CONTROL LINEAL

Función de transferencia a espacios de estado.

A mano.

$$G(s) = \frac{Num}{Den} = \frac{a}{s^3 + bs^2 + cs + d}$$

$$\frac{Y(s)}{U(s)} = \frac{Num}{Den} = \frac{a}{s^3 + bs^2 + cs + d}$$

$$y(s) * (s^3 + bs^2 + cs + d) = U(s) * (a)$$

$$\dddot{y} + b\ddot{y} + c\dot{y} + dy = au(t)$$

$$\dddot{y} = au(t) - b\ddot{y} - c\dot{y} - dy$$

$$\dddot{y} = au(t) - bx_3 - cx_2 - dx_1$$

$x_1 = y$

$x_2 = \dot{x}_1 = \dot{y}$

$x_3 = \dot{x}_2 = \ddot{y}$

$\dot{x}_3 = \dddot{y}$

Matrices de estado.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -d & -c & -b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} u(t)$$

Matriz de salida.

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [0]u(t)$$

MATLAB

```
%Funcion de transferencia, numerador = b,
denominador =a.
b = [2 5 6];
a = [1 1 2];
%Matrices de estados
[A,B,C,D] = tf2ss(b,a)
sys = ss(A,B,C,D);
%Grafica espacios de estado.
A=[-1 -1;6.5 0];
B=[1 1;1 0];
C=[1 0;0 1];
D=[0 0;0 0];
step(A,B,C,D)
```

Función de transferencia a espacios de estado.

$$G(s) = \frac{b_3s^3 + b_2s^2 + b_1s + b_0}{s^4 + a_3\, s^3 + a_2s^2 + a_1s + a_0}$$

$$G(s) = \frac{b_3s^3 + b_2s^2 + b_1s + b_0}{s^4 + a_3\, s^3 + a_2s^2 + a_1s + a_0} * \frac{z(s)}{z(s)}$$

$$y(s) = (b_3s^3 + b_2s^2 + b_1s + b_0) * z(s)$$

$$u(s) = (s^4 + a_3\, s^3 + a_2s^2 + a_1s + a_0) * z(s)$$

$$y = b_3\dddot{z} + b_2\ddot{z} + b_1\dot{z} + b_0z$$

$$u = \ddddot{z} + a_3\dddot{z} + a_2\ddot{z} + a_1\dot{z} + a_0z$$

$$\ddddot{z} = -a_0x_1 - a_1x_2 - a_2x_3 - a_3x_4 + u$$

$x_1 = z \rightarrow \dot{x}_1 = \dot{z} = x_2$
$x_2 = \dot{z} \rightarrow \dot{x}_2 = \ddot{z} = x_3$
$x_3 = \ddot{z} \rightarrow \dot{x}_3 = \dddot{z} = x_4$

$$x_4 = \dddot{z} \rightarrow \dot{x}_1 = \ddddot{z}$$
$$= -a_0x_1 - a_1x_2 - a_2x_3 - a_3x_4 + u$$

$$y = b_3x_4 + b_2x_3 + b_1x_2 + b_0x_1$$

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = cx + bu$$

$$y = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + 0u$$

Espacios de estado a función de transferencia.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

$$y = [C_1 \quad C_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [D]u$$

$$G(s) = C(sI - A)^{-1}B + D$$

1. $(SI - A)$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad SI = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

$$SI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} = \begin{bmatrix} A_1 - s & A_2 - 0 \\ A_3 - 0 & A_4 - s \end{bmatrix}$$

2. $(SI - A)^{-1} = \frac{adj(SI-A)}{\det(SI-A)}$

$$\det(SI - A) = \begin{bmatrix} A_1 - s & A_2 \\ A_3 & A_4 - s \end{bmatrix}$$

$$\det(SI - A) = \big((A_1 - s)(A_4 - s)\big) - \big((A_2)(A_3)\big)$$

$$adj(SI - A) = \big(cof(SI - A)\big)^T$$

$cof(SI - A) = 1. \begin{bmatrix} A_1 - s & A_2 \\ A_3 & A_4 - s \end{bmatrix}$ 2. $\begin{bmatrix} A_1 - s & A_2 \\ A_3 & A_4 - s \end{bmatrix}$ 3. $\begin{bmatrix} A_1 - s & A_2 \\ A_3 & A_4 - s \end{bmatrix}$ 4. $\begin{bmatrix} A_1 - s & A_2 \\ A_3 & A_4 - s \end{bmatrix}$

$$\begin{bmatrix} + & - \\ - & + \end{bmatrix} Positivo\ par\ (+), negativo\ inpar\ (-)$$

$$\big(cof(SI - A)\big)^T = \begin{bmatrix} A_4 - s & A_2 \\ A_3 & A_1 - s \end{bmatrix}$$

$$adj(SI - A) = \big(cof(SI - A)\big)^T = \begin{bmatrix} A_4 - s & A_2 \\ A_3 & A_1 - s \end{bmatrix}$$

$(SI - A)^{-1} =$

$$\frac{1}{\big((A_1 - s)(A_4 - s)\big) - \big((A_2)(A_3)\big)} * \begin{bmatrix} A_4 - s & A_2 \\ A_3 & A_1 - s \end{bmatrix}$$

$G(S) =$

$$[D_1 \quad D_2] * \frac{1}{\big((A_1 - s)(A_4 - s)\big) - \big((A_2)(A_3)\big)} * \begin{bmatrix} A_4 - s & A_2 \\ A_3 & A_1 - s \end{bmatrix} * \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

MATLAB

```
%Espacios de estado a función de
transferencia.
clc;
A = [-1 0.5; 0 -2]
B = [1; 0.5]
C = [1 0]
D = 0
[num,den] = ss2tf(A,B,C,D)
tf_sys = tf(num,den)
```

Controlabilidad.

$$C = [B \quad AB \quad AB^{n-1}] \quad C = [B \quad AB \quad A^2B]$$

Observabilidad.

$$O = \begin{bmatrix} C \\ CA \\ CA^{n-1} \end{bmatrix} \quad O = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix}$$

MATLAB

```
%Observabilidad
Ob = obsv(sys)
rank(Ob)
%Controlabilidad
Co = ctrb(sys)
rank(Co)
co = ctrb(A,B);
Unco = length(A) - rank(co);
if Unco == 0
    men = 'Es controlable';
else
    men = 'No es controlable';
end
disp(men)

Ob = obsv(A,C);
Unobsv = length(A)-rank(Ob);
if Unobsv == 0
    men2 = 'Es observable';
else
    men2 = 'No es observable';
end
disp(men2)
```

Routh – Hurwitz.

$$T(S) = \frac{N(s)}{a_n S^n + a_{n-1} S^{n-1} + \cdots + a_1 s + a_0} = \frac{N(s)}{D(s)}$$

T(s) será estable si:

1. D(s) no tiene raíces en el semi plano derecho.
2. D(s) no tiene raíces repetidas sobre el eje jw.
3. T(s) será asintóticamente estable si todas las raíces de D(s) están en el semiplano izquierdo del plano complejo.

$$D(s) = a_n S^n + a_{n-1} S^{n-1} + a_{n-2} s^{n-2} + a_{n-3} s^{n-3} + a_{n-4} s^{n-4} + \cdots + a_1 s + a_0$$

s^n	a_n	a_{n-2}	a_{n-4}	...
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	...
s^{n-2}	b_1	b_2	b_3	...
s^{n-3}	c_1	c_2	c_3	...
.	.	.		
.	.	.		
.	.	.		
s^2	e_1	e_2		
s^1	f_1			
s^0	g_1			

$$b_1 = \frac{(a_{n-1})(a_{n-2})-(a_n)(a_{n-3})}{a_{n-1}} \quad b_2 = \frac{(a_{n-1})(a_{n-4})-(a_n)(a_{n-5})}{a_{n-1}}$$

$$C_1 = \frac{(b_1)(a_{n-3})-(a_{n-1})(b_2)}{b_1} \quad C_2 = \frac{(b_1)(a_{n-5})-(a_{n-1})(b_3)}{b_1}$$

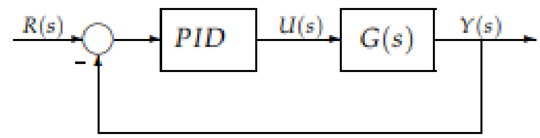
El número de cambios de signo en la primera columna corresponde al número de polos inestables.

MATLAB

```
%Criterio de estabilidad Routh Hurwitz.
clc;
clear
Num = [10 2 1];
Den = [6 8 12];
roots(Den)
G = tf(Num,Den)
%Grafica de raices, polos y ceros
pzmap(G)
%Grafica de impulso
impulse(G)
```

PID

El control PID utiliza tres componentes principales:



P: acción de control proporcional, da una salida del controlador que es proporcional al error, es decir: u(t) = KP.e(t),que descripta desde su funcion transferencia queda:

$$C_p(s) = K_p$$

I: acción de control integral: da una salida del controlador que es proporcional al error acumulado, lo que implica que es un modo de controlar lento.

$$u(t) = K_i \int_0^t e(\tau) d\tau \qquad C_i(s) = \frac{K_i}{s}$$

PI: acción de control proporcional-integral, se define mediante.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau$$

Donde T_i se denomina tiempo integral y es quien ajusta la acción integral. La función de transferencia resulta:

Con un control proporcional, es necesario que exista error para tener una acción de control distinta de cero. Con acción integral, un error peque no positivo siempre nos dará una acción de control creciente, y si fuera negativo la señal de control será decreciente. Este razonamiento sencillo nos muestra que el error en régimen permanente será siempre cero. Muchos controladores industriales tienen solo acción PI. Se puede demostrar que un control PI es adecuado para todos los procesos donde la dinámica es esencialmente de primer orden. Lo que puede demostrarse en forma sencilla, por ejemplo, mediante un ensayo al escalón.

PD: acción de control proporcional-derivativa, se define mediante:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

donde T_d es una constante de denominada tiempo derivativo. Esta acción tiene carácter de previsión, lo que hace mas rápida la acción de control, aunque tiene la desventaja importante que amplifica las señales de ruido y puede provocar saturación en el actuador. La acción de control derivativa nunca se utiliza por sí sola, debido a que solo es eficaz durante periodos transitorios. La función transferencia de un controlador PD resulta:

$$C_{PD}(s) = K_p + sK_p T_d$$

PID: acción de control proporcional-integral-derivativa, esta acción combinada reúne las ventajas de cada una de las tres acciones de control individuales. La ecuación de un controlador con esta acción combinada se obtiene mediante:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt}$$

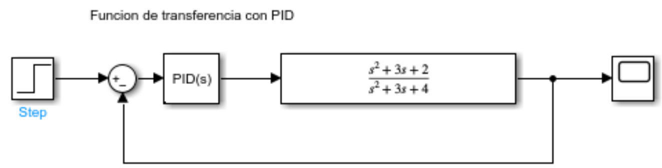
Y su función transferencia resulta:

$$C_{PID}(s) = K_p\left(1 + \frac{1}{T_i S} + T_d s\right)$$

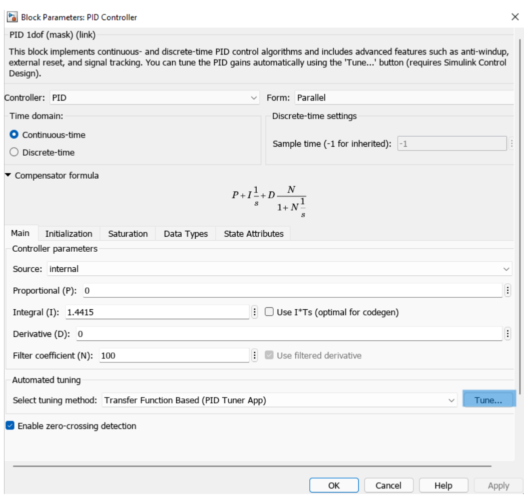
$$K_i = \frac{K_p}{\tau_i} \qquad K_d = K_p \tau_d$$

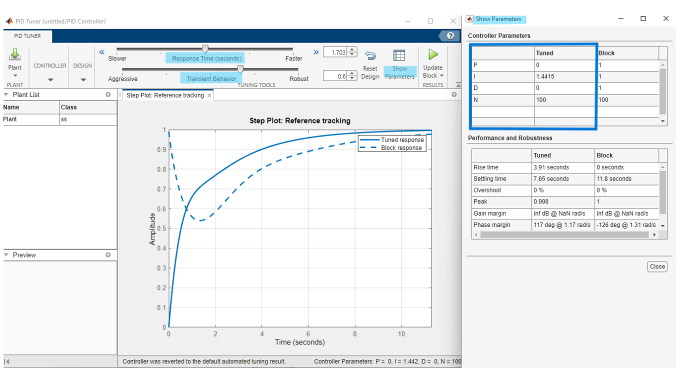
MATLAB

```
%PID----TUNER.
simulink
```



*click en el bloque de PID



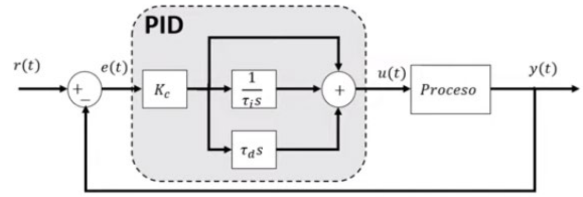


Reemplaza los valores para P, I, D.

Ziegler – Nichols

Lazo abierto.

Control PID.



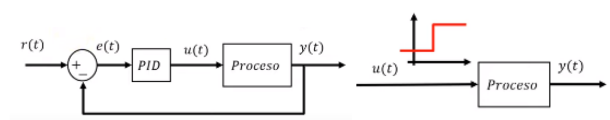
Función de transferencia del PID.

$$G_C(s) = K_P\left(1 + \frac{1}{\tau_i s} + \tau_d s\right) \rightarrow G_C(s) = K_p + \frac{K_i}{s} + K_d s$$

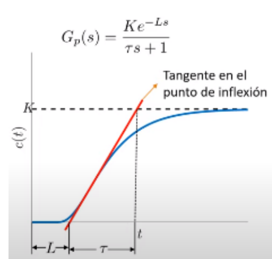
Si el controlador recibe la señal de error e(t) como entrada, la salida del controlador u(t) es dada por:

$$u(t) = K_p\left[e(t) + \frac{1}{\tau_i}\int_0^t e(t)dt + \tau_d \frac{de(t)}{dt}\right]$$

METODO 1. Se realiza con el sistema en lazo abierto.



$$Primer\ orden\ con\ retardo.\ G_p(s) = \frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{\tau s + 1}$$



$$G_c(s) = K_p\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)$$

Control	K_p	τ_i	τ_d
P	$\frac{\tau}{KL}$	∞	0
PI	$0{,}9\frac{\tau}{KL}$	$\frac{L}{0{,}3}$	0
PID	$1{,}2\frac{\tau}{KL}$	2L	0,5L

$$\text{Factor de incontrolabilidad. } 0.1 \leq \frac{L}{\tau} \leq 0.3$$

$$\text{Para PID } G(s) = 0.6\tau * \frac{\left(s+\frac{1}{L}\right)^2}{s}$$

Para controladores analogicos y no digitales. Sii el periodo de muestreo T_s del sistema es grande, las formulas anteriormente vistas pueden generar un declino mayor al 25% tendiendo para la inestabilidad. Una opcion es aumentar el retardo a un valor igual a la mitad del periodo de muestreo.

$$L' = L + \frac{T_s}{2}$$

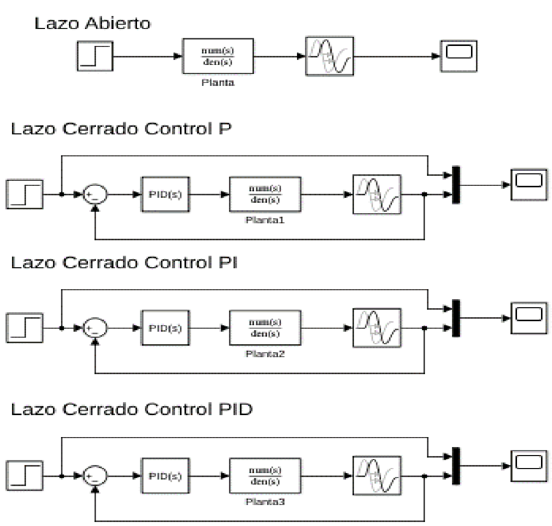
Antes de utilizar los valores de la tabla.

Muchas veces para mejorar la respuesta del sistema, deberemos disminuir la ganancia del controlador. Una práctica común es dividir la ganancia del controlador a la mitad para obtener una respuesta más suave.

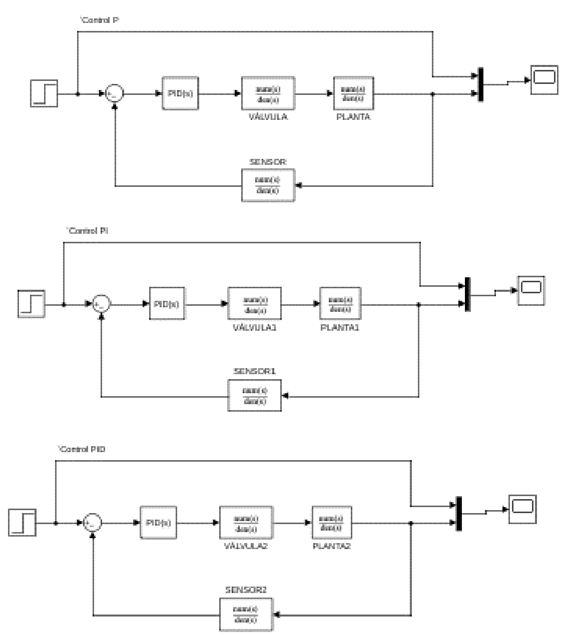
$$\frac{K_p}{2}$$

$$Si\ \left(\frac{L}{\tau}\right)\ es\ grande, es\ dificil\ de\ controlar.$$

MATLAB

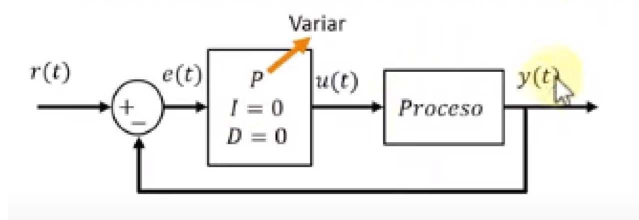


MATLAB



Ziegler – Nichols

Metodo 2. Se realiza en lazo cerrado.



Ganancia limite, ganancia ultima o ganancia critica.

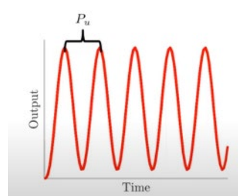
$$K_u$$

Y a partir del grafico Podemos calcular el period critico.

$$P_u$$

O apartir de la frecuencia.

$$P_u = \frac{2\pi}{W_u}$$



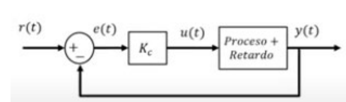
$$G_c(s) = K_p\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)$$

Control	K_p	τ_i	τ_d
P	$0,5K_u$	∞	0
PI	$0,45K_u$	$\frac{1}{1,2}P_u$	0
PID	$0,6K_u$	$0,5P_u$	$0,125P_u$

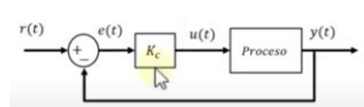
Para PID $$G_c(s) = 0.075K_uP_u\frac{\left(s+\frac{4}{P_u}\right)^2}{s}$$

Cuando el Sistema llega a la ganancia critica o ganancia limite, indica que esta el borde de la inestabilidad. Gráficamente en el diagrama de polos y ceros, nos indica que los polos del Sistema de lazo cerrado se encuentran sobre el eje imaginario y un pequeño incremento en la ganancia provocara la inestabilidad.

Para que el sistema oscile, deberá tener un orden igual o superior a 3 (tener polos complejos conjugados, por ejemplo), o por lo menos deberá tener un retardo de tiempo, que hará que los polos crucen por el eje imaginario.

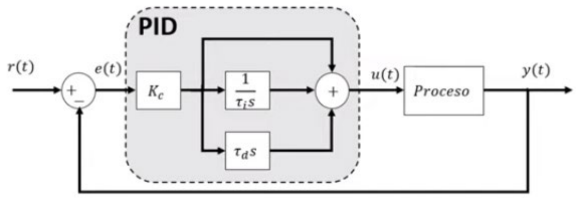


$$e^{-Ls} = \frac{1-\frac{L}{2s}}{1+\frac{L}{2s}}$$



$$G_{LC} = \frac{1}{(\tau_1 s+1)(\tau_2 s+1)}$$

PID Por asignación de polos



Funcion de transferencia de la planta.

$$G(s) = \frac{k}{s^2 + as + b} = \frac{A}{B}$$

Control PID.

$$C(s) = \frac{k_c \tau_d s^2 + k_c s + \frac{k_c}{\tau_i}}{s} = \frac{d_2 s^2 + d_1 s + d_0}{s} = \frac{D}{E}$$

Funcion de tranferencia en lazo cerrado.

$$H(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{\frac{D}{E}\frac{A}{B}}{1 + \frac{D}{E}\frac{A}{B}} = \frac{DA}{EB + DA}$$

$$H(s) = \frac{k(d_2 s^2 + d_1 s + d_0)}{s^3 + (a + kd_2)s^2 + (b + kd_1)s + kd_0}$$

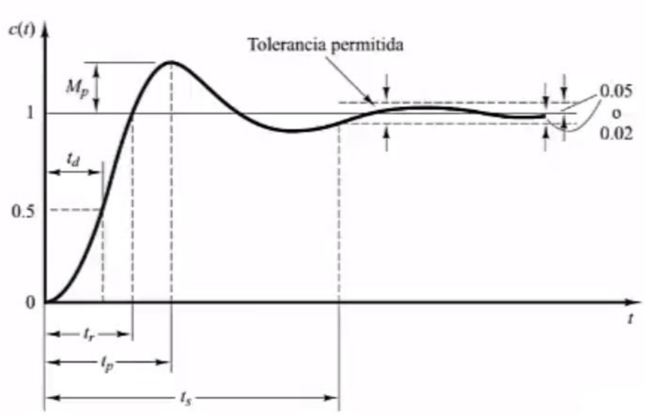
Parámetros de diseño.

Tiempo de establecimiento.

$$2\% \rightarrow t_s = \frac{4}{\zeta w_n} \qquad 5\% \rightarrow t_s = \frac{3}{\zeta w_n}$$

Pico maxico.

$$M_p = 100 e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$



$$P_d(s) = (s^2 + h_1 s + h_2)(s + p_1)$$

$$s^3 + (a + kd_2)s^2 + (b + kd_1)s + kd_0 = s^3 + a_1 s^2 + a_2 s + s_3$$

(1) $a + kd_2 = a_1$

(2) $b + kd_1 = a_2$

(3) $kd_0 = a_3$

(4) $a + k\tau_d k_c = a_1$

(5) $b + kk_c = a_2$

(6) $k\frac{k_c}{\tau_i} = a_1$

(7) $k_c = \frac{a_2 - b}{k}$

(8) $\tau_i = \frac{kk_c}{a_3}$

(9) $\tau_d = \frac{a_1 - a}{kk_c}$

MATLAB

```
clc
clear all
close all
%% Diseño del Controlador PID
%Funcion de transferencia del proceso
% C=tf(Kc*[Ti 1],[Ti 0]);
P=tf(20,[1 8 17]);
%Obtiene el numerador y denominador de la FT
[n,d]=tfdata(P,'v');
%Nombro los terminos de la FT
k=n(3);
a=d(2);
b=d(3);
% Especificaciones de Diseño
Mp=50; %Maximo Pico
ep=sqrt(((log(Mp/100))^2)/(pi^2+((log(Mp/100))^2)));
%Fator de amortecimento
tau=1/(abs(max(roots(d)))); %Toma o valor do polo
dominante
Tss=(tau*4)*0.75;
Wn=3/(ep*Tss); %Frequência Natural
Sd=[-ep*Wn+1i*Wn*sqrt(1-ep^2), -ep*Wn-1i*Wn*sqrt(1-
ep^2)]; %Alocação de Polos
p3=real(Sd(1))*10; %Polo nao dominante 20 veces longe do
dominante
Sd1=[Sd p3];
Pds=poly(Sd1);
alpha=0.01;
%Calculo del Controlador
Kc=(Pds(3)-b)/k;
ti=(k*Kc)/Pds(4);
td=(Pds(2)-a)/(k*Kc);
%Parametros del PID con Filtro en el termino Derivativo
d2=alpha*Kc*ti*td+Kc*ti*td;
d1=Kc*ti+alpha*Kc*td;
d0=Kc;
```

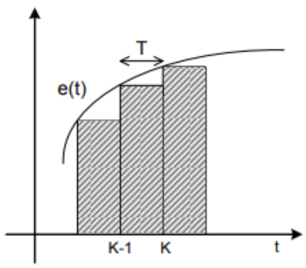
DISCRETIZACION.

Discretización de controladores continuos.

PID.

$$u(t) = K_p\left[e(t) + \frac{1}{\tau_i}\int_0^t e(t)dt + \tau_d\frac{de(t)}{dt}\right]$$

Euler hacia adelante.



La derivada se aproxima por:

$$\frac{de(t)}{dt} = \frac{e_k - e_{k-1}}{T}$$

Termino integral.

$$\int_0^t e(\tau)d\tau = \sum_{i=0}^{k-1} e(i)T = \sum_{i=0}^{k-1} Te_i$$

El PID queda.

$$u_{k-1} = K_p\left[e_{k-1} + \frac{T}{T_i}\sum_{i=0}^{k-2} e_i + \frac{T_d}{T}(e_{k-1} - e_{k-2})\right]$$

$$u_k - u_{k-1} = q_0e_k + q_1e_{k-1} + q_2e_{k-2}$$

$$q_0 = k_p\left(1 + \frac{T_d}{T}\right)$$

$$q_1 = k_p\left(-1 - 2\frac{T_d}{T} + \frac{T}{T_i}\right)$$

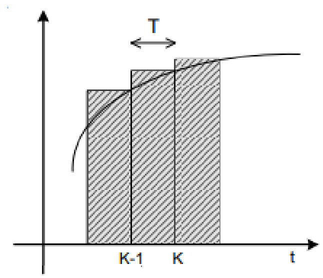
$$q_2 = k_p\left(\frac{T_d}{T}\right)$$

Aplica transformada Z

$$C(z) = \frac{U(z)}{E(z)} = \frac{q_0 + q_1z^{-1} + q_2z^{-2}}{1 - z^{-1}}$$

$$s = \frac{z-1}{T}$$

Euler hacia atrás.



Termino integral

$$\int_0^t e(\tau)d\tau = \sum_{i=1}^{k} e(i)T = \sum_{i=1}^{k} Te_i$$

$$u_k = K_p\left[e_k + \frac{T}{T_i}\sum_{i=1}^{k} e_i + \frac{T_d}{T}(e_{k-1} - e_{k-2})\right]$$

$$u_k - u_{k-1} = q_0e_k + q_1e_{k-1} + q_2e_{k-2}$$

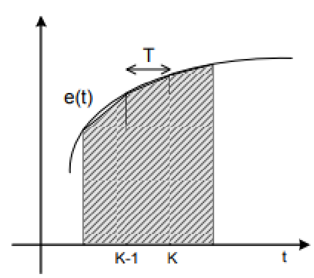
$$q_0 = k_p\left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right)$$

$$q_1 = k_p\left(-1 - 2\frac{T_d}{T}\right)$$

$$q_2 = k_p\left(\frac{T_d}{T}\right)$$

$$s = \frac{z-1}{Tz}$$

Trapezoidal.



Termino integral

$$\int_0^t e(\tau)d\tau = \sum_{i=1}^{k} e\left(i - 1 + \frac{e(i) - e(i-1)}{2}\right)T$$
$$= \sum_{i=1}^{k} T\frac{e_i + e_{i-1}}{2}$$

$$u_k - u_{k-1} = q_0e_k + q_1e_{k-1} + q_2e_{k-2}$$

$$q_0 = k_p\left(1 + \frac{T}{2T_i} + \frac{T_d}{T}\right)$$

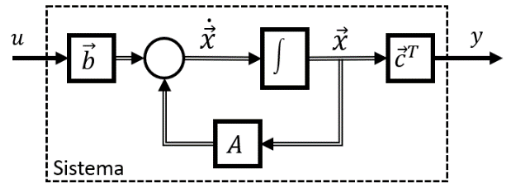
$$q_1 = k_p\left(\frac{T}{2T_i} - 1 - 2\frac{T_d}{T}\right)$$

$$q_2 = k_p\left(\frac{T_d}{T}\right)$$

MATLAB

```
clc;
clear
%% TF CONTROLADOR --- Continuo
s=tf('s')
Num = [1 3];
Den = [1 3 10 3];
Cs=tf(Num,Den)
%hold on
%step(Cs)
%impulse(Cs)
%% Controlador en Discreto_Metodos ZOH
FOH  IMPULSE  y MATCHED  con función c2d
T=0.2
Czoh=c2d(Cs,T,'zoh')      %% Técnica Zero
order hold
Cfoh=c2d(Cs,T,'foh')      %% Técnica
First order hold
Cimp=c2d(Cs,T,'impulse')  %% Técnica
invariancia al impulso
Cmat=c2d(Cs,T,'matched')  %% Técnica
mapeo de polos y zeros
figure
subplot(2,1,1)
hold on
step(Czoh)
step(Cfoh)
step(Cimp)
step(Cmat)
subplot(2,1,2)

hold on
step(Cs)
impulse(Cs)
```

Control por realimentación de estados.



$$\dot{x}(t) = A\vec{x}(t) + \vec{b}u(t)$$

$$y(t) = \vec{c}^T\vec{x}(t)$$

Las entradas del sistema deben ser computadas usando el vector de estados.

$$u(t) = x(\vec{x}(t), t)$$

$$u(t) = -k\vec{x}(t),\ \ \vec{x}\epsilon\Re^n, \qquad k\epsilon\Re^{1*n}$$

Función en espacios de estado final.

$$\dot{x}(t) = A\vec{x}(t) + \vec{b}u(t) \text{ donde: } u(t) = -k\vec{x}(t)$$

$$\dot{x}(t) = A\vec{x}(t) - \vec{b}k\vec{x}(t)$$

$$\dot{x}(t) = (A - \vec{b}k)\vec{x}(t)$$

$$\tilde{A} = (A - \vec{b}k)$$

Los autovalores de $\tilde{A}$ comienza a dominar la dinámica del nuevo sistema el cual depende de la ganancia k.

Como la ganancia k afecta al sistema.

$$G(s) = \frac{b_0 s}{s^2 + a_1 s + a_2}$$

Representando el sistema en la forma canónica controlable.

$$\dot{\vec{x}} = \begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix}\vec{x}$$

$$y = [b_0 \quad 0]\vec{x}$$

Aplicando la retroalimentación de estados.

$$\dot{x}(t) = (A - \vec{b}k)\vec{x}(t)$$

$$\dot{\vec{x}}(t) = \begin{bmatrix} -a_1 - k_1 & -a_2 - k_2 \\ 1 & 0 \end{bmatrix}\vec{x}(t)$$

De esa forma, se vuelve factible poder modificar las *n* raíces de la ecuación característica de nuestra función de transferencia. Lo más importante, es entender que para poder modificar el comportamiento dinámico del sistema es necesario que el par A, $\vec{b}$ sean totalmente controlables.

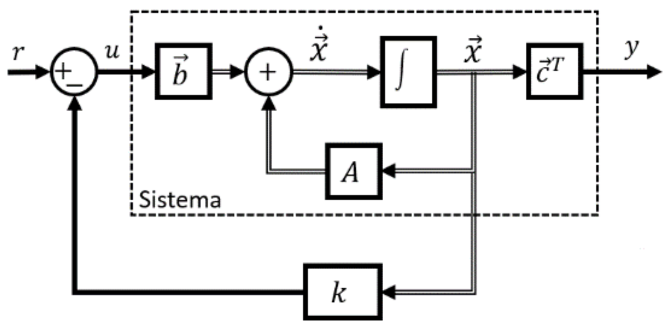
Seguimiento de referencia.

Por otro lado, si deseo seguir una referencia, basta con agregar la referencia a la ecuación de la ley de control:

$$u(t) = -k\vec{x}(t) + r(t)$$

La representación de **variables de estado** vendría dado por:

$$\dot{x}(t) = (A - \vec{b}k)\vec{x}(t) + \vec{b}r(t)$$



Representación entrada salida

$$G(s) = \vec{c}(sI - A)^{-1}\vec{b}$$

$$G_{LC}(s) = \vec{c}\left(sI - (A - \vec{b}k)\right)^{-1}\vec{b}$$

MATLAB

CONTROLADOR POR RETROALIMENTACION DE ESTADOS:

Codigo:

RETROALIMENTACION DE ESTADOS por ***regulación*** y ***servo-sistema***, en este ejercicio tendremos las variables de estados, para segundo orden.

```
clc;
clear
%Funcion de transferencia, numerador = b, denominador =a.
b = [5 1 2];
a = [1 10 25];
G = tf(b,a)
%Matrices de estados
[A,B,C,D] = tf2ss(b,a)
sys = ss(A,B,C,D);
```

Controlabilidad y observabilidad:

Aqui se comprueba si el sistemas es controlable y observable por medio de las funciones ***ctrb*** y ***obsv***.

```
co = ctrb(A,B);
Unco = length(A) - rank(co);
if Unco == 0
    men = 'Es controlable';
else
    men = 'No es controlable';
end
disp(men)
Ob = obsv(A,C);
Unobsv = length(A)-rank(Ob);
if Unobsv == 0
    men2 = 'Es observable';
else
    men2 = 'No es observable';
end
disp(men2)
```

Controlador:

establecemos los parametros que tendra nuestro controlador como el ***zita*** y el *tiempo de **establecimiento.***

Con esto encontraremos el polinomio deseado del controlador:

```
zitadesecont = 0.7;
tsdesecont = 1;
wndcont = 4/(zitadesecont*tsdesecont);
Aempa = [A,[0;0]; -C,0]
Bempa = [B;0]
pdeseadocontrolador = conv([1 2*zitadesecont*wndcont
wndcont^2],[1 (200*zitadesecont*wndcont)])
```

Funcion de polinomio de Akerman:

Hacemos que la ***Aempaquetada*** se reemplaze en cada una de las S del polinomio deseado:

```
phiAemp=(Aempa^3)+pdeseadocontrolador(2)*(Aempa^2)+pdese
adocontrolador(3)*(Aempa)+pdeseadocontrolador(4)*eye(3)
```

Calculo de las K:

Procedemos a encontrar cada una de las K las cuales seran las encargadas de controlar la planta:

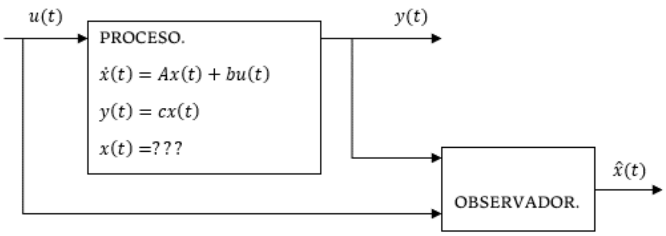
```
kemp = [0 0 1]*inv([Bempa Aempa*Bempa
(Aempa^2)*Bempa])*phiAemp
K12 = [kemp(1:2)]
ki = -kemp(3)
```

Observadores de estados.

En lazo abierto. Recordando que la ecuación de estados completo viene dado por la ecuación:

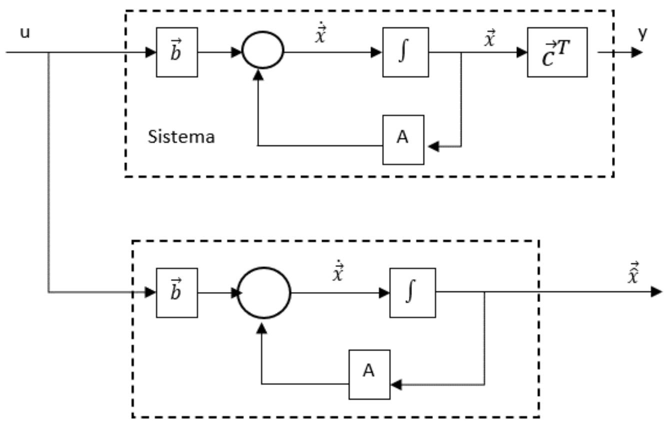
$$\dot{x}(t) = Ax(t) + bu(t)$$

$$y(t) = cx(t)$$



Objetivo. $t \rightarrow \infty \quad \hat{x}(t) \rightarrow x(t)$

$$\dot{\hat{x}}(t) = A\hat{x}(t) + bu(t)$$



Si la condición inicial del observador de estado de lazo abierto es la misma que la del sistema entonces $\hat{x}(t) = x(t), para\ t \geq 0$.

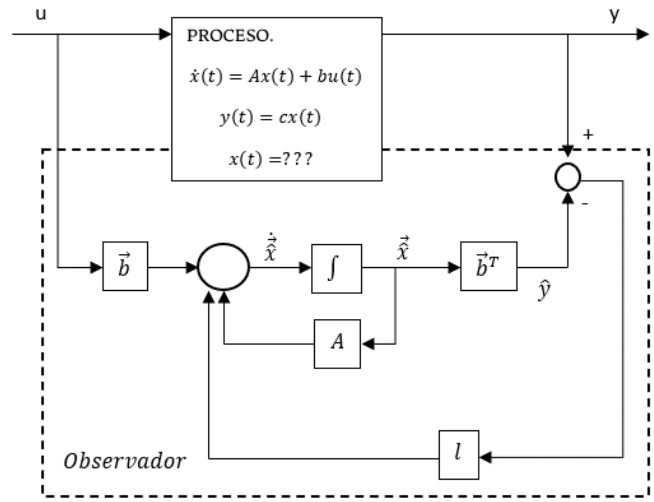
A partir de la condición inicial, podemos determinar el valor de estado en el instante $t_2 \geq t_1$ y actualizar el valor del observador.

$$\hat{x}(t_2) = x(t_2)$$

Recordando que la ecuación de estados completo viene dado por la ecuación:

$$x(t) = e^{At}x(0) + \int_0^t e^{-A(t-\tau)}Bu(\tau)d\tau$$

Observadores de estados.



$$I\big(y(t) - \hat{y}(t)\big) = I\big(y(t) - c\hat{x}(t)\big)$$

$$\dot{\hat{x}}(t) = A\hat{x}(t) + bu(t) + I\big(y(t) - c\hat{x}(t)\big) = (A - Ic)\hat{x}(t) + bu(t) + Iy(t)$$

El error de estimación.

$$e(t) = x(t) - \hat{x}(t)$$

Dinámica del error de estimación.

$$\dot{e}(t) = Ax(t) + bu(t) - (A\hat{x}(t) + bu(t) + Icx(t) - Ic\hat{x}(t))$$

$$\dot{e}(t) = (A - Ic)x(t) - (A - Ic)\hat{x}(t)$$

$$\dot{e}(t) = (A - Ic)e(t)$$

Los autovalores de (A-Ic) definen la velocidad de convergencia del observador ya que e(t)=0 y $\hat{x}(t) = x(t)$.

Si (A-Ic) es estable, entonces $e(t) \rightarrow 0\ cuando\ t \rightarrow \infty$

Sea el par (A, c). Todos los valores propios de (A-Ic) pueden ser arbitrariamente asignados a un vector apropiado I si y solo s i el par (A, c) es observable.

Procedimiento para calcular el observador de estados.

1. Diseñar una realimentación de estado para el sistema aplicando la dualidad

$$e_a(t) = (A' - c'k)e_a(t)$$

2. La ganancia del observador viene dada por (que es muy parecido a una realimentación de estados):

$$I = k'$$

Procedimiento para calcular el observador de estados.

$$g(s)=\frac{y(s)}{u(s)}=\frac{\beta_1 s^3+\beta_2 s^2+\beta_3 s+\beta_4}{s^4+\alpha_1 s^3+\alpha_2 s^2+\alpha_3 s+\alpha_4}$$

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\\ \dot{x}_4(t)\end{bmatrix}=\begin{bmatrix}-\alpha_1 & 1 & 0 & 0\\ -\alpha_2 & 0 & 1 & 0\\ -\alpha_3 & 0 & 0 & 1\\ -\alpha_4 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\\ x_4(t)\end{bmatrix}+\begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\\ \beta_4\end{bmatrix}u(t)$$

$$y(s)=[1\quad 0\quad 0\quad 0]\,x(t)$$

La cual es el sistema DUAL de la forma canónica controlable. En el observador tengo que calcular (A-lc)

$$\begin{bmatrix}-\alpha_1 & 1 & 0 & 0\\ -\alpha_2 & 0 & 1 & 0\\ -\alpha_3 & 0 & 0 & 1\\ -\alpha_4 & 0 & 0 & 0\end{bmatrix}-\begin{bmatrix}l_1\\ l_2\\ l_3\\ l_4\end{bmatrix}*[1\quad 0\quad 0\quad 0]$$

$$\begin{bmatrix}-\alpha_1 & 1 & 0 & 0\\ -\alpha_2 & 0 & 1 & 0\\ -\alpha_3 & 0 & 0 & 1\\ -\alpha_4 & 0 & 0 & 0\end{bmatrix}-\begin{bmatrix}l_1 & 1 & 0 & 0\\ l_2 & 0 & 1 & 0\\ l_3 & 0 & 0 & 1\\ l_4 & 0 & 0 & 0\end{bmatrix}$$

$$\begin{bmatrix}-\alpha_1-l_1 & 1 & 0 & 0\\ -\alpha_2-l_2 & 0 & 1 & 0\\ -\alpha_3-l_3 & 0 & 0 & 1\\ -\alpha_4-l_4 & 0 & 0 & 0\end{bmatrix}$$

PASOS.

1. Función de transferencia.
2. Espacios de estado.
3. Comprobar observabilidad y controlabilidad.
4. Hallar las k por método de ackerman.
5. Polinomio de Ackerman. Función PhiAemp. (Kemp, Ki2, Ki)
6. SIMULINK. (ampl 0.01, f 100Hz)

SERVOSISTEMA

Si salen mas de dos K se agrega un integrador.

Ubicación de polos por formula de Ackerman.

$$k=[0\quad 0\quad \dots\quad 1]\mathbb{C}^{-1}\emptyset(A)$$

Donde $\mathbb{C}$ es la matriz de controlabilidad del sistema y $\emptyset(A)$ viene dado por:

$$\emptyset(A)=\lambda^n+\alpha_1\lambda^{n-1}+\alpha_2\lambda^{n-2}+\cdots+\alpha_{n-1}\lambda+\alpha_n$$

Siendo $\alpha_1,\alpha_2,\dots,\alpha_n$ los coeficientes de la ecuación característica deseada.

MATLAB

Contolador por observador de estados

Primero establecemos nuestra planta:

```
clear all
clc
%Funcion de transferencia, numerador = b,
denominador =a.
b = [5 1 2];
a = [1 10 25];
G = tf(b,a)
%Matrices de estados
[A,B,C,D] = tf2ss(b,a)
sys = ss(A,B,C,D);
```

Controlabilidad y observabilidad:

Aqui se comprueba si el sistemas es controlable y observable por medio de las funciones ***ctrb*** y ***obsv***.

```
co = ctrb(A,B);
Unco = length(A) - rank(co);
if Unco == 0
    men = 'Es controlable';
else
    men = 'No es controlable';
end
disp(men)
Ob = obsv(A,C);
Unobsv = length(A)-rank(Ob);
if Unobsv == 0
    men2 = 'Es observable';
else
    men2 = 'No es observable';
end
disp(men2)
```

Establecemos el observador:

El tiempo de establecimiento del observador tiene que ser 10 veces mas rapido que el tiempo del controlador.

```
N = [C;C*A];
n = rank(A);
zitaobser = 1;
tsobser = 0.1;
wno = 4/(zitaobser*tsobser);
podeseadoobserva = [1 2*zitaobser*wno wno^2];
phiAtob = podeseadoobserva(1)*(A')^2+...
    podeseadoobserva(2)*(A')+...
    podeseadoobserva(3)*eye(n);
Ninv=(N^(-1));
ke = phiAtob*Ninv*[0;1]
```

Controlador:

establecemos los parametros que tendra nuestro controlador como el ***zita*** y el *tiempo de* ***establecimiento.***

Con esto encontraremos el polinomio deseado del controlador:

```
zitadesecont = 0.8;
tsdesecont = 0.1;
wndcont = 4/(zitadesecont*tsdesecont);
Aempa = [A,[0;0]; -C,0]
Bempa = [B;0]
pdeseadocontrolador = conv([1 2*zitadesecont*wndcont
wndcont^2],[1 (200*zitadesecont*wndcont)])
```

Funcion de polinomio de Akerman:

Hacemos que la ***Aempaquetada*** se reemplaze en cada una de las S del polinomio deseado:

```
phiAemp=(Aempa^3)+pdeseadocontrolador(2)*(Aempa^2)+pdeseadoco
ntrolador(3)*(Aempa)+pdeseadocontrolador(4)*eye(3)
```

Calculo de las K:

Procedemos a encontrar cada una de las K las cuales seran las encargadas de controlar la planta:

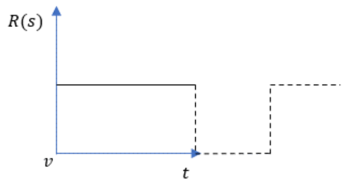
```
kemp = [0 0 1]*inv([Bempa Aempa*Bempa
(Aempa^2)*Bempa])*phiAemp
K12 = [kemp(1:2)]
ki = -kemp(3)
```

MATLAB

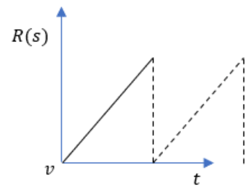
```
% Control por VARIABLES DE ESTADO
% Realimentación de Estados + Referencia
% Tiempo Discreto
%tercer orden
clc
clear all
close all

Ts = 1; %Periodo de Muestreo
B = [0 0 1 0.5];      %Numerador
A = [1 -1 0.01 0.12]; %Denominador
Gd = tf(B,A)

%% Espacio de estados
[Ad,Bd,Cd,Dd] = tf2ss(B,A)
% Ad = [0 1 0;
%       0 0 1;
%       -0.12 -0.01 1];
% Bd =[0;0;1];
% Cd=[0.5 1 0];
% Dd = 0;

%% Sistema aumentado: Planta + Controlador
Aa = [Ad Bd;zeros(1,length(Ad)) 0];
Ba = [zeros(length(Bd),1);1];
Ca = [Cd 0];
Da = [Dd 0];
sys_a =ss(Aa,Ba,Ca,0,Ts)
%% Controlabilidad
Co = [Ba Aa*Ba Aa^2*Ba Aa^3*Ba]
rank(Co)
%% Ecuación característica del Sistema
E_Ca=poly(eig(sys_a));
%% Realimentación de Estados
% Polos deseados sin integrador
Ed = [0.5+i*0.5;0.5-i*0.5;0.1;0.2];
% Ed = [0;0;0;0];
Ps = poly(Ed);
%% Ganancia k_bar
k_bar=Ps(2:end)-E_Ca(2:end);
%% Transformación de Similitud
Ci = toeplitz(E_Ca(1:end-1));
Ci = triu(Ci);
Q=(Co*Ci)^(-1)
%% Ganancia de Realimentación de Estados
k = k_bar*Q
% k = place(Aa,Ba,Ed)
% k = acker(Aa,Ba,Ed)
%% Ganancias del controlador
Km = [Ad-eye(length(Ad)) Bd;
      Cd*Ad              Cd*Bd];
In = [zeros(1,length(k)-1) 1];
K = (k+In)/Km;
K1 = K(1:end-1)
K2 = K(end)
%% Lazo cerrado
% Af= [Ad                Bd;
%      K1-K1*Ad-K2*Cd*Ad  1-K1*Bd-K2*Cd*Bd];
% Bf = [zeros(length(Bd),1);K2];
% Cf = Ca;
Af= [Ad-Bd*K1          Bd*K2;
     -Cd*Ad+Cd*Bd*K1  1-Cd*Bd*K2];
Bf = [zeros(length(Bd),1);1];
Cf = Ca;

sys_f=ss(Af,Bf,Cf,0,Ts)
```

ERRORES EN ESTADO ESTACIONARIO.

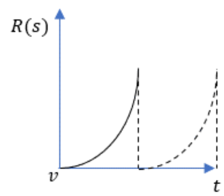
Entrada de posición.



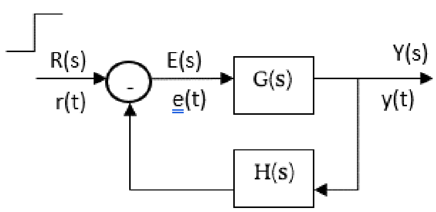
Entrada de velocidad.



Entrada de aceleración.



El error de estado estable es el error después de que la respuesta transitoria ha decaído, dejando solo la respuesta continua.



$$\frac{E(s)}{R(s)} = \frac{1}{1 + G(s)H(s)}$$

$$e_{ss} = \underbrace{\lim_{t\to\infty}[e(t)]}_{Tiempo} = \underbrace{\lim_{s\to 0}[sE(s)]}_{Laplace}$$

$$e_{ss} = \lim_{s\to 0} \frac{s}{1 + G(s)H(s)} R(s)$$

SIEMPRE AL DERIVAR HAY QUE IMPLEMENTAR UN FILTRO PASA BAJO (filtro derivativo para corregir el ruido).

Errores en estado permanente.

$$e_{ss} = \lim_{s\to} \frac{sR(s)}{1 + G(s)H(s)}$$

Entre mayor sea la ganancia menos error en estado estacionario se tiene.

Constante k_p de error estático de "posición".

Constante k_v de error estatico de "velocidad".

Constante k_a de error estatico de "aceleración".

Error estático de posición.

Error en estado estable para una entrada escalón; $R(s) = \frac{A}{s}$

$$e_{ss} = \lim_{s\to 0} \frac{s}{1 + G(s)H(s)} \frac{A}{s}$$

$$e_{ss} = \lim_{s\to 0} \frac{A}{1 + G(s)H(s)}$$

$$e_{ss} = \lim_{s\to 0} \frac{A}{1 + G(0)H(0)}$$

Se define:

$$k_p = \lim_{s\to 0} G(s)H(s) = G(0)H(0)$$

Por lo tanto:

$$e_{ss} = \frac{A}{1 + k_p}$$

Sistema tipo 0 (N=0)

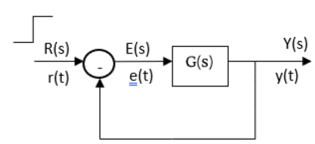
$$k_p = \lim_{s\to 0} = \frac{K(\tau_{z_1}s + 1)(\tau_{z_2}s + 1)\ldots(\tau_{z_m}s + 1)}{s^N(\tau_{p_1}s + 1)(\tau_{p_2}s + 1)\ldots(\tau_{p_q}s + 1)} = K$$

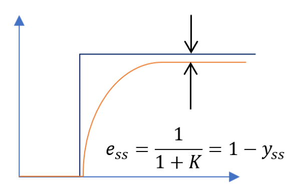
$$e_{ss} = \frac{A}{1 + K_P} = \frac{A}{1 + K}$$

Sistema tipo 1 o mayor (N=1, 2, 3, …)

$$k_p = \lim_{s\to 0} = \frac{K(\tau_{z_1}s + 1)(\tau_{z_2}s + 1)\ldots(\tau_{z_m}s + 1)}{s^N(\tau_{p_1}s + 1)(\tau_{p_2}s + 1)\ldots(\tau_{p_q}s + 1)} = \infty$$

$$e_{ss} = \frac{A}{1 + K_P} = \frac{A}{1 + \infty} = 0$$





Error estático de velocidad.

Error en estado estable para una entrada escalón; $R(s) = \frac{A}{s^2}$

$$e_{ss} = \lim_{s\to 0} \frac{s}{1 + G(s)H(s)} \frac{A}{s^2}$$

$$e_{ss} = \lim_{s\to 0} \frac{A}{s + sG(s)H(s)}$$

$$e_{ss} = \lim_{s\to 0} \frac{A}{sG(0)H(0)}$$

Se define:

$$k_v = \lim_{s\to 0} sG(s)H(s)$$

Por lo tanto.

$$e_{ss} = \frac{A}{k_v}$$

Sistema tipo 0.

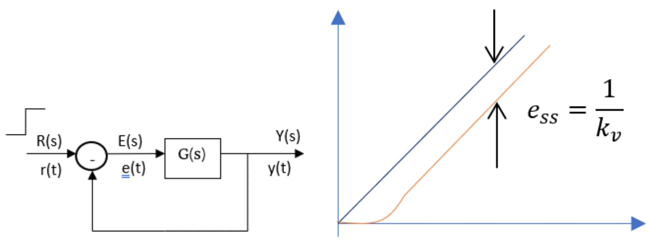
$$e_{ss} = \frac{A}{k_v} = \frac{A}{0} = \infty$$

Sistemas tipo 1.

$$e_{ss} = \frac{A}{k_v} = \frac{A}{K}$$

Sistemas tipo 2 o mayor.

$$e_{ss} = \frac{A}{\infty} = 0$$



Error estático de aceleración.

Error en estado estable para una entrada escalón; $R(s) = \frac{A}{s^3}$

$$e_{ss} = \lim_{s\to 0} \frac{s}{1 + G(s)H(s)} \frac{A}{s^3}$$

$$e_{ss} = \lim_{s\to 0} \frac{A}{s^2 + s^2G(s)H(s)}$$

$$e_{ss} = \lim_{s\to 0} \frac{A}{s^2G(s)H(s)}$$

Sistema tipo 0 y tipo 1.

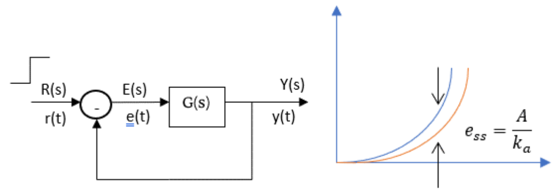
$$e_{ss} = \frac{A}{k_a} = \frac{A}{0} = \infty$$

Sistema tipo 2.

$$e_{ss} = \frac{A}{k_a} = \frac{A}{K}$$

Sistema tipo 3 o mayor.

$$e_{ss} = \frac{A}{\infty} = 0$$



Tablas.

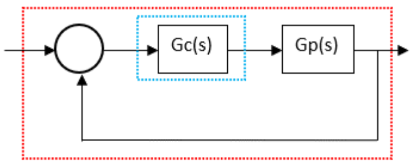
En terminos de K.

Tipo de sistema	$r(t) = A$	$r(t) = At$	$r(t) = \frac{At^2}{2}$
0	$\frac{A}{(1+K)}$	∞	∞
1	0	$\frac{A}{K}$	∞
2	0	0	$\frac{A}{K}$

En terminus de las constantes.

Tipo de sistema	$r(t) = A$	$r(t) = At$	$r(t) = \frac{At^2}{2}$
0	$\frac{A}{(1+K_p)}$	∞	∞
1	0	$\frac{A}{K_v}$	∞
2	0	0	$\frac{A}{K_a}$

COMPENSADORES DE FRECUENCIA.

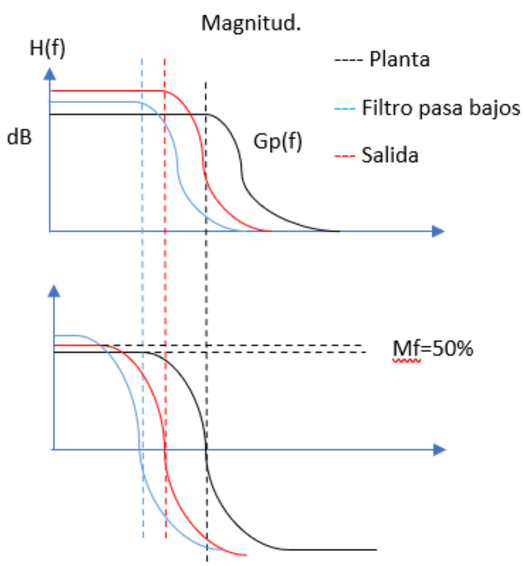


1. Compensador en atraso.

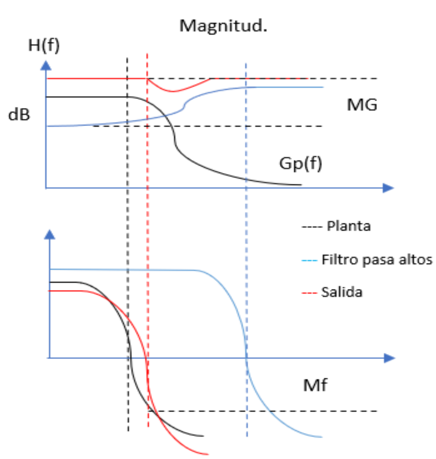
(Mayor estabilidad, más lento)

Diagrama de Bode. (Registra la frecuencia)

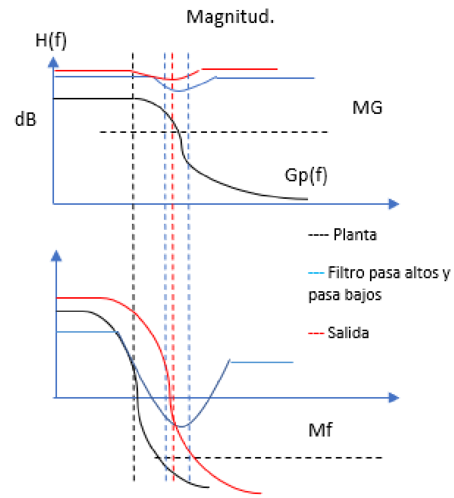
Lazo cerrado.



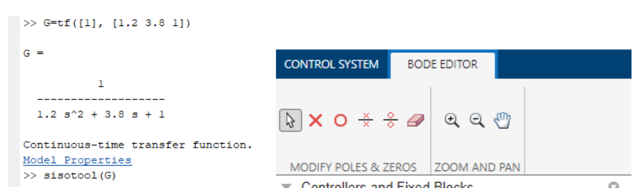
2. Compensador en adelanto

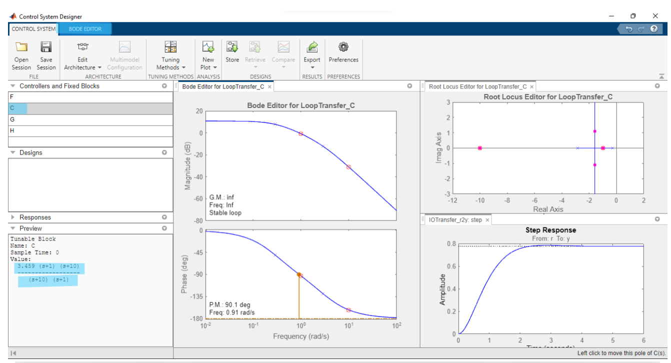


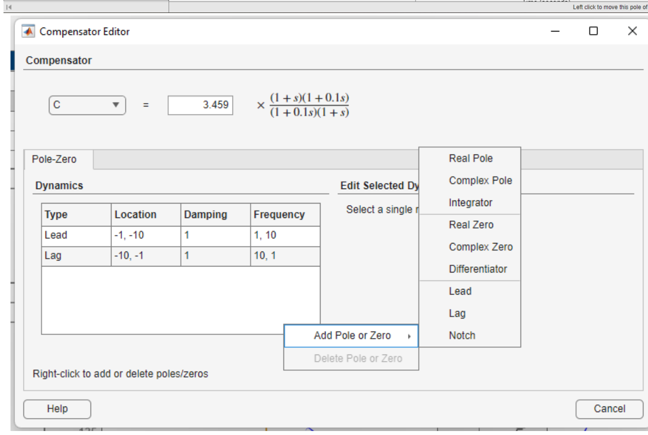
3. Compensador en atraso adelanto.



MATLAB







PRIMER CORTE MATLAB.

```
clc;
clear
%Funcion de transferencia, numerador = b,
denominador =a.
b = [5 1 2];
a = [1 10 25 102];
G = tf(b,a)
%Matrices de estados
[A,B,C,D] = tf2ss(b,a)
sys = ss(A,B,C,D);
%Observabilidad
Ob = obsv(sys)
rank(Ob)
%Controlabilidad
Co = ctrb(sys)
rank(Co)
co = ctrb(A,B);
Unco = length(A) - rank(co);
if Unco == 0
    men = 'Es controlable';
else
    men = 'No es controlable';
end
disp(men)

Ob = obsv(A,C);
Unobsv = length(A)-rank(Ob);
if Unobsv == 0
    men2 = 'Es observable';
else
    men2 = 'No es observable';
end
disp(men2)


%Criterio de estabilidad Routh Hurwitz.
roots(a)
figure
subplot(2,1,1)
%Grafica Lugar de raices con ruta.
rlocus(b,a)
grid;

subplot(2,1,2)
title ('Lugar de las raíces de G(s) =
(10s^2+2s+1)/(6s^2+8s+12)')
%Grafica de raices,  polos y ceros
pzmap(G)

figure
subplot(2,1,1)
Gct1 = pidtune(G,'PID') %Usar pidtool
Hd = c2d(G,0.1,'foh')
step(G,'-r',Hd,'b')

subplot(2,1,2)
%Grafica de impulso
impulse(G)
```

Función de transferencia.

```
G =

        5 s^2 + s + 2
  --------------------------
  s^3 + 10 s^2 + 25 s + 102
```

Espacios de estado.

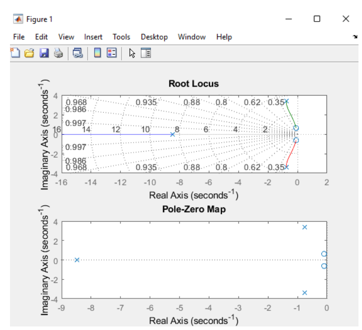
```
A =

   -10   -25  -102
     1     0     0
     0     1     0

B =

     1
     0
     0

C =

     5     1     2

D =

     0
```

Observabilidad y controlabilidad.

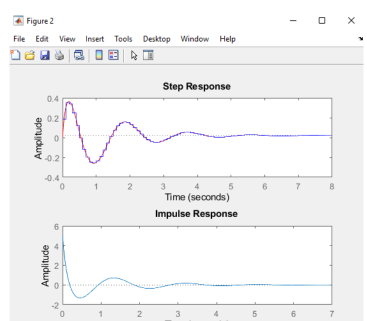
```
Ob =

         5           1           2
       -49        -123        -510
       367         715        4998

ans =

     3

Co =

     1   -10    75
     0     1   -10
     0     0     1

ans =

     3

Es controlable
Es observable
```

Criterio de estabilidad Routh Hurwitz.

```
ans =

  -8.4702 + 0.0000i
  -0.7649 + 3.3848i
  -0.7649 - 3.3848i
```



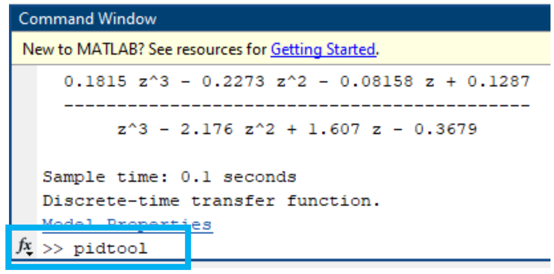
Grafica de impulso de entrada vs salida PID



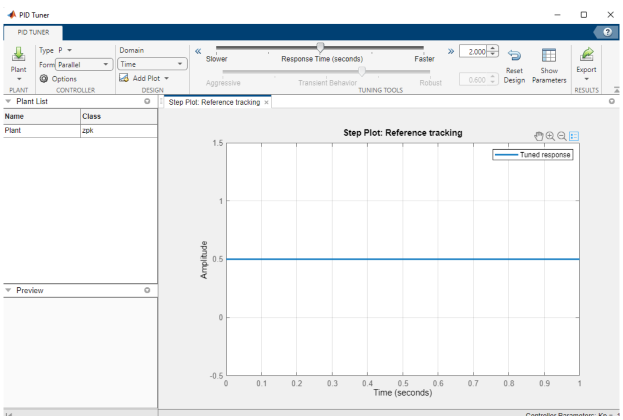
PRIMER CORTE MATLAB.

Para hallar los valores de Kp, Ki, y Kd. Utiliza PID tool.

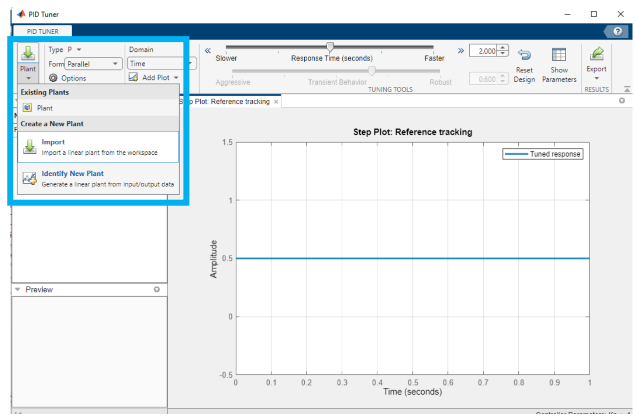
1. Escribe el comando "pidtool"



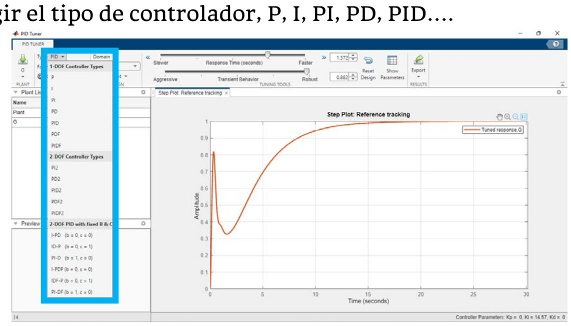
2. Inmediatamente se abre una ventana adicional.



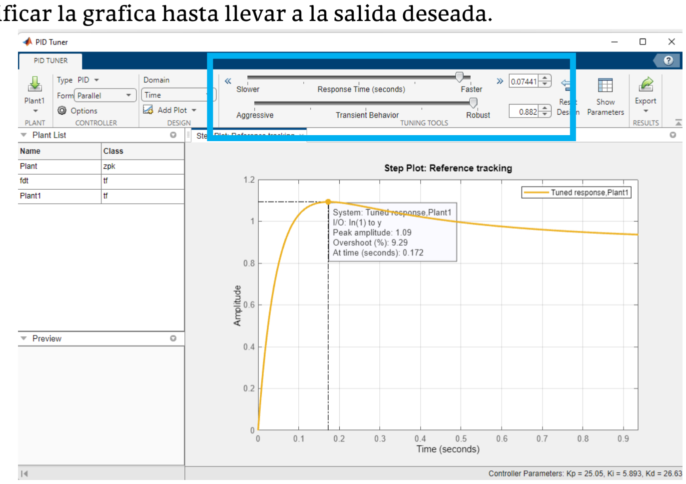
3. Es necesario ingresar la planta que se desea controlar.



4.En la parte superior podeos elegir el tipo de controlador, P, I, PI, PD, PID....



5. Por último podemos modificar la grafica hasta llevar a la salida deseada.



SEGUNDO CORTE MATLAB.

```
comp=tf('s');
num=[1 3];
den=[1 0 2 10];
fdt=tf(num,den)
[matrizA,matrizB,matrizC,matrizD] = tf2ss(num,den);
A=matrizA
B=matrizB
C=matrizC
D=matrizD
%%%%%%%Determinacion de controlabilidad y observabilidad%%%%%%
M=[B,A-B,A^2*B]
N=[C;C*A;C*A^2]
n=rank(A)
RM= rank(M)
RN= rank(N)
if RM==n
    controlable=1
else
    controlable=0
end
if RN==n
    observable=1
else
    observable=0
end
```

Punto 1: PID discreto

```
Gct1 = pidtune(fdt,'PID') %Usar pidtool
Hd = c2d(fdt,0.1,'foh')
step(fdt,'-r',Hd,'b')
```

Punto 2: Diseñar retro con observador

```
%U vector
zitaobser = 1;
tsobser = 0.7; %Debe ser 10 veces menor al del controlador
wno = 4/(zitaobser*tsobser);
podeseadoobserva = [1 2*zitaobser*wno wno^2];
phiAtob = podeseadoobserva(1)*(A')^2+...
    podeseadoobserva(2)*(A')+...
    podeseadoobserva(3)*eye(n)
Ninv=(N^(-1));
ke = phiAtob*Ninv*[0;0;1]
```

función de transferencia.

```
fdt =

      s + 3
  --------------
  s^3 + 2 s + 10

Continuous-time transfer function.
Model Properties
```

Matrices de estado.

```
A = 3×3
     0    -2   -10
     1     0     0
     0     1     0

B = 3×1
     1
     0
     0

C = 1×3
     0     1     3

D = 0
```

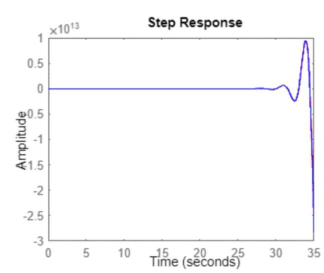
Controlabilidad y observabilidad.

```
M = 3×5
     1    -1    -3   -11    -2
     0     1     0     0     0
     0     0     1     0     1

N = 3×3
     0     1     3
     1     3     0
     3    -2   -10

n = 3
RM = 3
RN = 3
controlable = 1
observable = 1
```

PID Discreto.

```
Gct1 =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 79.1, Ki = 63.3, Kd = 24.7

Continuous-time PID controller in parallel form.
Model Properties
Hd =

  0.00179 z^3 + 0.006361 z^2 - 0.003616 z - 0.00154
  -------------------------------------------------
           z^3 - 2.975 z^2 + 2.985 z - 1

Sample time: 0.1 seconds
Discrete-time transfer function.
Model Properties
```

Señal sin controlar.

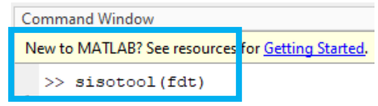


SEGUNDO CORTE MATLAB.

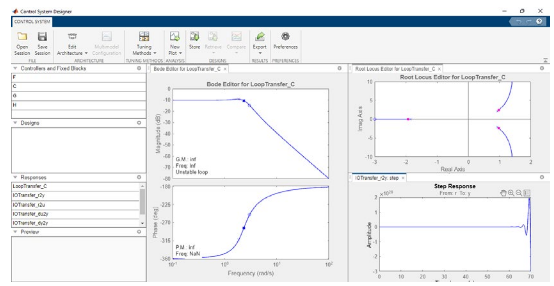
Punto 3: Diseñar retro discreto con observador

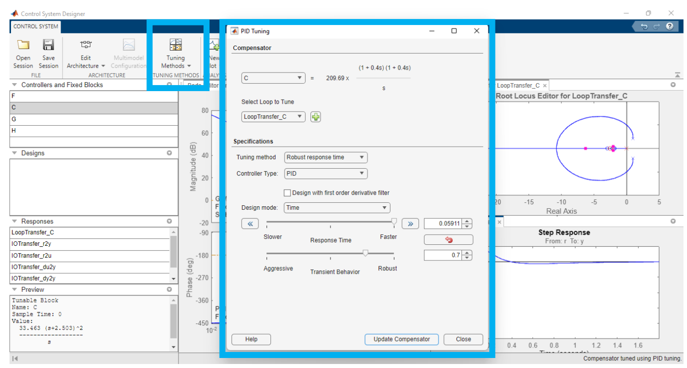
```
zitadesecontdis=0.9;
tssdesecontdis=10;
wdcontdis=4/(zitadesecontdis*tssdesecontdis);
ws=20*wdcontdis
TM=(2*pi)/ws
[m,n]=size(A)
syms z s t
G1=(s*eye(n)-A)
G2= inv(G1)
GT= ilaplace(G2)
G3=(subs(GT,t,TM)) %% Matriz A
G=vpa(G3)
H=vpa(int(GT,0,TM)*B) %% Matriz B
%%%%%% Determinacion de controlabilidad y
observabilidad %%%%%
Md=[H,G*H,G^2*H]
Nd=[C;C*G;C*G^2]
n= rank(G)
RM= rank(Md)
RN= rank(Nd)
if RM==n
    controlable=1
else
    controlable=0
end
if RN==n
    observable=1
else
    observable=0
end
magpolodeseado=exp(((2*pi*zitadesecontdis)/sqrt(1-
zitadesecontdis^2))*(wdcontdis/ws))
angdeseado=2*pi*(wdcontdis/ws)
[real,imag]=pol2cart(magpolodeseado,angdeseado)
polodeseado=real+i*imag
polodeseadoconj=real-i*imag
pdeseadocontroladordis=conv(conv(conv([1 -
polodeseado],[1 -polodeseadoconj]),...
    [1 0.0005]),[1 0.0005])
Gempa=[G, [0;0;0];C*G,1]
Hempa= [H; C*H]
phiGemp=(Gempa^4)+pdeseadocontroladordis(2)*(Gempa^
3)+pdeseadocontroladordis(3)*...
```

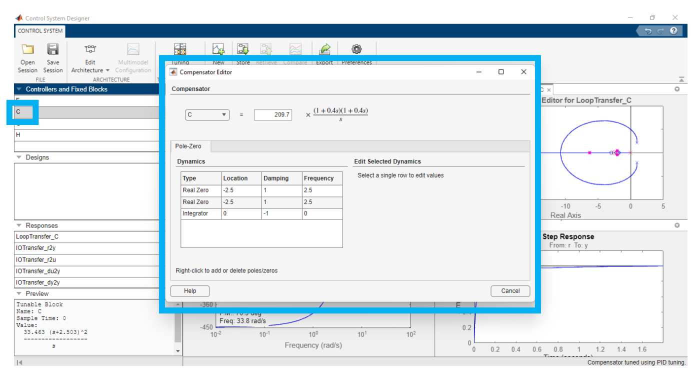
Ingresa el comando



Inmediatamente se abre la siguiente Ventana.







SEGUNDO CORTE MATLAB.

```
(Gempa^2)+pdeseadocontroladordis(4)*(Gempa)+pdeseadocontroladordis(5)*eye(4)
kempdis=[0 0 0 1]*inv([Hempa Gempa*Hempa
(Gempa^2)*Hempa (Gempa^3)*Hempa])*phiGemp
k13dis=kempdis(1:3)
kidis=kempdis(4)
k13ev=eval(k13dis)
kiev=eval(kidis)
%%%%%%%%%%%% Observador de estados en discreto
%%%%%%%%%%%%%%%%%%
H1= eval(H)
polinomiodeseaobservadis=z^(n)
phiGobser=G^(n)
Ninvv=(Nd^(-1))
kedis = eval(phiGobser*Ninvv*[0;0;1])
Gev=eval(G)
```

Punto 4: Diseñar compensador lead-led

MP=70° y Oshoot=10%

```
% sisotool(fdt) %Programa para hacer el compensador
% Value:
%   18951 (s+1)^3 (s+0.8032) (s+10)^2
%   ---------------------------------
%      (s+248.4) (s+10)^2 (s+1)^3
num1=18951*(s+1)^3*(s+0.8032)*(s+10)^2;
den1=(s+248.4)*(s+10)^2*(s+1)^3;
comp=num1/den1
comp1=eval(comp)
gc=tf([18951 (9513402/625)],[1 (1242/5)])
step(gc)
```