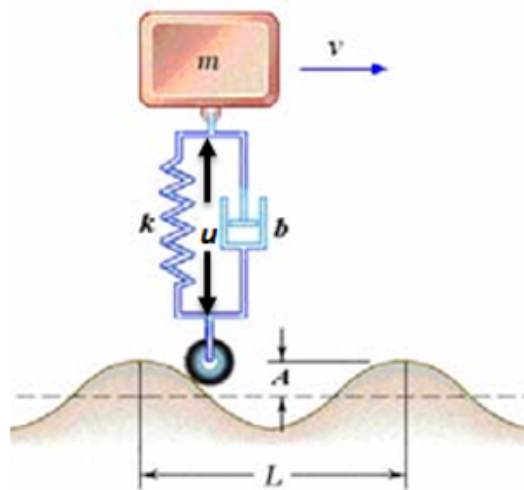


TALLER SEGUNDO CORTE

- Laura Camila Machado Prada
- Nestor Javier Mahecha Parra



MODELO

- $-M\ddot{y} = K(y + P) + B(\dot{y} + \dot{P}) - u(t)$

ECUACIONES

- $Z_2 = \dot{Z}_1$
- $-M\dot{Z}_2 = K(Z_1 + P) + B(Z_2 + \dot{P}) - u(t)$

ECUACIÓN EN MATRICES DE ESTADO

```
% clc
% clear all
% syms K M b s
% A1=[0 1;
%      -K/M -b/M]
% B1=[0; 1/M]
% C1=[1 0] %Controlar posición
% D1=0
% E1=[0; -K/M]
% F1=[0; -b/M]
% funcion=(C1*(s*eye(2)-A1)^(-1))*B1
%
% %ecuacion=
```

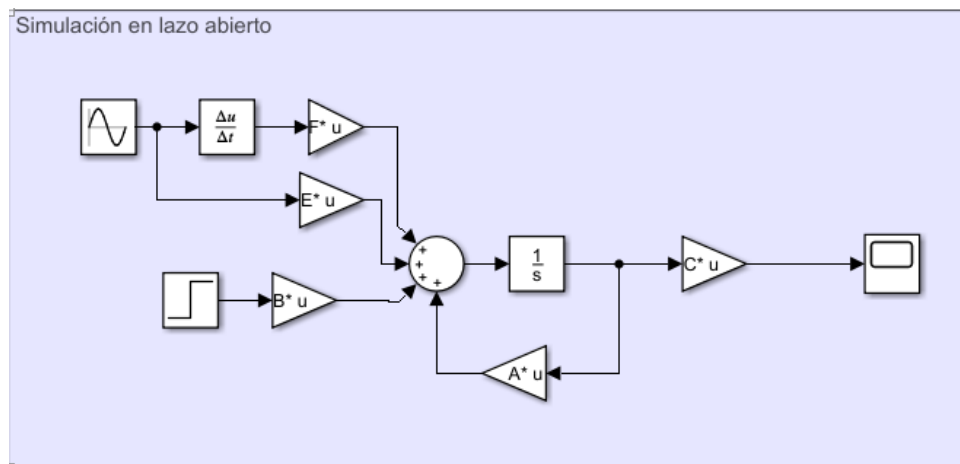
```

% syms s
% b=10;
% K=20;
% M=3;
% A=[0    1;
%    -K/M -b/M]
% B=[0; 1/M]
% C=[1 0]
% D=0
% E=[0; -K/M]
% F=[0; -b/M]
% [num,den]=ss2tf(A,B,C,D)
% sys=tf(num,den)
% El tiempo de la perturbación es 8s, por tener una frecuencia de pi/4 rad/s

```

- **SIMULACIÓN DE LA PLANTA EN LAZO ABIERTO**

Esquema de simulación



Características de la perturbación

Block Parameters: Sine Wave

Sine Wave

Output a sine wave:

$$O(t) = \text{Amp} * \sin(\text{Freq} * t + \text{Phase}) + \text{Bias}$$

Sine type determines the computational technique used. The parameters in the two types are related through:

$$\text{Samples per period} = 2 * \pi / (\text{Frequency} * \text{Sample time})$$

$$\text{Number of offset samples} = \text{Phase} * \text{Samples per period} / (2 * \pi)$$

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

Parameters

Sine type: Time based

Time (t): Use simulation time

Amplitude: 0.5

Bias: 0

Frequency (rad/sec): $\pi/4$

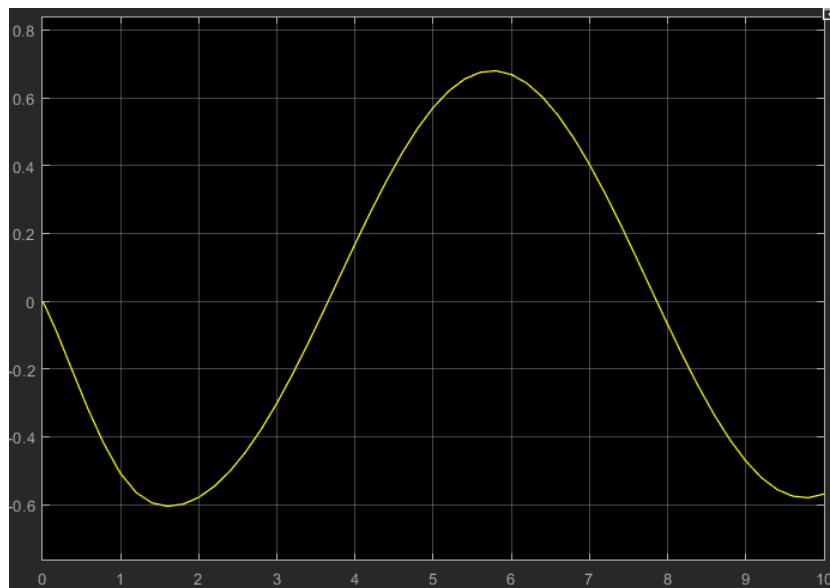
Phase (rad): 0

Sample time: 0

☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Respuesta en lazo abierto



- PID CONTINUO

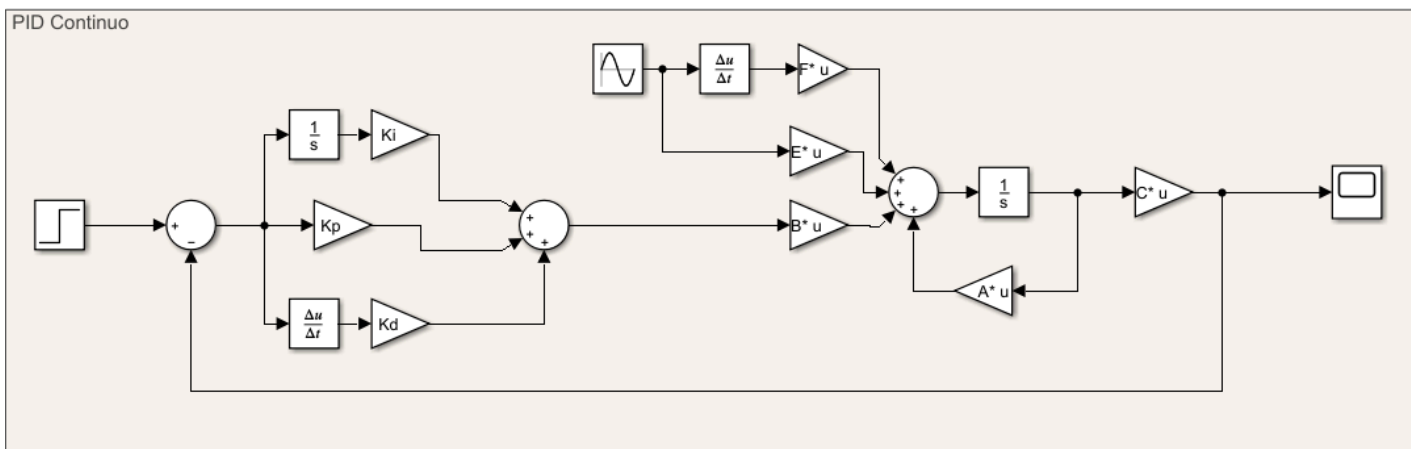
```
% syms x kd kp ki z
% numc=kd*s^2+kp*s+ki;
% denc=s;
% %Separar denominador y numerador de la funcion de transferencia y dejarlo
```

```

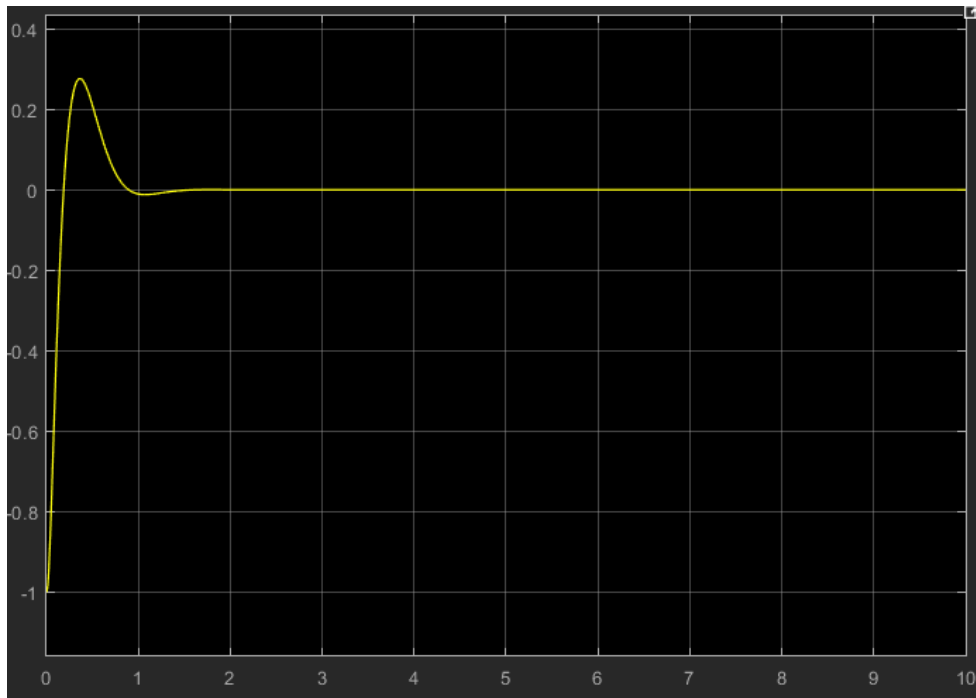
% %con s
% [numdis,dendis]=tfdata(sys,'v')
% NumeradorR=poly2sym(numdis,s)
% DenominadorR=poly2sym(dendis,s)
% %hallar denominador del polinomio caracteristico
% pc=collect(NumeradorR*numc+DenominadorR*denc)
% %hallar polinomio deseado
% zd=0.7;
% tsd=1;
% wnd=4.6/(zd*tsd)
% %Manejando simbolicos polinomio
% pd=expand((s^2+2*wnd*zd*s+wnd^2)*(s+5*wnd*zd))
% Pccoef=coeffs(pc,s)
% Pdcoef=coeffs(pd,s)
% Sol=double(vpa(struct2array(solve(Pdcoef==Pcccoef,[ki kp kd])),2))
% Ki=Sol(1)
% Kp=Sol(2)
% Kd=Sol(3)

```

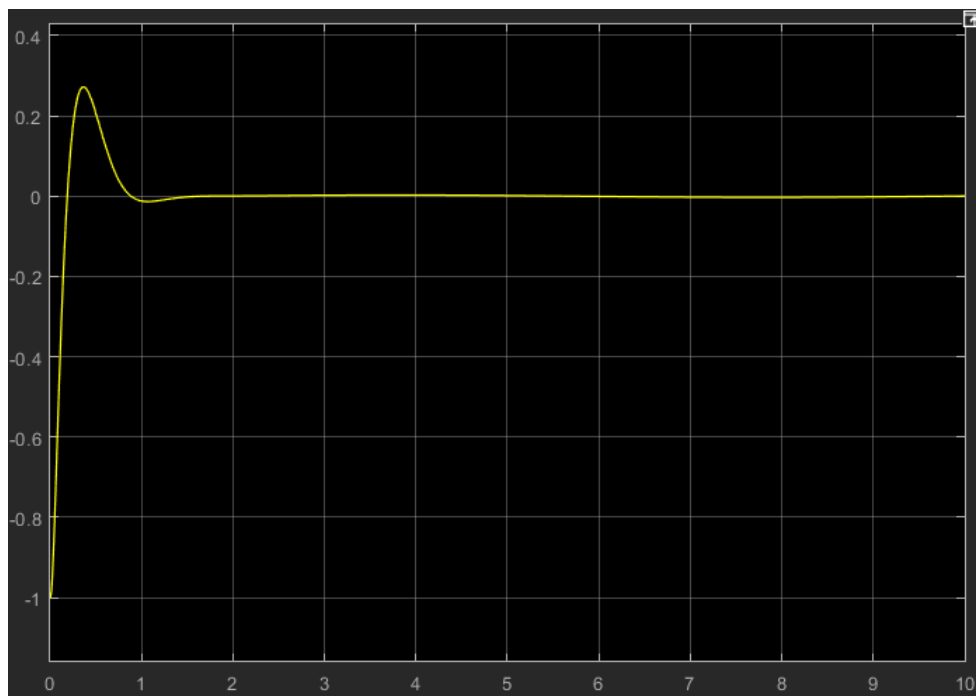
Esquema de simulación



Respuesta sin perturbaciones



Respuesta con perturbaciones



Ecuación en diferencias PID Continuo

$$u(k) = 86.6 * e(k) + 744.351 * e(k - 1) + 2.9797 * 10^3 * e(k - 2)$$

PID CONTINUO DISCRETO

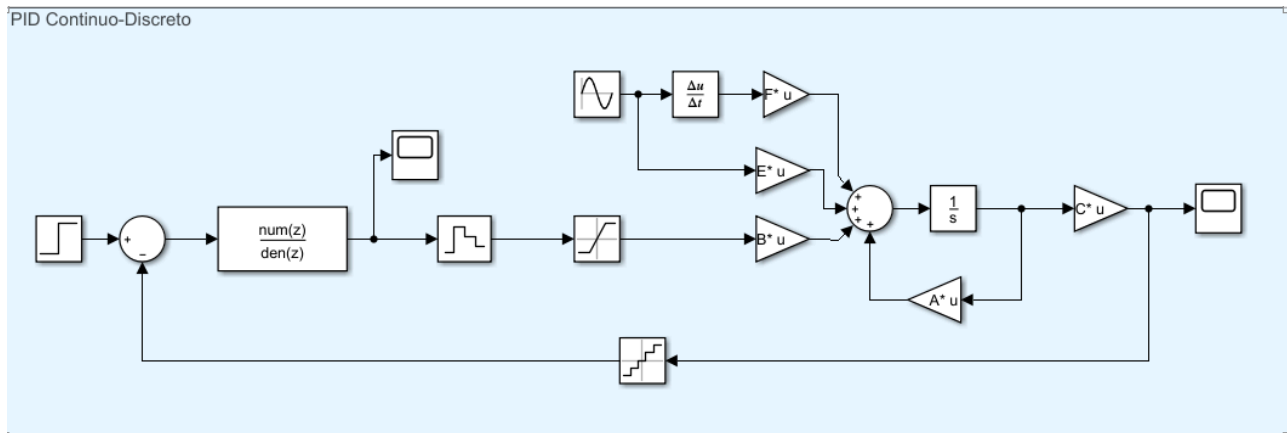
```
% %continuo a discreto
% numcd=[Kd Kp Ki]
% dencd=[1 0]
% ts=tsd/40;
```

```

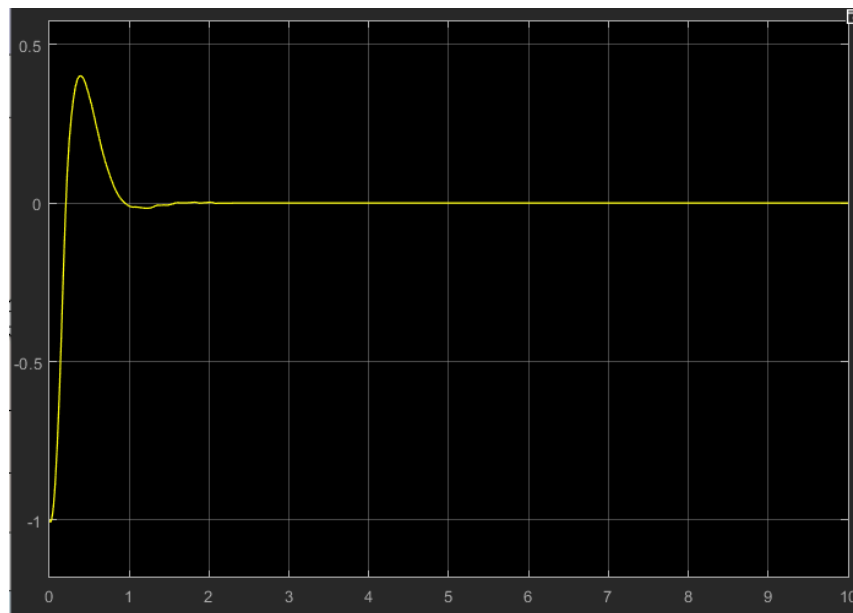
% td=Kd/Kp
% ti=Kp/Ki
% q0=Kp*(1+(ts/(2*ti)))+(td/ts))
% q1=Kp*((ts/(2*ti))-1-((2*td)/ts))
% q2=Kp*(td/ts)
% N1=[q0 q1 q2]
% D1=[1 -1]

```

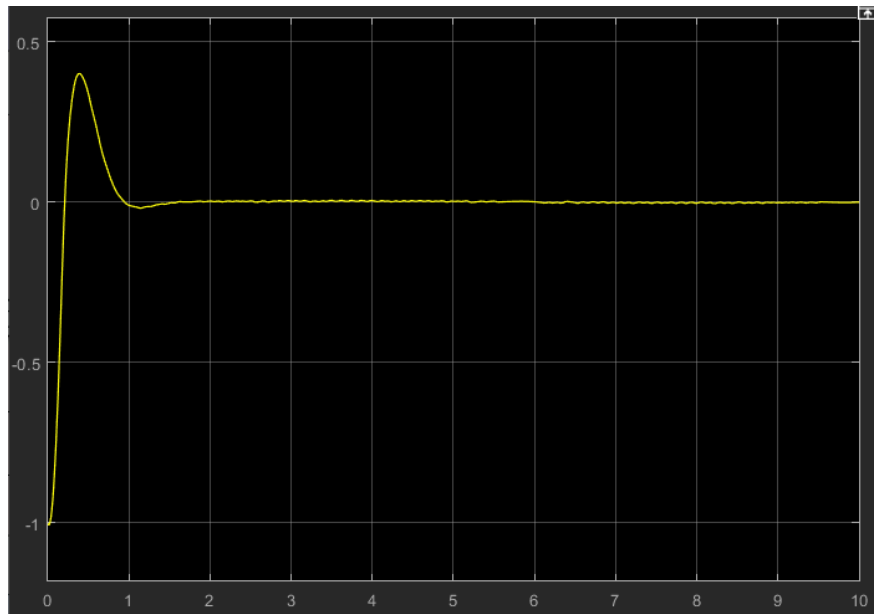
Esquema de simulación



Respuesta sin perturbaciones



Respuesta con perturbaciones



Ecuación en diferencias PID Continuo- Discreto

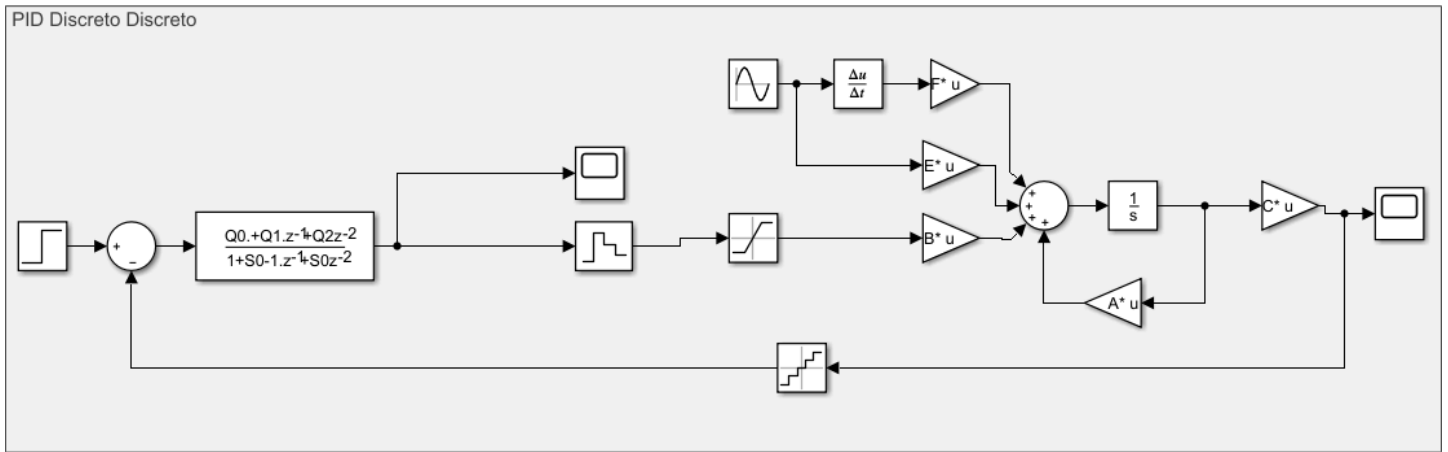
$$u(k) = 4.2456 \cdot 10^3 \cdot e(k) - 7.6351 \cdot 10^3 \cdot e(k-1) + 3.4640 \cdot 10^3 \cdot e(k-2) + u(k-1)$$

PID DISCRETO DISCRETO

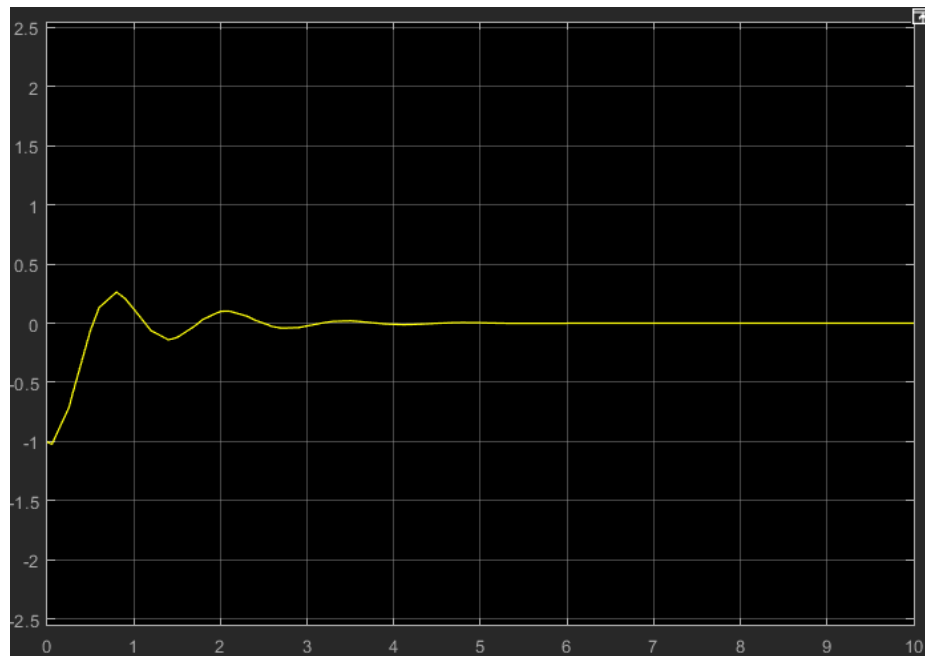
```
% zd=0.7;
% tsd=3;
% tm=tsd/10;
% wnd=4.6/(zd*tsd)
% syms q0 q1 q2 s0 z
% funciond=c2d(sys,tm,'zoh')
% numd=[q0 q1 q2];
% dend=[1 s0-1 -s0 ];%(z-1)*(z+S0)
% pcnum=poly2sym(numd,z);
% pcden=poly2sym(dend,z);
% [numdpd,dendpd]=tfdata(funciond,'v');
% numpd=vpa((poly2sym(numdpd,z)),4);
% denpd=vpa((poly2sym(dendpd,z)),4);
% planta=numpd/denpd;
% controlador=pcnum/pcden
% controladorplanta=(simplify((planta*controlador)/(1+(planta*controlador))));
% [u,j]=numden(controladorplanta);
% u1=coeffs(j,z,'All');
% u2=vpa(u1/u1(1),2)
% %%Polinomio deseado
% PD_Cont = [1 (2*zd*wnd) wnd^2]          %polinomio deseado en continuo
% FT_PD = tf(1,PD_Cont);
% FT_PD_D = c2d(FT_PD, tm, 'zoh')
% PD_Disc = FT_PD_D.den{1};
% PD_PID = conv(PD_Disc, conv([1 -0.1], [1 -0.1]))
% %Solución
% sold=struct2array((solve(u2==PD_PID,[q0 q1 q2 s0])))
% Q0=double(sold(1))
% Q1=double(sold(2))
```

```
% Q2=double(sold(3))
% S0=double(sold(4))
```

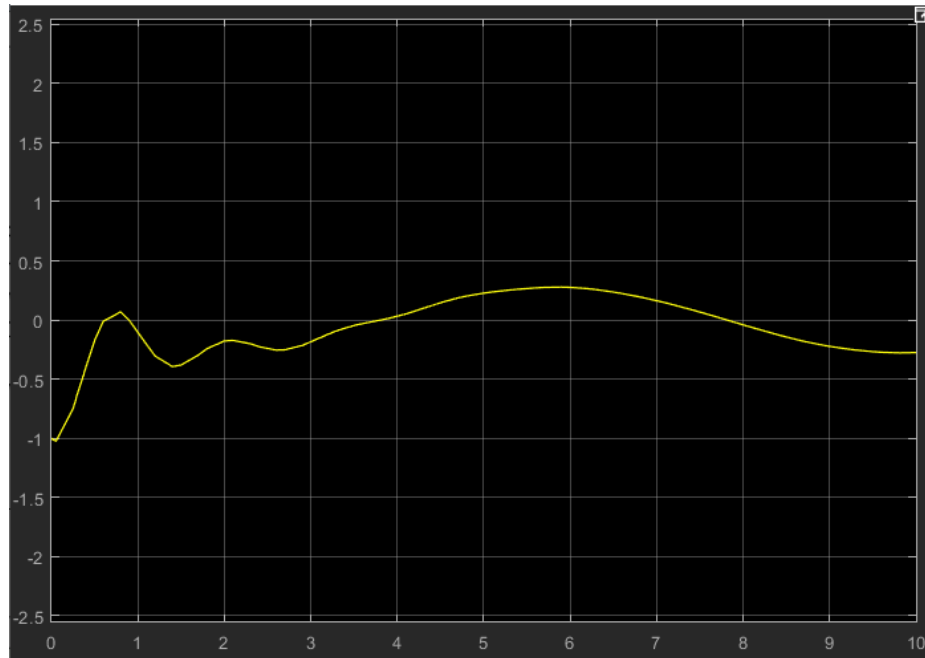
Esquema de simulación



Respuesta sin oscilaciones



Respuesta con oscilaciones



Ecuacion en diferencias discreto discreto

$$u(k) = 36.222 * e(k) - 39.1747 * e(k-1) + 15.179 * e(k-2) + 0.7007 * u(k-1) - 0.2993 * u(k-2)$$

CONTROLADOR POR LGR

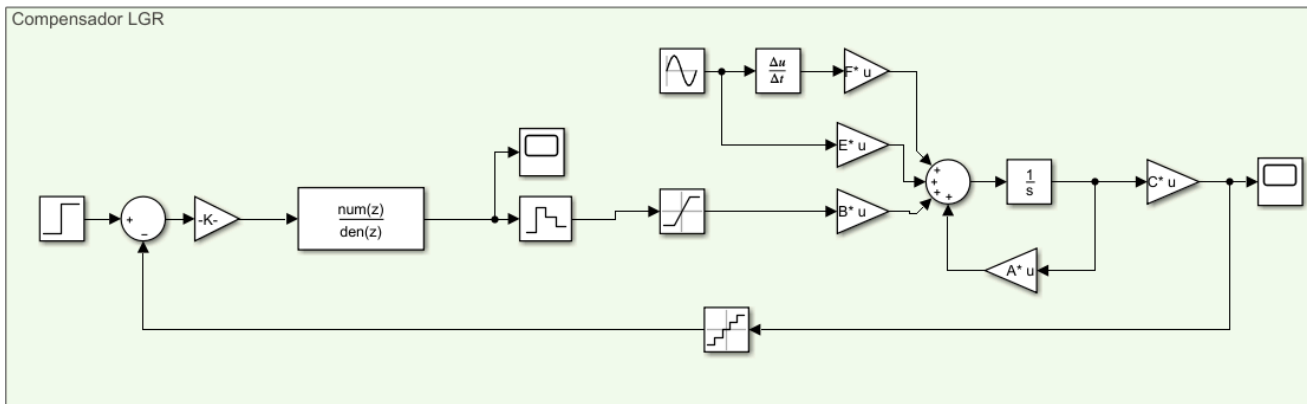
```
% syms z
% tsd=3;
% zd=0.7;
% sysd=c2d(sys,0.3,'zoh') %Por el tiempo de establecimiento en 3
% sysd2=zpk(sysd)
% [cero,polo,gananciaplanta]=zpkdata(sysd)
% ceros=cell2mat(cero)
% polos=cell2mat(polo)
% T=tsd/10;
% wn=4/(tsd*zd)
% ws=2*pi/T
% wd=wn*sqrt(1-(zd)^2)
% %Punto de prueba
% ppm=exp(-T*zd*wn);
% ppf=T*wd;
% zp= complex(ppm*cos(ppf),ppm*sin(ppf))
% %Encontrar ángulos
% %Cero
% %Como dan ángulos conjugados entonces
% v1=[zp]-[ceros];
% angulo1=angle(v1)*180/pi
% %Polo 1
% v2=[zp]-real(polos(1))-imag(polos(1))*j
% angulo2=angle(v2)*180/pi
% %Polo 2
% v3=[zp]-real(polos(2))+imag(polos(2))*j
% angulo3=angle(v3)*180/pi
```

```

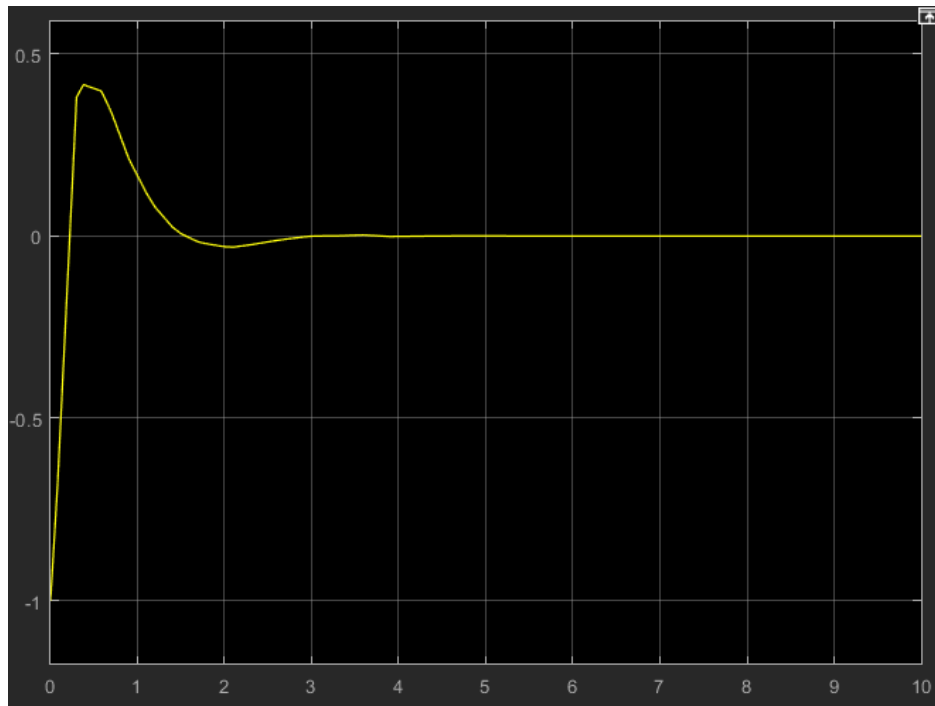
% %Polo en 1 dado por el integrador
% v4=zp-1;
% angulo4=angle(v4)*180/pi
% %Ángulo a compensar
% compensar=abs(angulo1-angulo2-angulo3-angulo4+180)
% %Ángulo B
% anguloB=(angulo1-angulo4+180)
% %Encontrar B
% B=-imag(zp)/tand(anguloB)+real(zp)
% %Encontrar K
% ganancia=real(1/abs(((gananciaplanta*(zp-ceros)/((zp-B)*(zp-1))))))
% %Compensador
% numerador=conv([1 -polos(1)],[1 -polos(2)])
% denominador=([1 -(B+1) B])
% %K*(z-polos(1))*(z-polos(2))/(z-B)
%

```

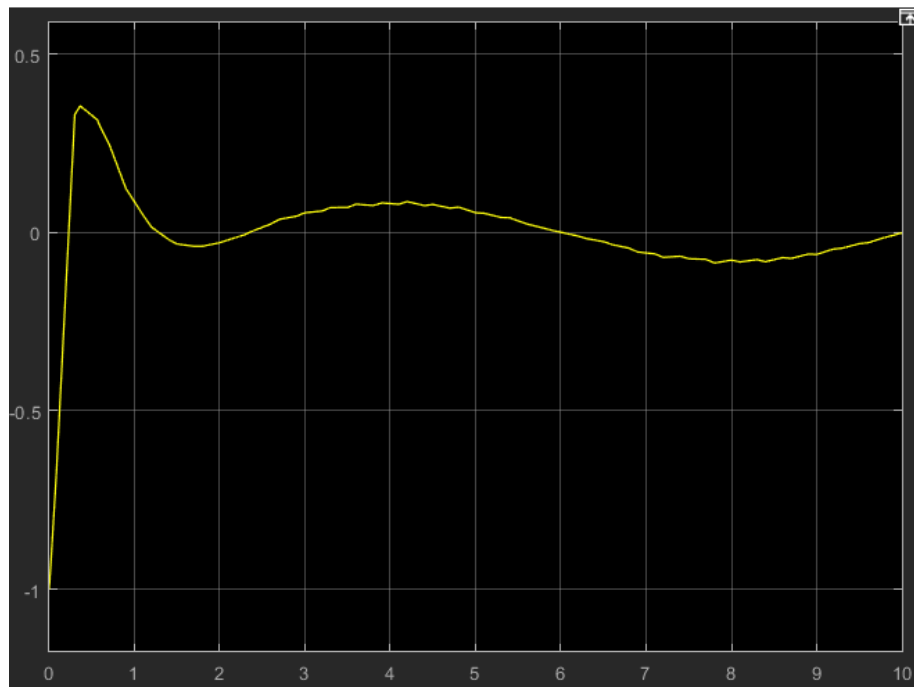
Esquemas de simulación



Respuesta sin oscilaciones



Respuesta con oscilaciones



Ecuaciones en diferencia LGR

$$u(k) = 12.12 * e(k) - 12.2 * e(k-1) + 4.459 * e(k-2) + 1.3582 * u(k-1) - 0.3582 * u(k-2)$$

COMPENSADOR POR ANULACIÓN DE PLANTA

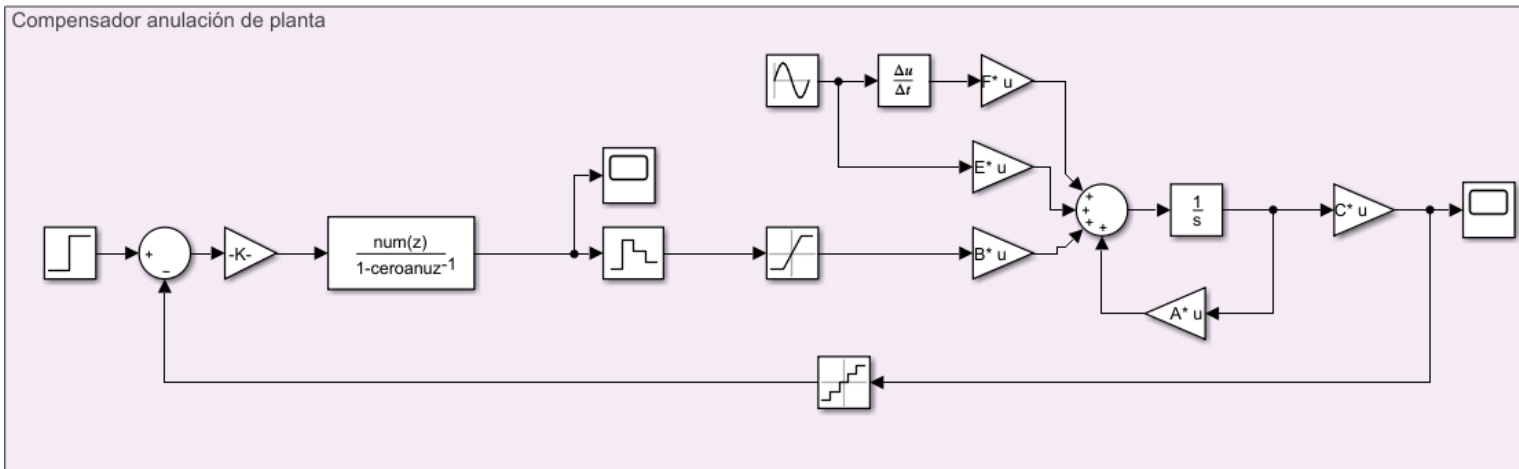
```
% El periodo de muestreo es tsd/10 y T=tsd/5
% M retroalimentado en Z
% tsd=0.7;
% tm=tsd/10;
```

```

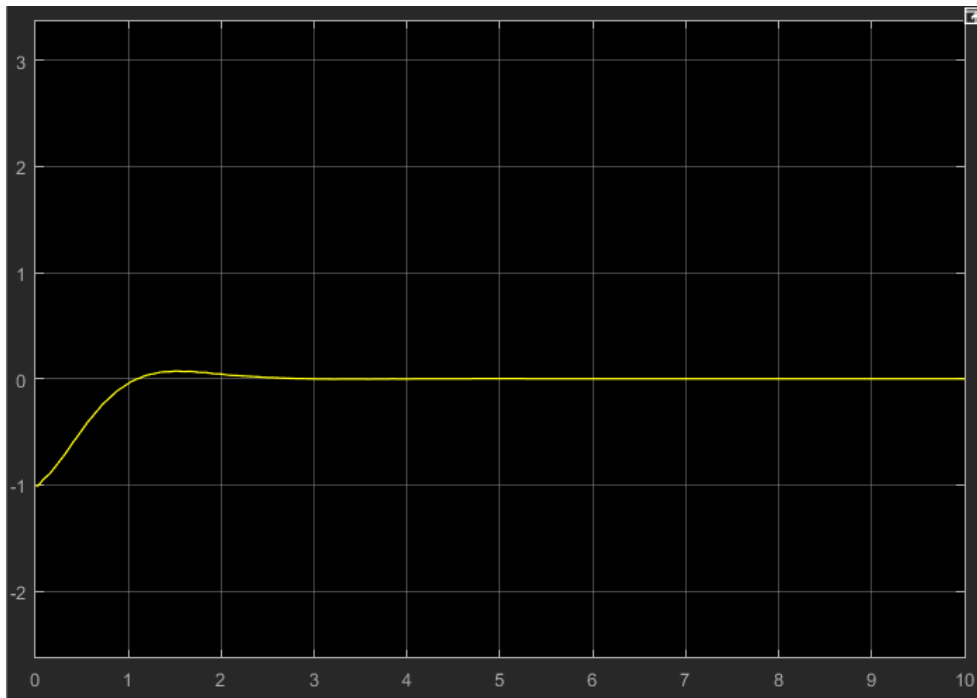
% nummrz=[1-exp(-1/2)];
% denmrz=[1 -1]
% %Planta discretizada
% sysd=c2d(sys,tm,'zoh')
% sysd2=zpk(sysd)
% [Zi,Pi,Ki]=zpkdata(sysd);
% %Estos van con signo contrario al igualarlos a cero
% ceroanu=cell2mat(Zi)
% Pi=cell2mat(Pi)
% numanu=conv([ 1 -Pi(1)],[1 -Pi(2)])
% denanu=conv([1 -ceroanu],[1 -1])
% %hallar K
% gananciaanu=nummrz/Ki
% %Ecuacion en diferencias
% %uk=K(ek-Pi(2)*ek-1)-(-Zi)*uk-1

```

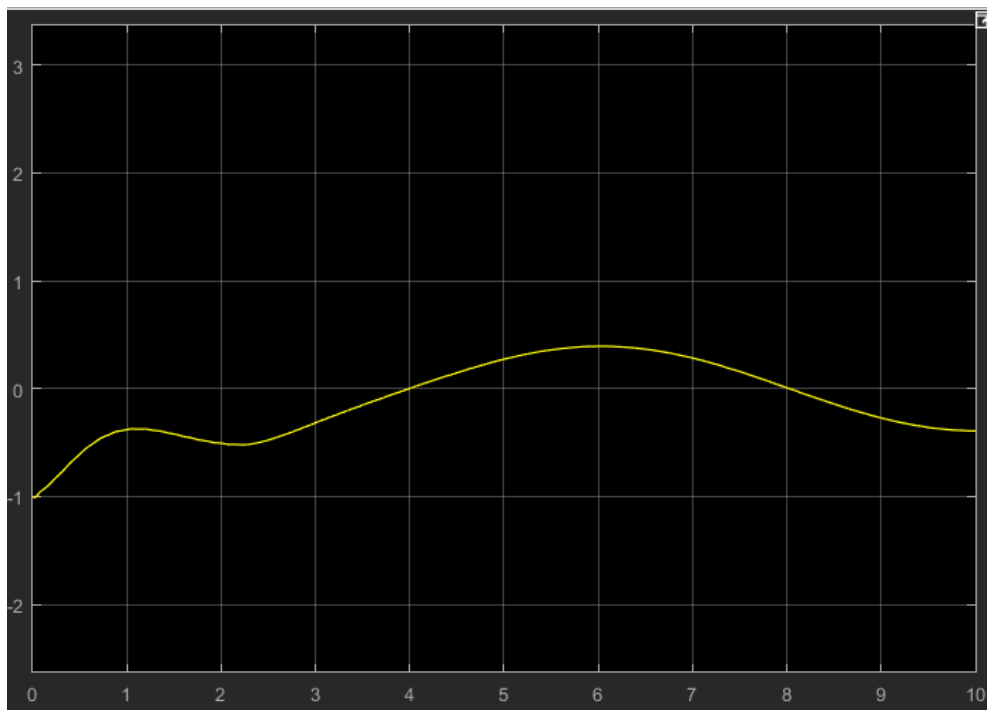
Esquema de simulación



Respuesta sin perturbación



Respuesta con perturbación



Ecuación en diferencias anulación de planta

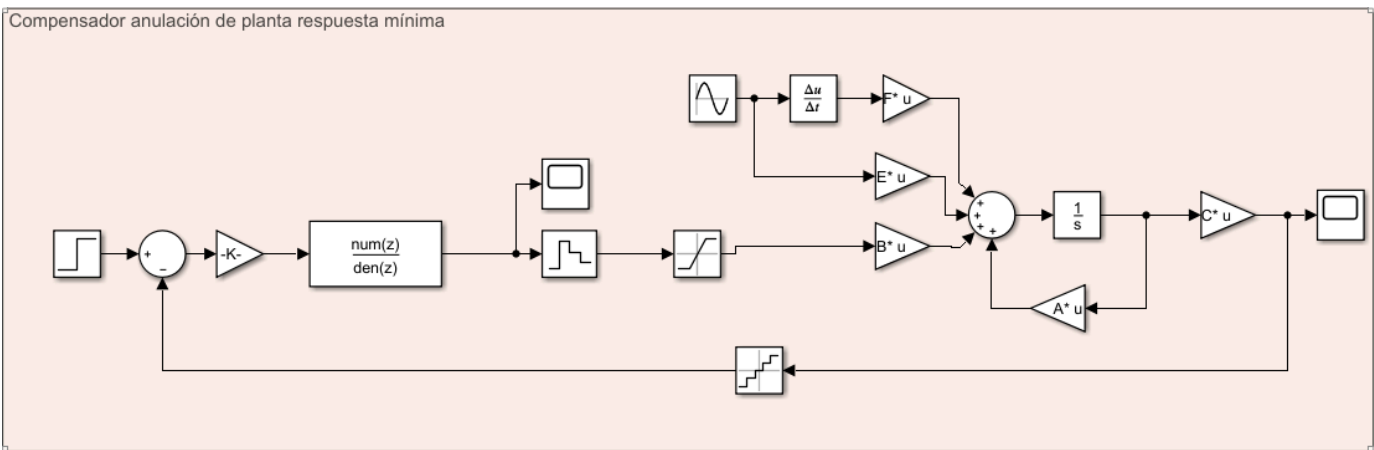
$$u(k) = 403.8 * e(k) - 698 * e(k - 1) + 309.3 * e(k - 2) + 0.0851 * u(k - 1) + 0.9149 * u(k - 2)$$

COMPENSADOR POR ANULACIÓN DE PLANTA RESPUESTA MÍNIMA

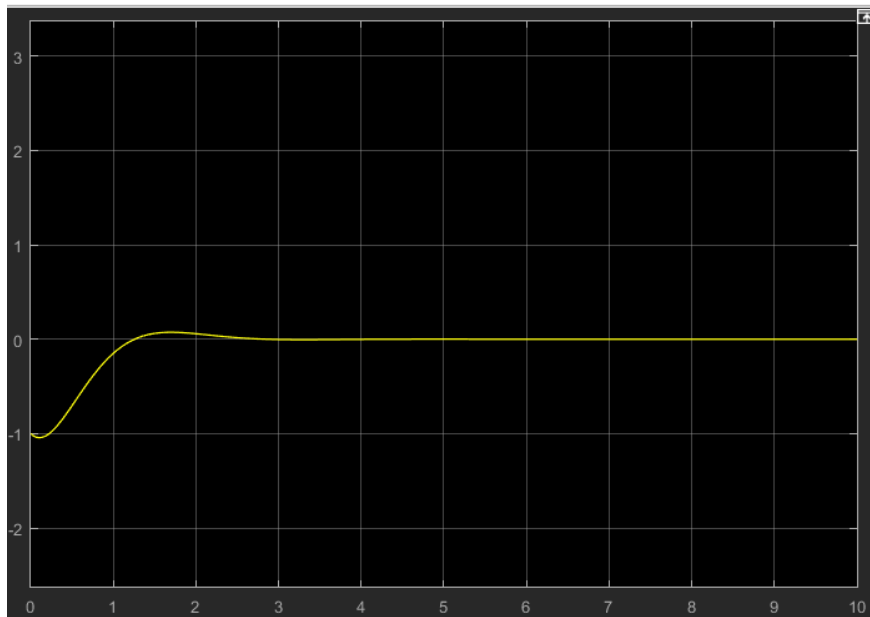
```
% El periodo de muestreo es tsd/10 y T=tsd/5
% M retroalimentado en Z
% tsd=0.7;
```

```
% tm=tsd/10;  
% %Planta discretizada  
% sysd=c2d(sys,tm,'zoh')  
% sysd2=zpk(sysd)  
% [Zi2,Pi2,Ki2]=zpkdata(sysd);  
% %Estos van con signo contrario al igualarlos a cero  
% cerores=cell2mat(Zi2)  
% Pi2=cell2mat(Pi2)  
% numres=conv([ 1 -Pi2(1)],[1 -Pi2(2)])  
% denres=conv([1 -cerores],[1 -1])  
% %hallar K  
% ganancias=1/Ki2  
% %Ecuacion en diferencias  
% %uk=K(ek-Pi(2)*ek-1)-(-Zi)*uk-1
```

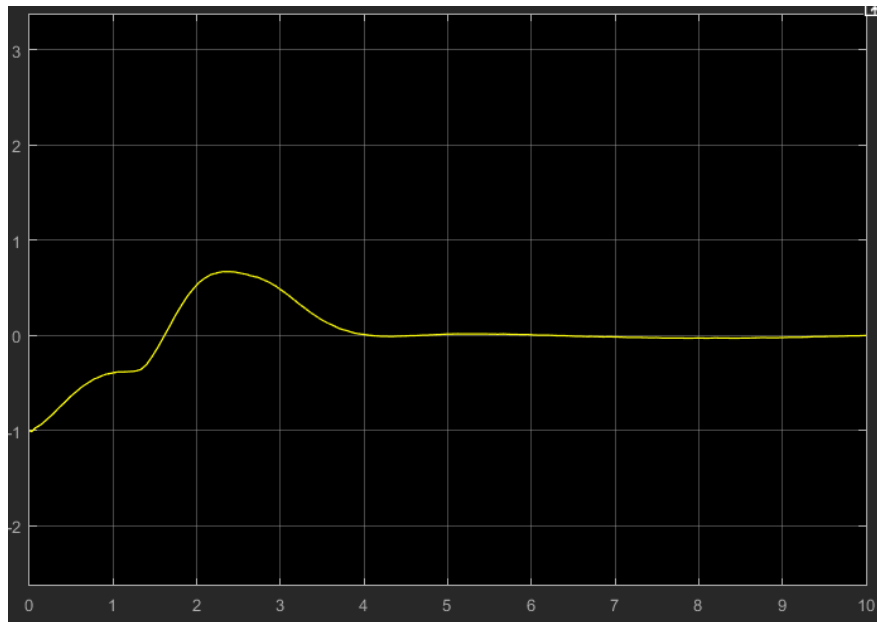
Esquema de simulación



Respuesta del sistema sin oscilaciones:



Respuesta del sistema con oscilaciones



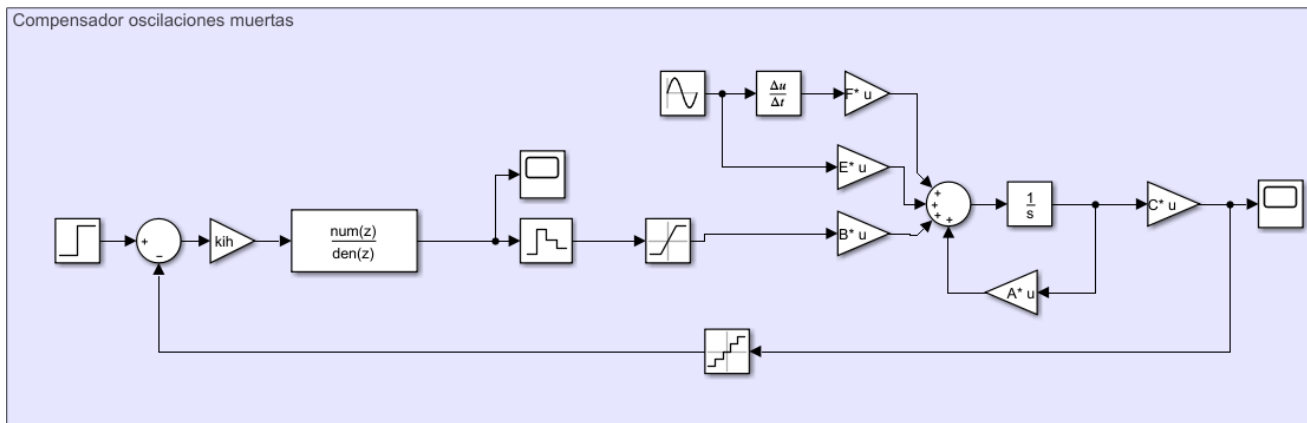
Ecuación en diferencias Tiempo de respuesta mínimo

$$u(k) = 1.325 \cdot 10^3 \cdot e(k) - 2.335 \cdot 10^3 \cdot e(k-1) + 1.049 \cdot 10^3 \cdot e(k-2) - 0.9912 \cdot u(k-1)$$

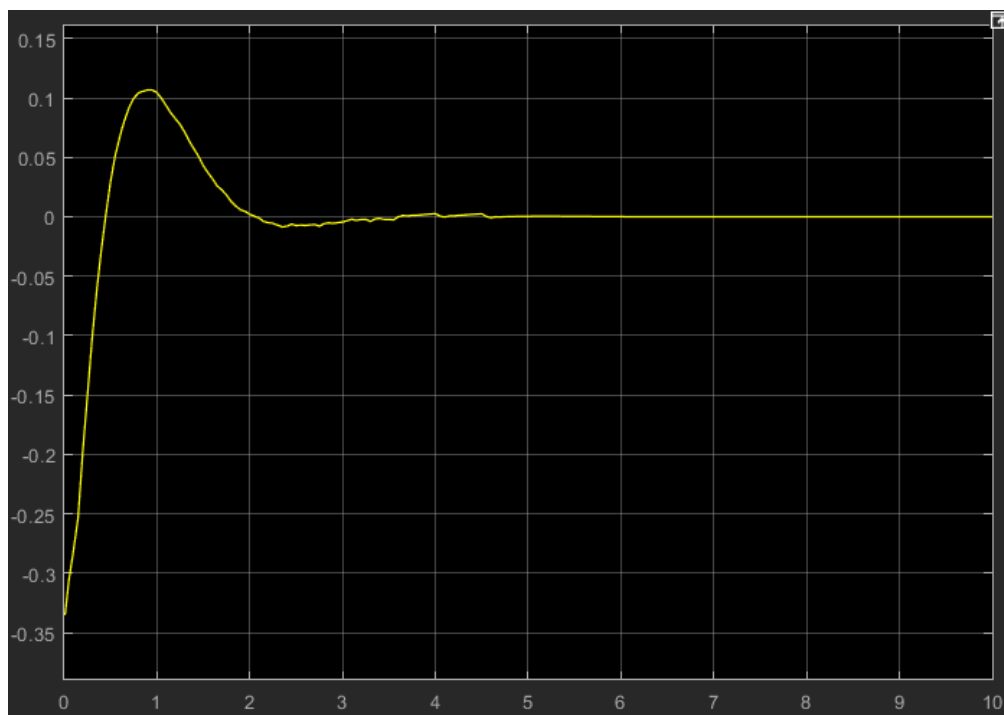
COMPENSADOR POR OSCILACIONES MUERTAS

```
% syms z m1 a1
% tiempo=0.5/10
% sysd=c2d(sys,tiempo,'zoh')
% sysd2=zpk(sysd)
% [cero polo ganancia]=zpkdata(sysd);
% ceros=cell2mat(cero)
% polos=cell2mat(polo)
% mz=collect(expand((1-ceros*z)*(m1*z)))
% c1=coeffs(mz,z)
% mz2=-collect(expand((1-z)*(1+a1*z)-1))
% c2=coeffs(mz2,z)
% [A1 M1]=solve(c1==c2,[a1 m1]);
% vpa(A1,4)
% vpa(M1,4)
% Mzr=(1-ceros*z^-1)*M1*z^-1
%
% ncont=double(conv([1 -polos(2)],[1 -polos(1)]))%Numerador
% %ncr=coeffs(ncont,z);
% dcont=conv([1 -1],[1 double(A1)])%Denominador
% kih=double(M1/ganancia) %Ganancia
```

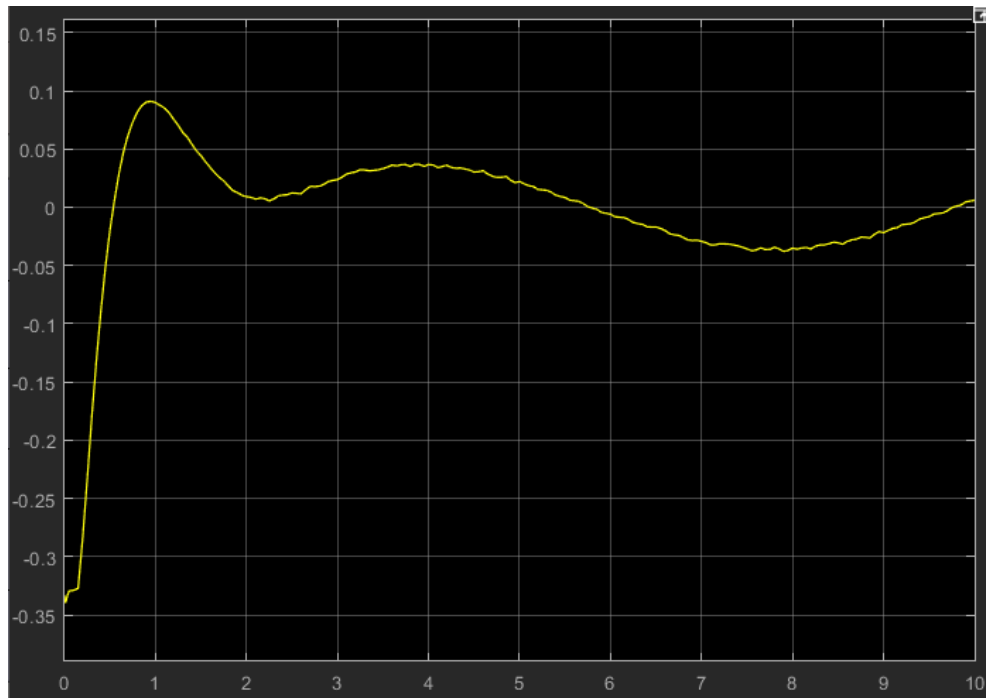
Esquema de simulación



Respuesta sin perturbación



Respuesta con perturbación



Ecuacion en diferencias Oscilaciones Muertas

$$u(k) = 1.3046 \cdot 10^3 \cdot e(k) - 2.3889 \cdot 10^3 \cdot e(k-1) + 1.1043 \cdot 10^3 \cdot e(k-2) + 0.5139 \cdot u(k-1) + 0.4861 \cdot u(k-2)$$

COMPENSADOR POR FRECUENCIA

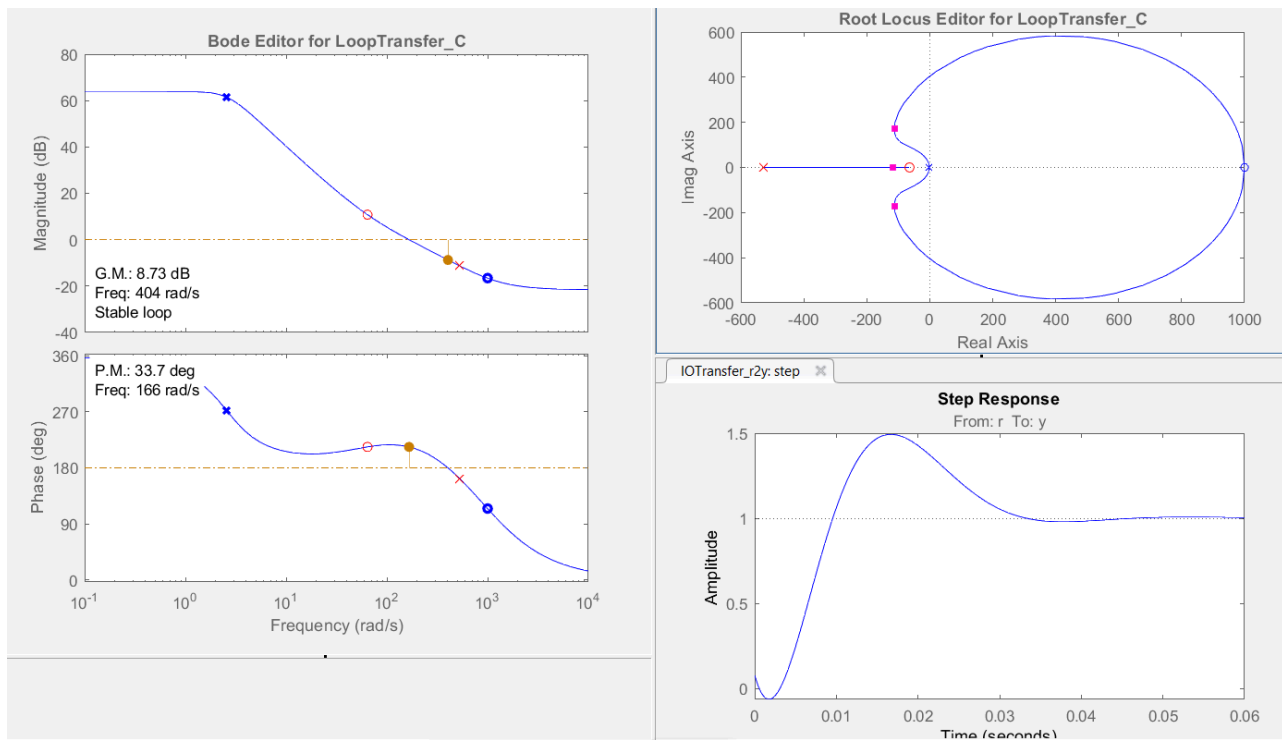
```
% ts=0.08;
% T=ts/40;
% syms z W
% sysd=c2d(sys,T,'zoh')
%
% [cero polo ganacia ]=zpkdata(sysd)
%
% ceros=cell2mat(cero)
% polos=cell2mat(polo)
%
% %PLANO    BILINEAL
% z=(1+(T*W/2))/(1-(T*W/2))
%
% Gw=vpa((ganacia)/((z-polos(1))*(z-polos(2))),4)
% [v r]=numden(vpa(simplify(Gw),2))
% v2=fliplr(coeffs(r,W))
% v1=vpa(v/v2(1),2)
%
% r1=vpa(r/v2(1),2)
% r3=coeffs(r1,W);
%
% r4=coeffs(v1,W)
%
% numgw=double(fliplr(r4))
% dengw=double(fliplr(r3))
%
```

```

%
% sysgw=tf(numgw,dengw)
% %sisotool(sysgw)
%
% syms Z
% %Sustituir Z por esto ((2*(Z-1))/(T*(Z+1)))
% nCt=5.0034e5*(Z+64.22)
% TNCT=subs(nCt,Z,(2/T)*((Z-1)/(Z+1)))
% dCt=(Z+528.1)
% TDCT=subs(dCt,Z,((2/T)*((Z-1)/(Z+1))))
%
% ReTY=simplify(TNCT/TDCT)
%
%
% [nty dty]=numden(ReTY)
% cnty=double(fliplr(coeffs(nty,Z)))
% cdy=double(fliplr(coeffs(dty,Z)))

```

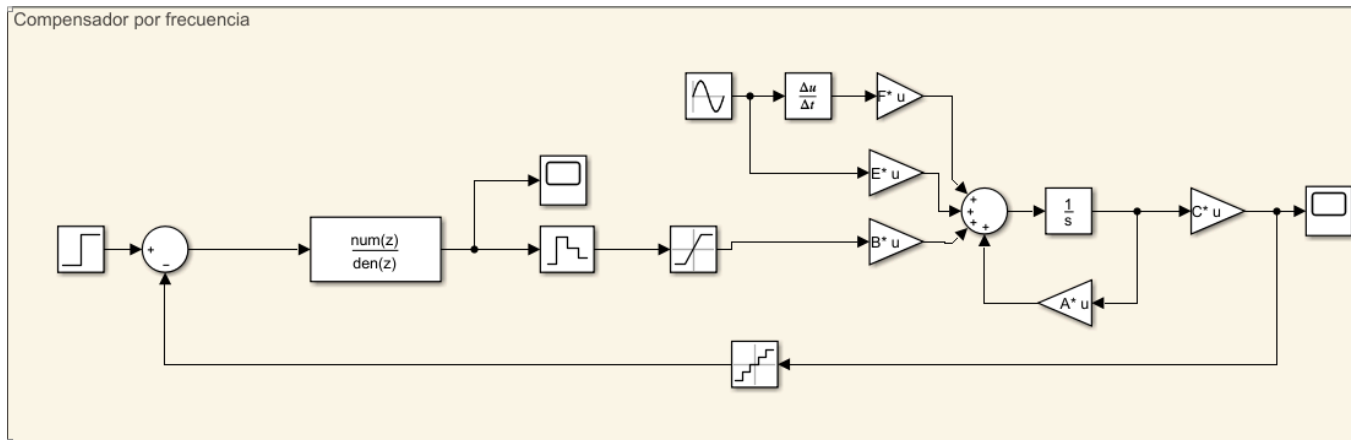
Diagrama en sisotool



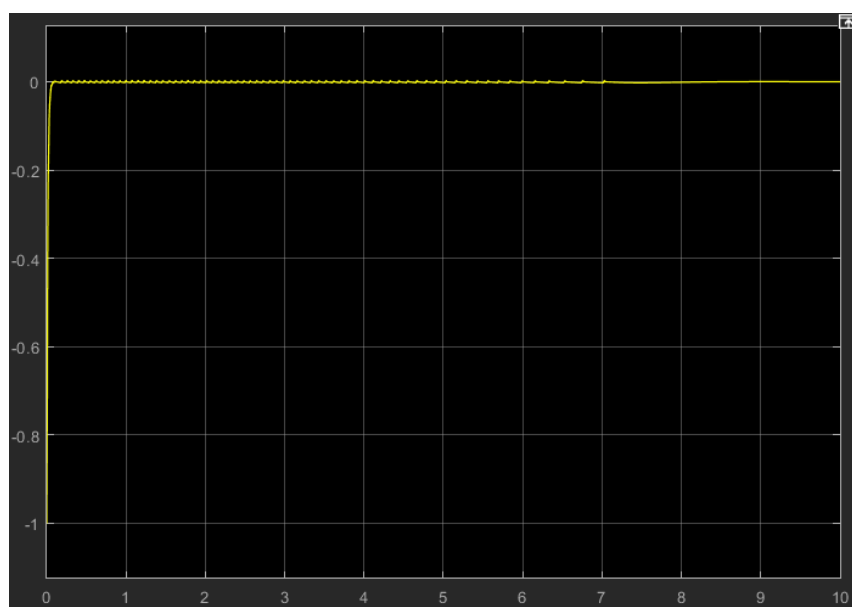
Señal de control

$$\frac{5.0034e05 (s+64.22)}{(s+528.1)}$$

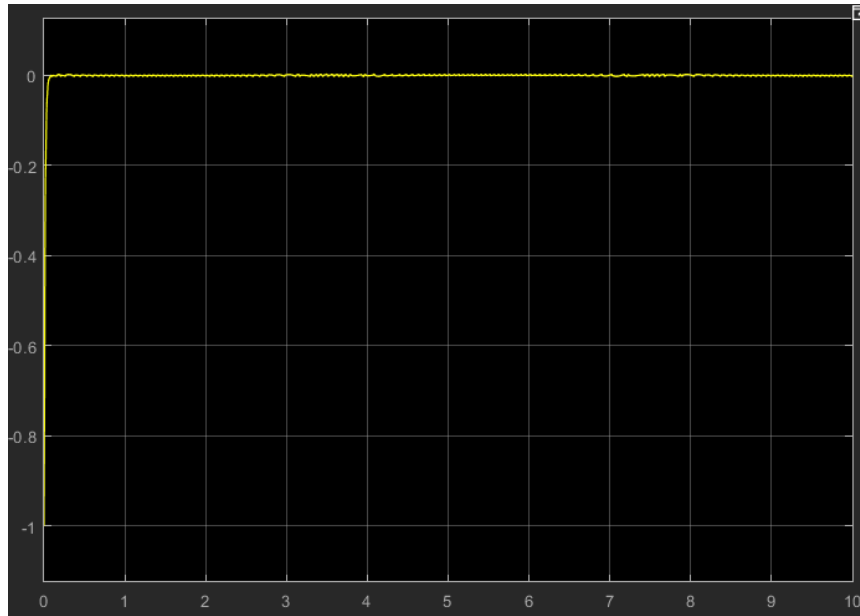
Esquema de simulación



Respuesta sin perturbación



Respuesta con perturbación



Ecuacion en diferencia Frecuencia

$$u(k) = \frac{5.3242 * 10^9 * e(k) - 4.6821 * 10^9 * e(k-1) + 4719 * u(k-1)}{15281}$$

SERVOSISTEMA

```
syms s

zd = 0.7;
tsd = 0.08;
Ts = tsd/10;

wn = 4/(zd*tsd);

T = 10;

num = [1/3];
den = [1 10/3 20/3];
[A, B, C, D] = tf2ss(num, den);
Tf = tf(num, den);
[AL,BL] = c2d(A, B, Ts);

Ss = -(zd*wn)+(1j*wn*sqrt(1-(zd^2)));
PD = exp(Ts*Ss);

[m,n] = size(AL);

ce = zeros(n, 1);

cero = zeros(n, 2);
ceros = zeros(1, n);
cer = zeros(1, 2);
cer(1, 2) = 1;
cerr = zeros(1, 2);
```

```
S = [BL AL*BL]
```

```
S = 2×2
    0.0079    0.0077
    0.0000    0.0001
```

```
det_S = det(S)
```

```
det_S = 4.9850e-07
```

```
%Servosistema entrada escalón
```

```
AE = [[AL; C*AL] [ce;1]]
```

```
AE = 3×3
    0.9735   -0.0526         0
    0.0079    0.9998         0
    0.0026    0.3333    1.0000
```

```
BE = [BL; C*BL]
```

```
BE = 3×1
    0.0079
    0.0000
    0.0000
```

```
[mE,nE] = size(AE);
```

```
nE=3;
```

```
convol = conv([1 -real(PD) -imag(PD)*1j], conv([1 -real(PD) imag(PD)*1j], [1 exp(-Ts*T*zd*wn)]
```

```
convol = 1×6 complex
    1.0000 + 0.0000i   -1.2122 + 0.0000i    0.3560 + 0.0000i    0.0069 - 0.0000i ...
```

```
[t_c_f, t_c_c] = size(convol)
```

```
t_c_f = 1
t_c_c = 6
```

```
for NBN = nE+2 : t_c_c
```

```
    convol(nE+1) = convol(nE+1)+convol(NBN)
```

```
end
```

```
convol = 1×6 complex
    1.0000 + 0.0000i   -1.2122 + 0.0000i    0.3560 + 0.0000i    0.0777 - 0.0000i ...
```

```
convol = 1×6 complex
    1.0000 + 0.0000i   -1.2122 + 0.0000i    0.3560 + 0.0000i    0.0790 - 0.0000i ...
```

```
C_PDE = convol(1:nE+1)
```

```
C_PDE = 1×4 complex
    1.0000 + 0.0000i   -1.2122 + 0.0000i    0.3560 + 0.0000i    0.0790 - 0.0000i
```

```
PDE = roots(C_PDE)
```

```
PDE = 3×1 complex
```

```

0.6779 + 0.3018i
0.6779 - 0.3018i
-0.1435 + 0.0000i

```

```

KKE = acker(AE, BE, PDE);
e_ig = eig(AE-BE*KKE)

```

```

e_ig = 3x1 complex
-0.1435 + 0.0000i
0.6779 + 0.3018i
0.6779 - 0.3018i

```

```

Kiservo = KKE(end)

```

```

Kiservo = 1.0583e+04

```

```

KE = KKE(1:end-1)

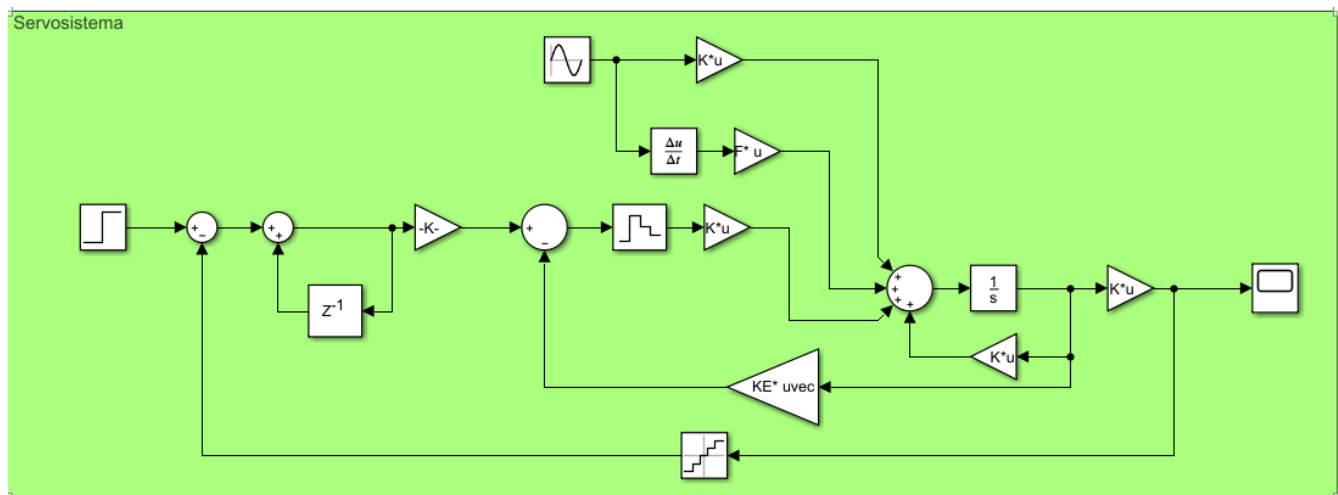
```

```

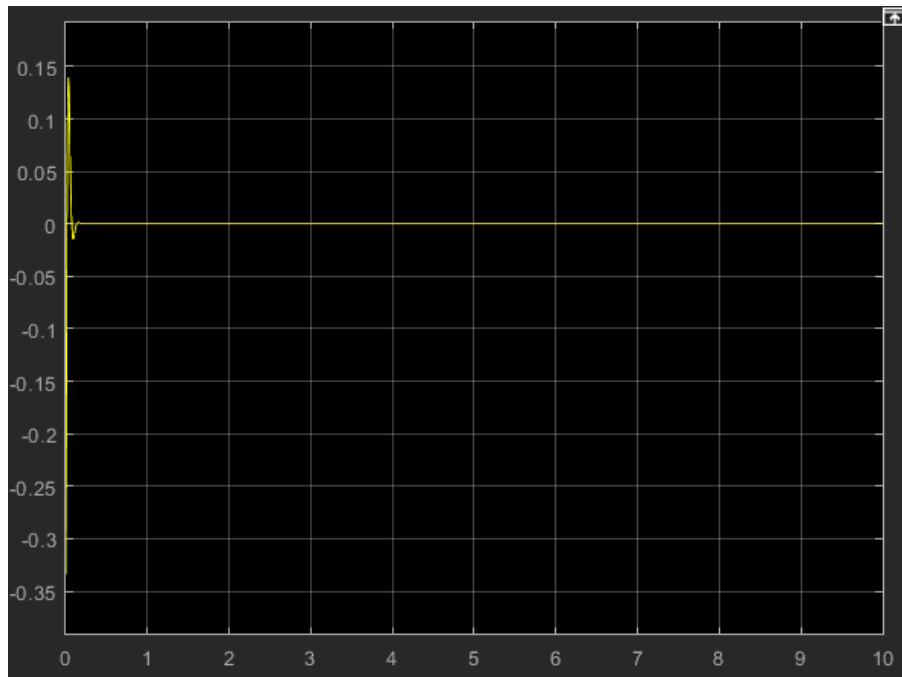
KE = 1x2
103 x
0.1710    9.4449

```

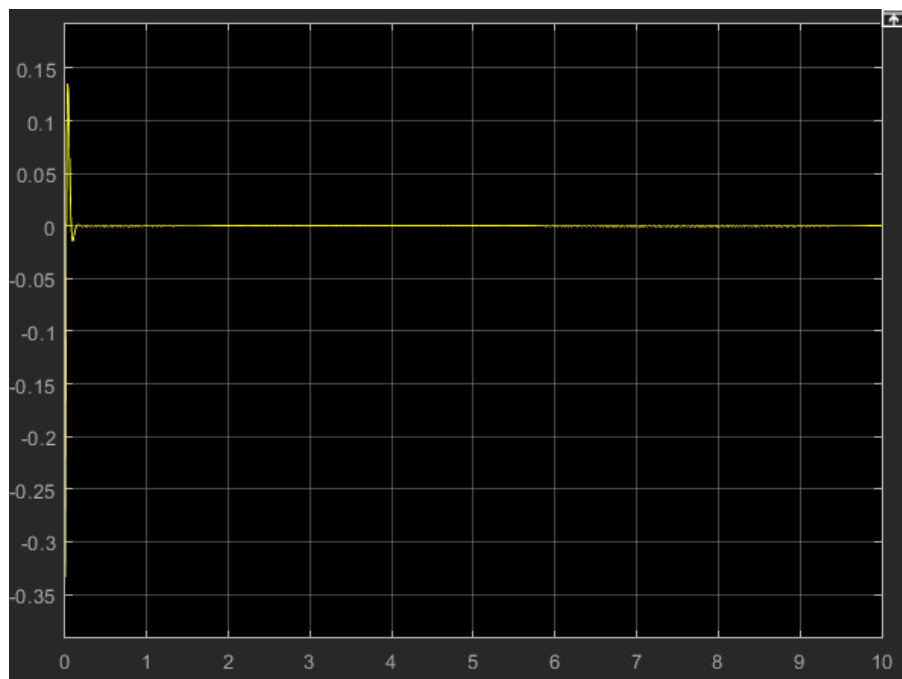
Esquema de simulación



Respuesta sin perturbación



Respuesta con perturbación



Ecuación en diferencias Servosistema

$$u(k) = -171 * x(k) - 9444.9 * v(k)$$

ANÁLISIS DE SISTEMAS

El sistema con la mejor respuesta ante perturbaciones y en el menor tiempo sin ser propenso a oscilar en su respuesta fue el compensador por frecuencias.

Los tiempos de muestreo en los compensadores se dividieron básicamente en 4 tipos, siendo t_{sd} el tiempo de establecimiento, el cual varió en cada uno de los compensadores con el fin de mitigar en su mayoría los efectos provocados por las oscilaciones:

- Para PID continuo- discreto y compensador por frecuencias, el tiempo de muestro fue $t_{sd}/40$
- Para PID discreto discreto, compensador LGR, servosistema, compensador por nulación de planta y tiempo mínimo de respuesta fue $t_{sd}/10$.
- Para compensador por oscilaciones muertas se tomó $t_{sd}/2$, ya que la planta era de orden 2