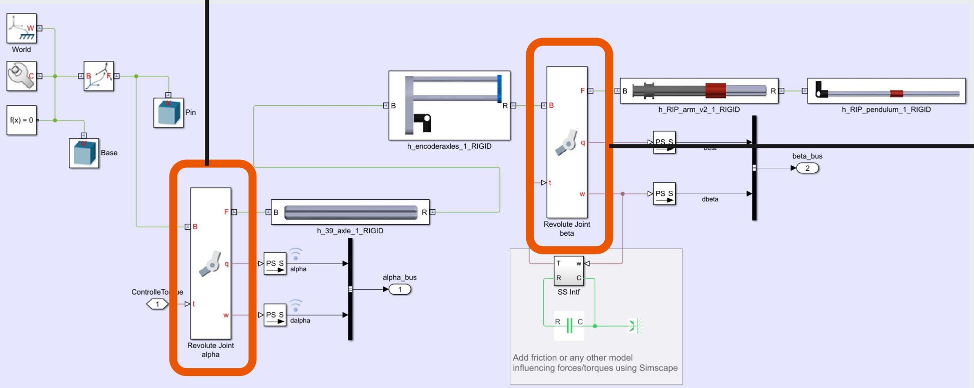
Largo de la base (L1)

$$L_1 = 0,1035m$$

Experimentos

Datos del sistema

Pendulum_CAD_Control2293



Coef. de amortiguamiento (b1)

$$\boxed{b_1 = 10^{-6}}$$

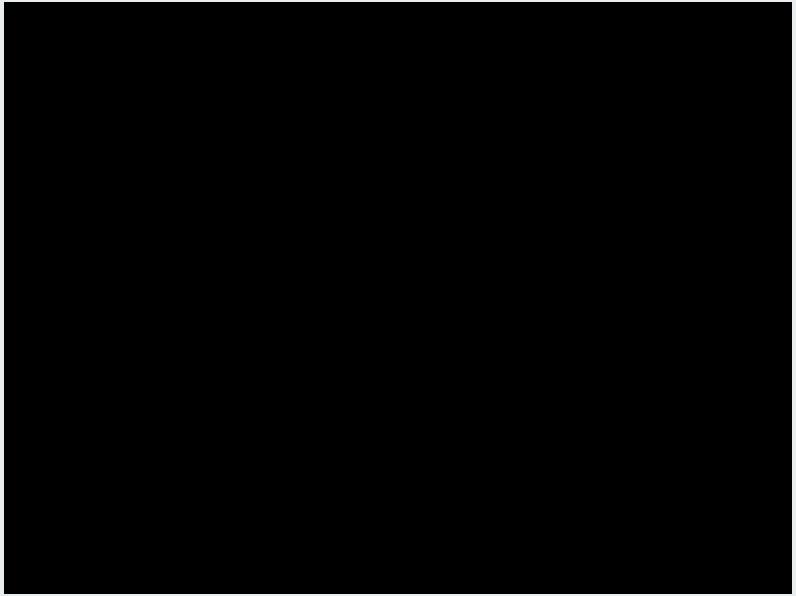
Coef. de amortiguamiento (b2)

$$\boxed{b_2 = 10^{-6}}$$

Experimentos

Experimentos

Pendulum_CAD_Control2293B

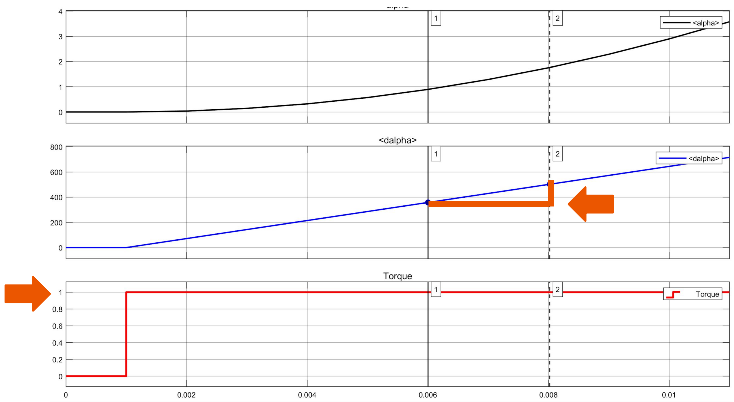


Momento de inercia (J1)

$$\tau = J_1 \cdot \alpha$$

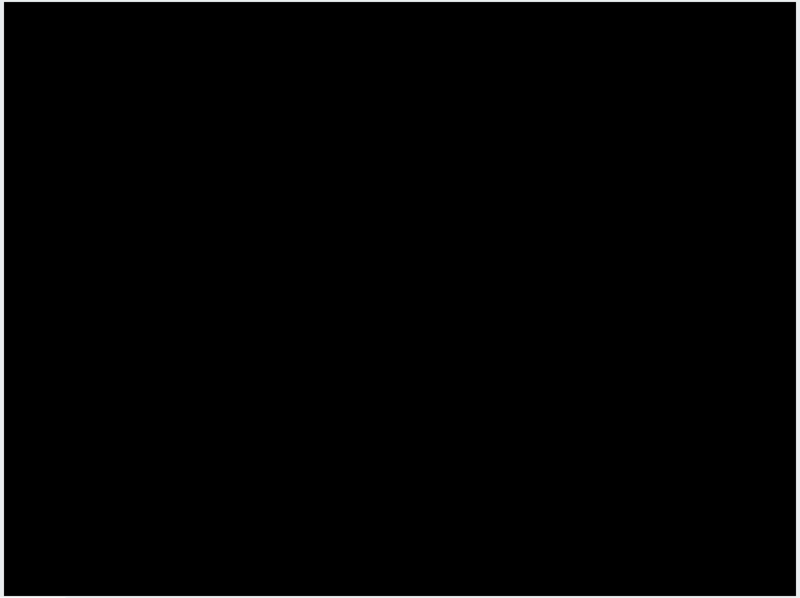
$$J_1 = \frac{\tau}{\alpha} = \frac{1Nm}{5,71 \cdot 10^4 \frac{rad}{s^2}} = \frac{1\frac{kg \cdot m^2}{\cancel{s^2}}}{\frac{5{,}71 \cdot 10^4}{2\pi}\frac{1}{\cancel{s^2}}}$$

$$\boxed{J_1 = 1,1 \cdot 10^{-4} kg \cdot m^2}$$



Experimentos

Pendulum_CAD_Control2293D



Distancia al centro de masa (l2)

$$\tau_1 = 0$$

Ecuación dinámica de rotación:

$$\tau_2 = g \cdot m_2 \cdot l_2 \cdot sin(\theta_2)$$

$$l_2 = \frac{\tau_2}{g \cdot m_2 \cdot sin(\theta_2)}$$

$$l_2 = \frac{2 \cdot 10^{-3} Nm}{9,81 \frac{m}{s^2} \cdot 0,00575kg \cdot sin(0.6592)}$$

$$\boxed{l_2 = 0,0579m}$$

Experimentos

Pendulum_CAD_Control2293C

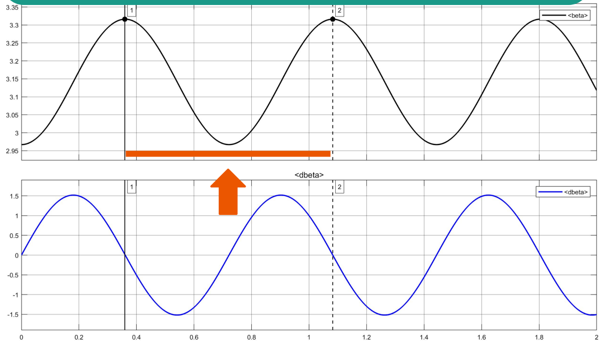
Momento de inercia (J2)

Ecuación diferencial M.A.S:

$$p = 2\pi \sqrt{\frac{J_2}{g \cdot m_2 \cdot l_2}}$$

$$J_2 = \left(\frac{p}{2\pi}\right)^2 \cdot g \cdot m_2 \cdot l_2$$

$$\boxed{J_2 = 4,3 \cdot 10^{-5} kg \cdot m^2}$$



Linealización de ecuaciones de estado

Ecuación (36) de "On the dynamics of the Furuta Pendulum'(*)

$$\begin{bmatrix}\dot\theta_1\\ \dot\theta_2\\ \ddot\theta_1\\ \ddot\theta_2\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ A_{31} & A_{32} & A_{33} & A_{34}\\ A_{41} & A_{42} & A_{43} & A_{44}\end{bmatrix}}_{A}\begin{bmatrix}\theta_1\\ \theta_2\\ \dot\theta_1\\ \dot\theta_2\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 0\\ B_{31}\\ B_{41}\end{bmatrix}}_{B}\tau_1$$

Finalmente, tomamos:

$$\underbrace{\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}}_{C}$$

$$A_{31} = 0,$$

$$A_{32} = \frac{g m_2^2 l_2^2 L_1}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$A_{33} = \frac{-b_1\hat{J}_2}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$A_{34} = \frac{-b_2 m_2 l_2 L_1}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$A_{41} = 0,$$

$$A_{42} = \frac{g m_2 l_2 \hat{J}_0}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$A_{43} = \frac{-b_1 m_2 l_2 L_1}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$A_{44} = \frac{-b_2\hat{J}_0}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$B_{31} = \frac{\hat{J}_2}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$B_{41} = \frac{m_2 L_1 l_2}{\left(\hat{J}_0\hat{J}_2 - m_2^2 L_1^2 l_2^2\right)}$$

$$\begin{cases}\hat{J}_0 = J_1 + m_2 \cdot L_1^2\\ \hat{J}_2 = J_2\end{cases}$$

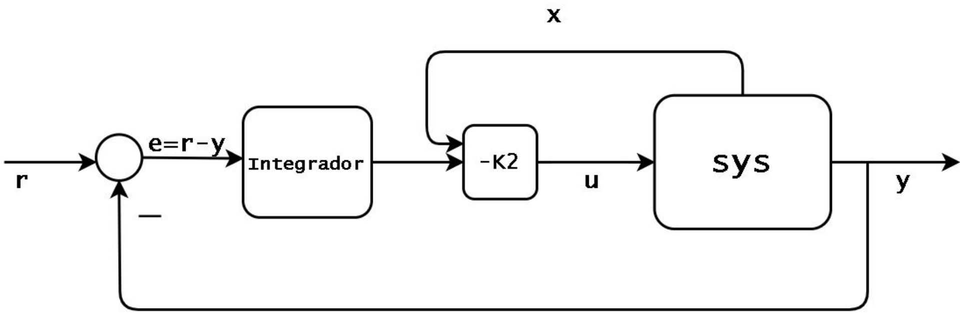
Linealización

$(*)$ *Para nuestro caso se toma* $\tau_2 = 0$

Control LQI

(Linear Quadratic Integral)

Tiempo continuo

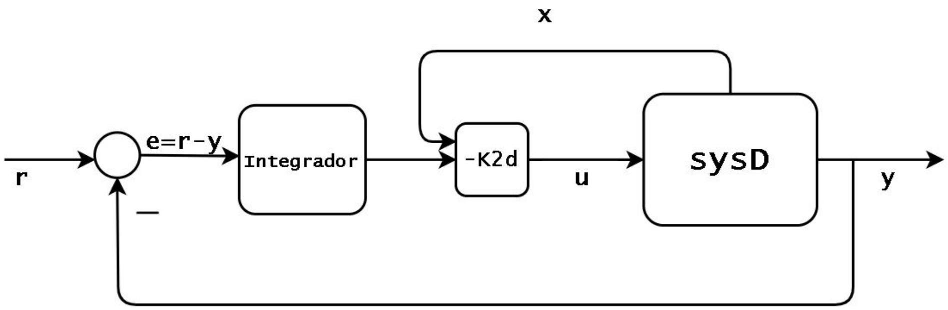



```
% control LQI - Tiempo Continuo
sys = ss(A,B,C,0);
Q2 = diag([1 7 1 7 0.05]);
R2 = 1;
[K2,S2,e2] = lqi(sys,Q2,R2);
Aa = [A zeros(4,1);-C 0];
Ba = [B;0];
poles2 = eig(Aa-Ba*K2);
```

Control LQI

(Linear Quadratic Integral)

Tiempo discreto




```matlab
% Tiempo Discreto
Ts = 10e-3;
sys = ss(A,B,C,0);
sysD = c2d(sys,Ts,'zoh');
Ad = sysD.a;
Bd = sysD.b;
Cd = sysD.c;
% control LQI - Tiempo Discreto
Q2d = diag([1 7 1 7 0.5]);
R2d = 1;
[K2d,S2d,e2d] = lqi(sysD,Q2d,R2d);
Aad = [Ad zeros(4,1);-Cd*Ts 1];
Bad = [Bd;0];
poles2d = eig(Aad-Bad*K2d);
```

Control LQI con observador

Tiempo continuo

```
% LQI con observador - Tiempo continuo
Aw = A;
Bw = B;
Aaa = Aw(1:2,1:2);
Aab = Aw(1:2,3:4);
Aba = Aw(3:4,1:2);
Abb = Aw(3:4,3:4);
Ba = Bw(1:2);
Bb = Bw(3:4);
% Para el calculo de L
Ao = Abb;
Co = Aab;
Lobs = place(Ao',Co',10*[poles2(1) poles2(1)]);
Ke = Lobs';
poles3_obs = eig(Ao-Ke*Co);
% Matrices equivalentes para simulink
A_h = Abb - Ke*Aab;
B_h = A_h*Ke + Aba - Ke*Aaa;
F_h = Bb - Ke*Ba;
C_h = [0 0;0 0;1 0;0 1];
D_h = [1 0;0 1;Ke];
```

Hacemos modo reducido

$-Mido\ \alpha\ y\ \beta$

$-Estimo\ \dot{\alpha}\ y\ \dot{\beta}$

Control LQI con observador

Tiempo discreto

```
% LQI con observador - Tiempo discreto
Aw = Ad;
Bw = Bd;
Aaa = Aw(1:2,1:2);
Aab = Aw(1:2,3:4);
Aba = Aw(3:4,1:2);
Abb = Aw(3:4,3:4);
Ba = Bw(1:2);
Bb = Bw(3:4);
% Para el calculo de L
Aod = Abb;
Cod = Aab;
Lobs = place(Aod',Cod',[poles2d(1) poles2d(1)].^10);
Ked = Lobs';
poles3d_obs = eig(Aod-Ked*Cod);
% Matrices equivalentes para simulink
Ad_h = Abb - Ked*Aab;
Bd_h = Ad_h*Ked + Aba - Ked*Aaa;
Fd_h = Bb - Ked*Ba;
Cd_h = [0 0;0 0;1 0;0 1];
Dd_h = [1 0;0 1;Ked];
```

Hacemos modo reducido

$-Mido\ \alpha\ y\ \beta$

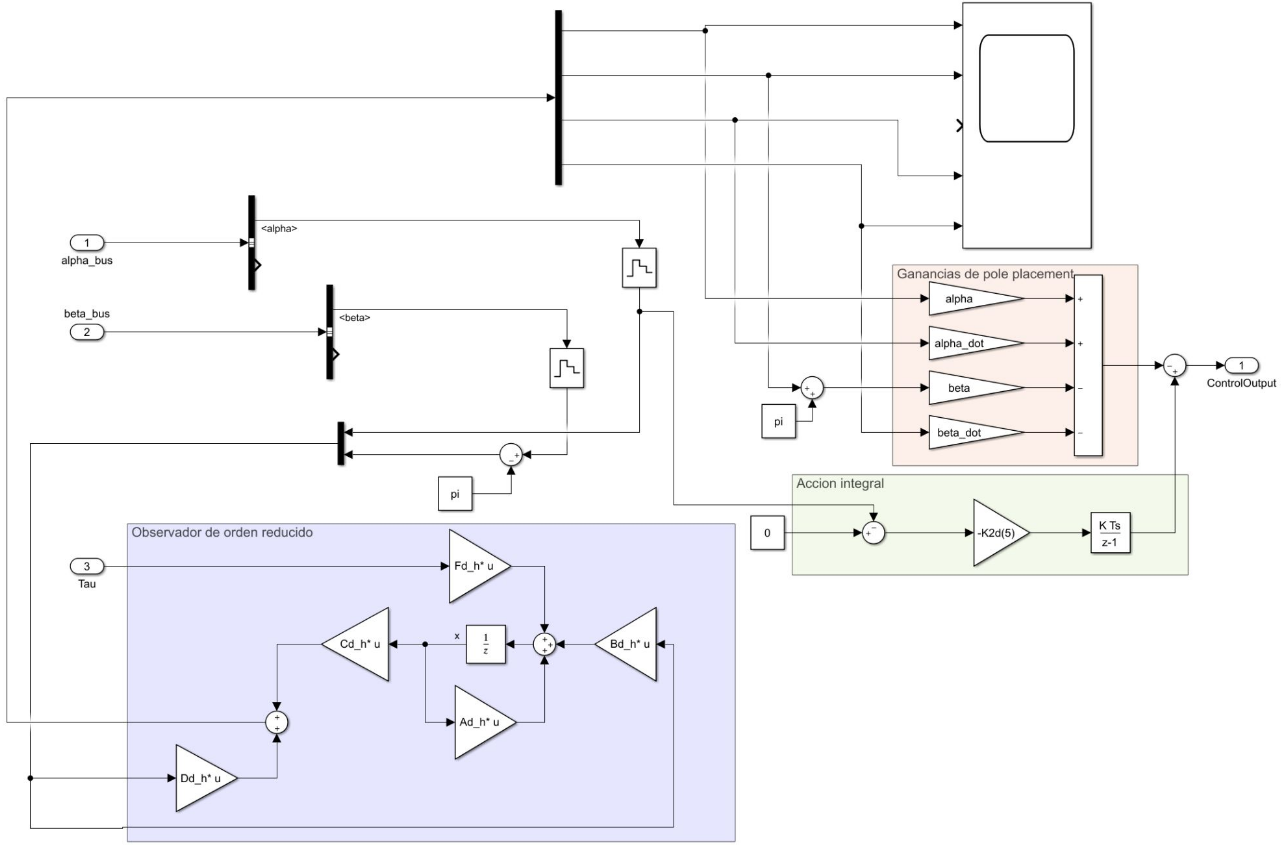
$-Estimo\ \dot{\alpha}\ y\ \dot{\beta}$

Pendulum_CAD_Control2293_PolePlacementV2_Discreto

Control LQI con observador

Tiempo discreto

Pendulum_CAD_Control2293_PolePlacementV2_Discreto >> Controller >> StayUp_Control



Control LQI con observador

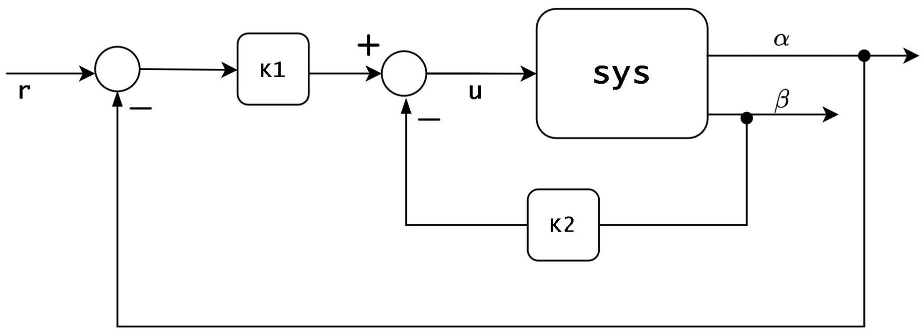
Implementation en uC

```
%% Implementacion en uC
% Realimentaci n de estados uC
ka = K2(1:2);
kb = K2(3:4);
A_mn = A_h - F_h*kb;
B_mn = B_h - F_h*(ka+kb*Ke);
C_mn = -kb;
D_mn = -(ka+kb*Ke);
Obs_TF = -(C_mn*((s*eye(2)-A_mn)^-1)*B_mn + D_mn);
Obs_TF = minreal(Obs_TF,0.01);
aOBS = cell2mat(Obs_TF.Numerator);
aOBS = [aOBS(1:2); aOBS(3:4)];
bOBS = cell2mat(Obs_TF.Denominator);
bOBS = bOBS(1:2); % El denominador va una sola vez
[Aobs2, Bobs2, Cobs2, Dobs2] = tf2ss(aOBS,bOBS);
aux_obs = ss(Aobs2, Bobs2, Cobs2, Dobs2);
aux_obsD = c2d(aux_obs,Ts2,'zoh');
obs_uCa = aux_obsD.a;
obs_uCb = aux_obsD.b;
obs_uCc = aux_obsD.c;
obs_uCd = aux_obsD.d;
```

Pendulum_CAD_Control2293_PolePlacementV2_uC

LoopShaping

(Control en cascada)



Tiempo Continuo

```
%% Loop Shaping
s = tf('s');
[n,d] = ss2tf(A,B,[0 1 0 0],0);
Ga = minreal(zpk(tf(n,d)),0.01); % Sale con beta
% Para beta armamos un PD Regularizado
C2 = minreal((s+9.518)/(1+(s/50)),0.01);
Lazo2 = minreal(C2*Ga,0.01);
figure();
bode(Lazo2);
K12 = 1/db2mag(-12);
Lazo2 = Lazo2*K12;
figure();
nyqlog(Lazo2);
```

LoopShaping

Tiempo Continuo

```
C2 = ss(C2);
C2.u='beta';
C2.y='ut';
% Interconectamos C2 con el Sistema Completo:
% Nos queda el sistema con C2 enganchado
% que realimenta el angulo "beta"
Gss=ss(A,B,[1 0 0 0;0 1 0 0],[0;0]);
Gss.u='u';
Gss.y='y';
Sum = sumblk('u = ut + up');
Sys=ss([],[],[],[1 0]);
Sys.u='y';
Sys.y='alfa';
Sys2=ss([],[],[],[0 1]);
Sys2.u='y';
Sys2.y='beta';
Gb=connect(Gss,C2,Sum,Sys,Sys2,'up','alfa');
Gb = zpk(Gb);
Gb = minreal(Gb,0.01);
```

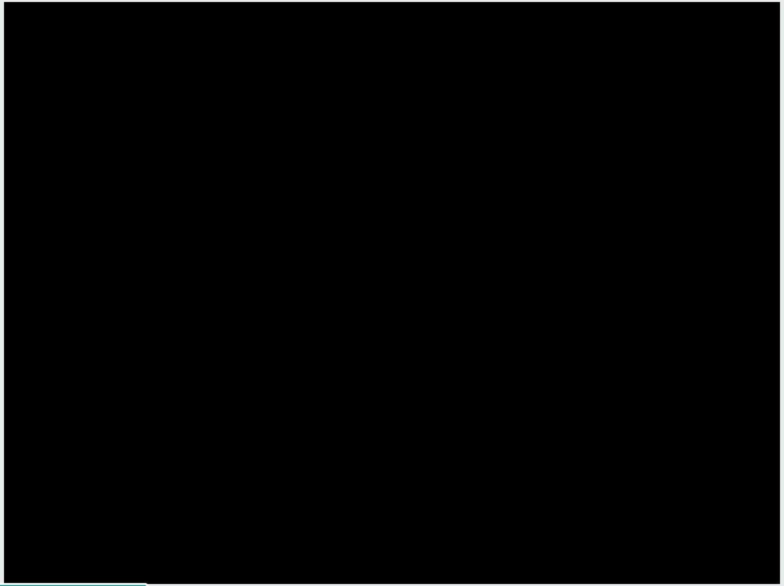
```
% Para alfa armamos un PID
close all;
C1p = minreal((1/s)*
(s+9.518)*(s^2 + 40.52*s + 631)/
(6.94*(s+8.696)*(s+50)*(s+8.719)),0.01);
zpk(minreal(C1p*zpk(Gb),0.01));
C1pp = -((s+0.15)^2/(1+(s/50)));
C1 = minreal(C1p*C1pp);
Lazo1 = minreal(C1*Gb,0.01);
figure();
bode(Lazo1);
Kl1 = 1/db2mag(0);
Lazo1 = Lazo1*Kl1;
figure();
nyqlog(Lazo1);
close all;
```

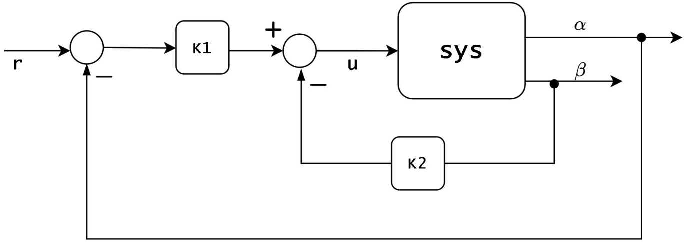
LoopShaping

(Control en cascada)

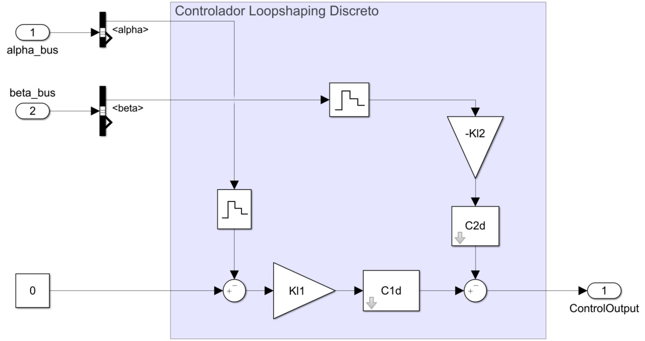
Tiempo Discreto

Pendulum_CAD_Control2293_LoopShaping_Discreto





Pendulum_CAD_Control2293_LoopShaping_Discreto >> Controller >> StayUp_Control

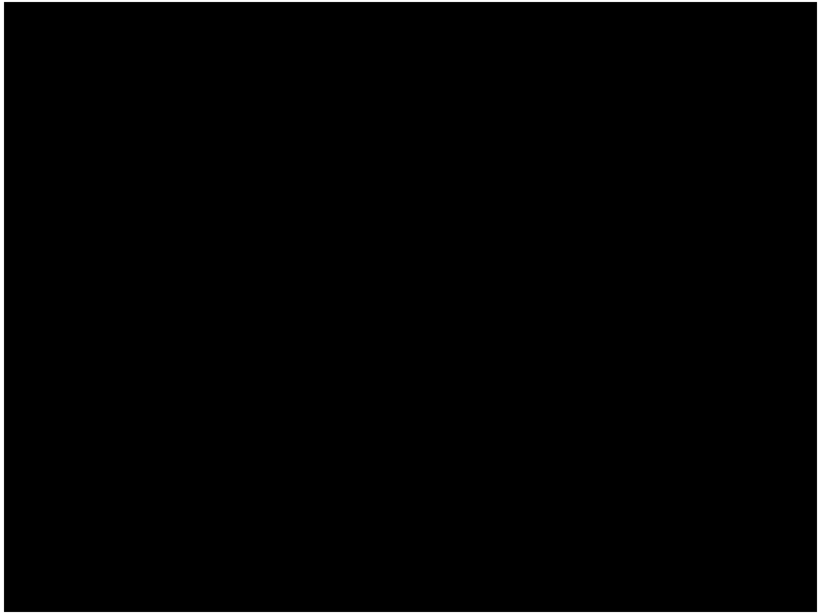



```
% Tiempo Discreto
% Polo mas lejos: -50 => 8Hz
% Para cumplir Nyquist:
% Al menos muestrear a 16Hz => 6.25ms
Ts2 = 1e-3; % Funciona bien
C2d = c2d(C2_aux,Ts2,'zoh');
C1d = c2d(C1,Ts2,'zoh');
```

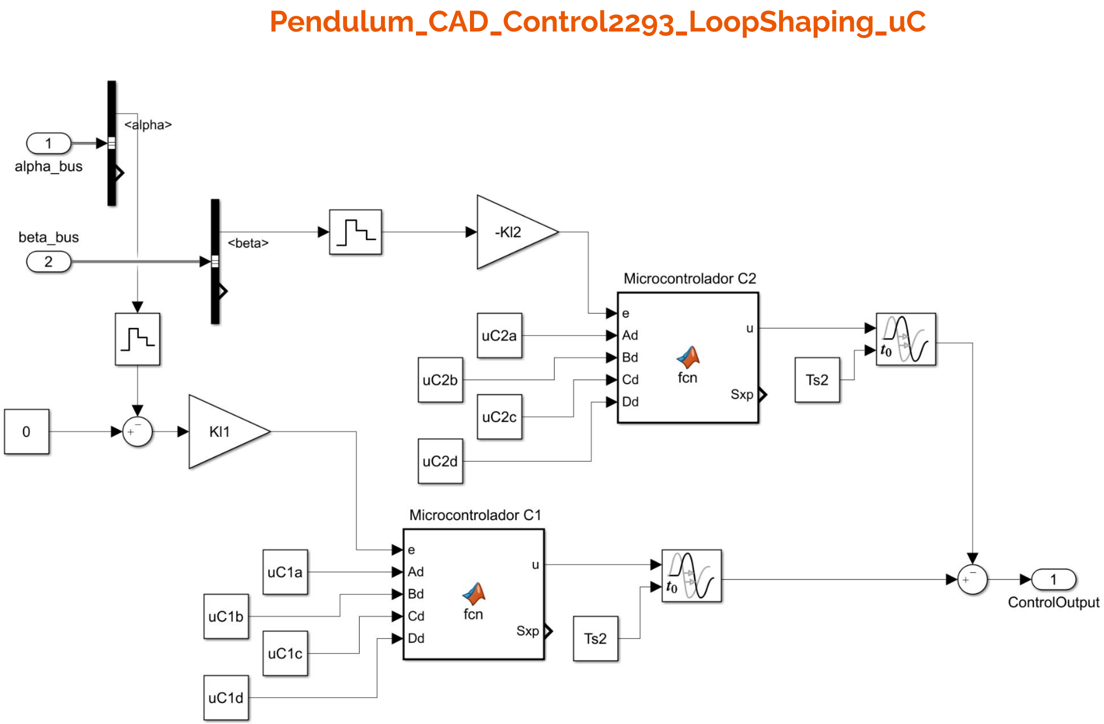
LoopShaping
Implementation con uC

```
%% Implementacion en uC
% Loop Shaping uC
C1 = tf(C1);
[Ac1, Bc1, Cc1, Dc1] =
tf2ss(cell2mat(C1.Numerator),
      cell2mat(C1.Denominator));
% Queda de dimension 5 porque son 5 polos
aux1 = ss(Ac1,Bc1,Cc1,Dc1);
aux1d = c2d(aux1,Ts2,'zoh');
uC1a = aux1d.a;
uC1b = aux1d.b;
uC1c = aux1d.c;
uC1d = aux1d.d;
C2_aux = tf(C2_aux);
[Ac2, Bc2, Cc2, Dc2] =
tf2ss(cell2mat(C2_aux.Numerator),
      cell2mat(C2_aux.Denominator));
% Queda de dimension 1 porque es un solo polo
aux2 = ss(Ac2,Bc2,Cc2,Dc2);
aux2d = c2d(aux2,Ts2,'zoh');
uC2a = aux2d.a;
uC2b = aux2d.b;
uC2c = aux2d.c;
uC2d = aux2d.d;
```

Pendulum_CAD_Control2293_LoopShaping_uC



LoopShaping
Implementation con uC



¡Muchas gracias!

¿Preguntas?