CONTROL DIGITAL CON MATLAB

Por: M. I. Jorge A. Polanía P.

Contenido

1. LA TRANSFORMADA Z

1.1 DEFINICIÓN DE LA TRANSF_Z

Sea** $x(t)$ **: Señal en tiempo continuo

$x(kT)$ ***: Señal en tiempo discreto***

La transformada z se define de la siguiente manera:

$$Z[x(kT)] = X(z) = \sum_{k=0}^{\infty} x(kT) z^{-k}$$

$$= x(0) + x(T)\,z^{-1} + x(2T)\,z^{-2} + \cdots\cdots + x(kT)\,z^{-k} + \cdots\cdots$$

1.2 TZ DE FUNCIONES ELEMENTALES

1.2.1 ESCALÓN UNITARIO

La función del escalón unitario es la siguiente:

$$x(kT) = 1, \quad kT \geq 0 \;, \; y \quad x(kT) = 0, \; kT < 0$$

$$\sum_{k=0}^{\infty} 1 * z^{-k} = \sum_{k=0}^{\infty} z^{-k}$$

$$X(z) = Z[x(kT)] = Z[1(kT)] =$$

$X(z) = 1 + z^{-1} + z^{-2} + z^{-3} + \cdots\cdots$ ***, entonces,***

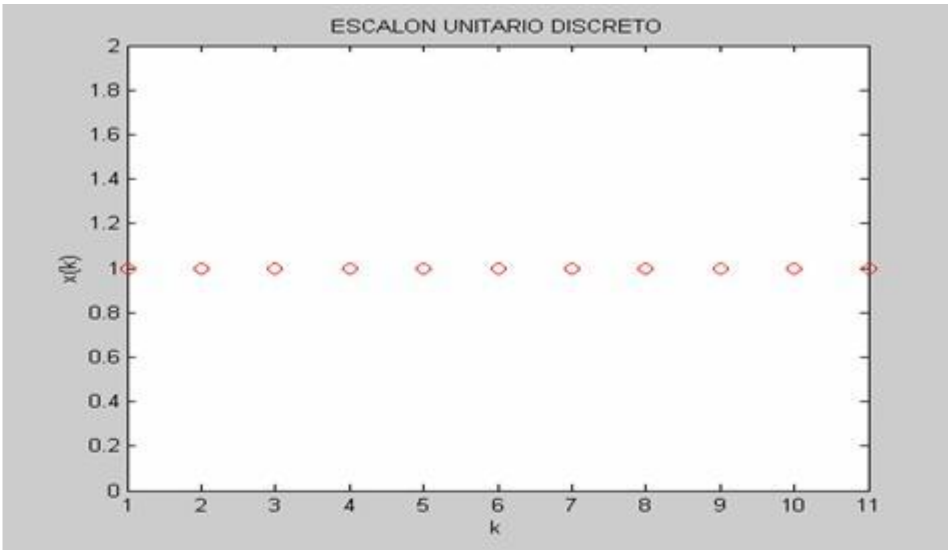
$$X(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

La anterior relación se obtiene por Matlab de la siguiente forma:

```
>> syms z k                          % variables simbólicas z, k

>> symsum(z^(-k),k,0,inf)
```

Para graficar la señal escalón unitaria discreta por Matlab, se hace:

```
% GENERACIÓN DE ESCALÓN UNITARIO DISCRETO

x = ones (1,11);   % define once valores de 1's

v = [ 0 10 0 2];    % define valores de ejes

axis (v);

plot (x,'ro')       % grafica círculos de color rojo

xlabel ('k')        % asigna rotulo al eje x

ylabel ('x(k)')    % asigna rotulo al eje y

title (‘ESCALON UNITARIO DISCRETO’)
```



1.2.2 RAMPA UNITARIA

La función de la rampa unitaria es:

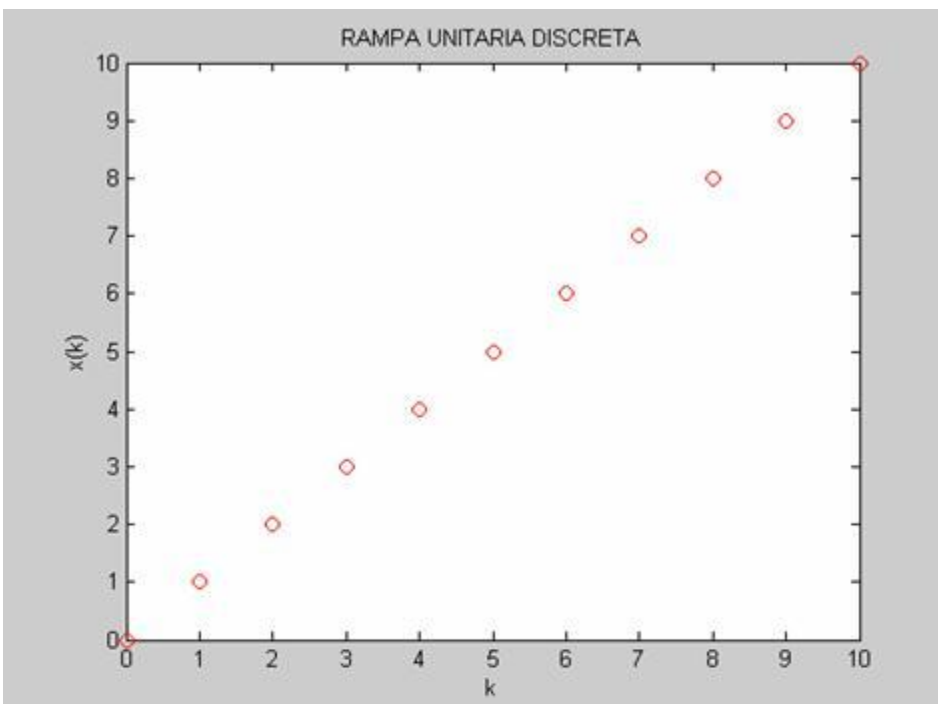
$$x(kT) = \begin{cases} kT, & \text{para } k = 0, 1, 2, 3, \cdots\cdots \\ 0, & \text{para } k < 0 \end{cases} \qquad \text{(señal discreta)}$$

$$X(z) = Z[x(kT)] = Z[kT] = \sum_{k=0}^{\infty} kT*z^{-k} = \sum_{k=0}^{\infty} Tk*z^{-k}$$

$$X(z) = T(z^{-1} + 2z^{-2} + 3z^{-3} + \cdots\cdots\cdots\cdots)$$

$$X(z) = \frac{T}{(1-z^{-1})^2} = \frac{Tz}{(z-1)^2}$$

```
% GENERACIÓN DE LA RAMPA UNITARIA DISCRETA

k = 0:10;                  % define valores de k

x = k;                     % función rampa para x

axis([0 10 0 10]);         % define ejes

grid                       % rejilla para grafica

plot(k, x,'ro')            % grafica x en función de k

xlabel('k');               % rotulo para eje x

ylabel('x(k)');            % rotulo para eje y

title('RAMPA UNITARIA DISCRETA')
```



1.2.3 POTENCIAL: a^k (a = constante)

$$x(k) = \begin{cases} a^k, & \text{para } k = 0, 1, 2, 3, \cdots\cdots \\ 0, & \text{para } k < 0 \end{cases}$$

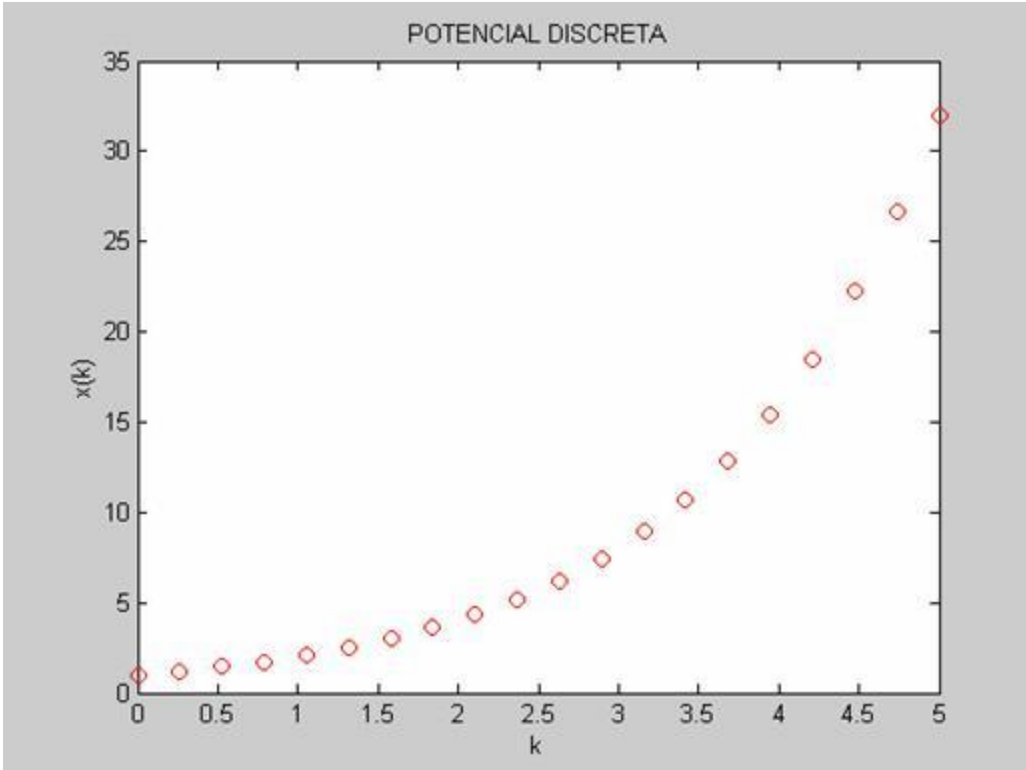
(señal discreta)

$X(z) = Z[x(k)] = na^k \cdot z^{-k}$

$X(z) = 1 + a z^{-1} + a^2 z^{-2} + a^3 z^{-3} + \cdots\cdots\cdots\cdots$, entonces,

$$X(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}$$

```
%GENERACION DE LA FUNCION POTENCIAL  x(k) = 2 k
k=linspace(0,5,20);    % define valores de k
x=2.^ k;               % función potencial
grid                   % rejilla para grafica
plot(k, x,'ro')        % grafica x en función de k
xlabel('k');           % rotulo para eje x
ylabel('x(k)');        % rotulo para eje y
title('POTENCIAL DISCRETA')
```



1.2.4 EXPONENCIAL: e^{-akT} (a = constante)

La función exponencial es de la forma:

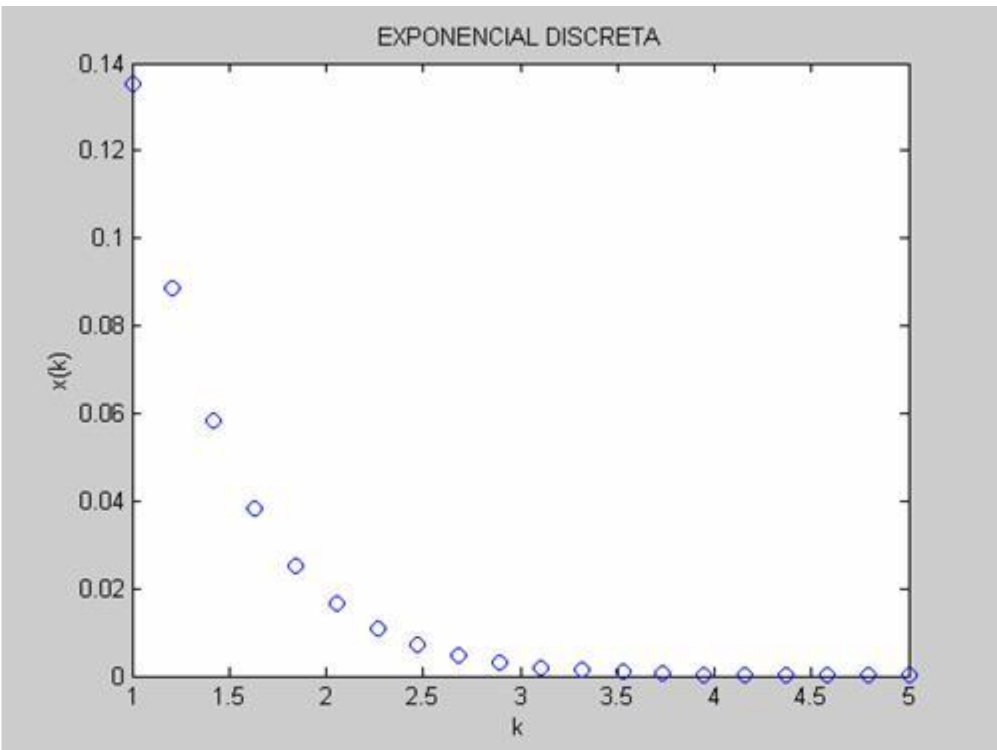
$$x(k) = \begin{cases} e^{-akT}, & \text{para } k = 0, 1, 2, 3, \cdots\cdots \\ 0, & \text{para } k < 0 \end{cases} \qquad \textit{(señal discreta)}$$

$X(z) = Z[x(k)] = n\ e^{-akT} z^{-k}$

$X(z) = 1 + e^{-aT} z^{-1} + e^{-2aT} z^{-2} + e^{-3aT} z^{-3} + \cdots\cdots\cdots\cdots$, ***entonces,***

$$X(z) = \frac{1}{1 - e^{-aT} z^{-1}} = \frac{z}{z - e^{-aT}}$$

```
%GENERACION DE LA FUNCION EXPONENCIAL  x(k) = e -2k
k = linspace (1,5,20);    % define valores de k con espaciamiento lineal
x = exp(-2* k);                % función exponencial
grid                           % rejilla para grafica
plot(k, x,'bo')                % grafica x en función de k
xlabel('k');                   % rotulo para eje x
ylabel('x(k)');                % rotulo para eje y
title('EXPONENCIAL DISCRETA')
```



1.2.5 SENOIDAL : sen(wkT)

La función es de la forma:

$$x(k) = \begin{cases} sen(wkT), & \text{para } k = 0, 1, 2, 3, \cdots\cdots \\ 0, & \text{para } k < 0 \end{cases} \qquad \text{(señal discreta)}$$

X(z) = Z[x(k)] = Z[sen(wkT)]

Por al ecuación de Euler:

Sen(wkT) = (1/2j) (e^{jwkT} - e^{-jwkT}), reemplazando,

Z[sen(wkT)] = Z[(1/2j)(e^{jwkT} - e^{-jwkT})],

aplicando la transf_z de la exponencial,

$$X(z) = \frac{1}{2j} \cdot \frac{1}{1 - e^{jwT} z^{-1}} - \frac{1}{2j} \cdot \frac{1}{1 - e^{-jwT} z^{-1}}$$

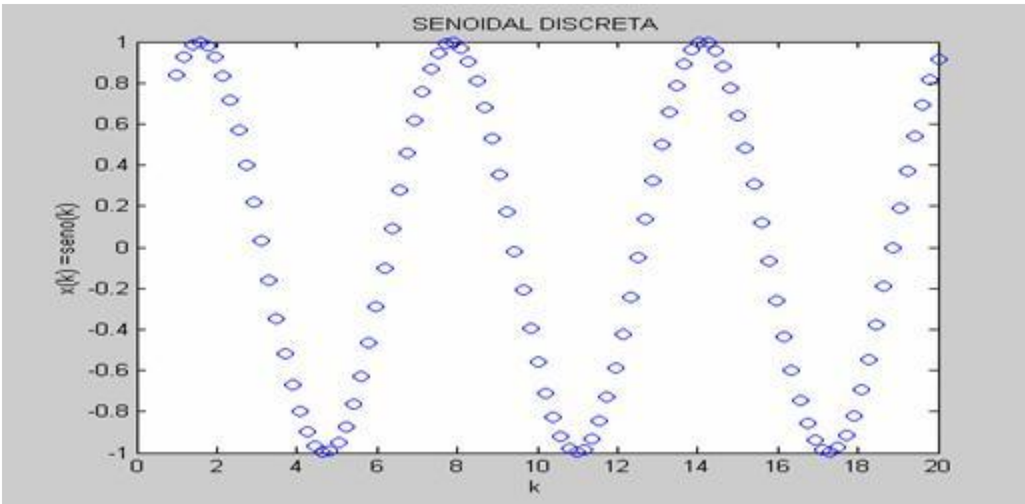
reemplazando las exponenciales :

$$X(z) = \frac{z^{-1} sen(wT)}{1 - 2z^{-1}\cos(wT) + z^{-2}}$$

%GENERACION DE LA FUNCION SENO: x(k) = sen(wkT)

```
k = linspace(1,20);          % define valores de k con espaciamiento lineal
x = sin(k);                  % función exponencial
grid                         % rejilla para grafica
plot(k, x,'bo')              % grafica x en función de k
```

```
xlabel('k');                          % rotulo para eje x
ylabel('x(k) =seno(k)');     % rotulo para eje y
title('SENOIDAL DISCRETA')
```



1.3 PROPIEDADES Y TEOREMAS.

1.3.1 MULTIPLICACIÓN POR UNA CONSTANTE

$Z[a\ x(k)] = a\ Z[x(k)] = a\ X(z)$

1.3.2 LINEALIDAD

Si $x(kT) = a\ f(kT) + b\ g(kT)$, entonces,

$X(z) = a\ F(z) + b\ G(z)$

1.3.3 MULTIPLICACIÓN POR a^k

Si $y(kT) = a^k\ x(kT)$, entonces,

$\sum_{k=0}^{\infty}$

$Z[a^{k}\, y(kT)] =$

$$a^{k}\, x(kT)\, z^{-k}$$

$$\sum_{k=0}^{\infty} x(kT)\,(a^{-1} z)^{-k} = X(a^{-1} z)$$

1.3.4 TEOREMA DEL TRASLACIÓN

Si $y(kT) = e^{-akT}\, x(kT)$*, entonces,*

$Z[e^{-akT}\, x(kT)] =$

$$\sum_{k=0}^{\infty} e^{-akT}\, x(kT)\, z^{-k}$$

$$\sum_{k=0}^{\infty} x(kT)\,(e^{aT} z)^{-k} = X(e^{aT} z)$$

1.3.5 TEOREMA DEL CORRIMIENTO

Corrimiento hacia atrás:

$Z[x(k-n)T] = z^{-n}\, Z[x(k)] = z^{-n}\, X(z)$

Corrimiento hacia adelante:

$$Z[x(k+n)T] = z^{n}\left[X(z) - \sum_{k=0}^{n-1} x(kT)*z^{-k}\right]$$

$$= z^{n}\, X(z) - z^{n}\, x(0) - z^{n-1}\, x(1) - z^{n-2}\, x(2) - \cdots - z\, x(n-1)$$

Ejemplo:

$Z[x(k+3)T] = z^3\ X(z) - z^3\,x(0) - z^2\,x(1) - z\ x(2)$

1.3.6 SUMA DE FUNCIONES

Sea $y(k) \quad \sum_{k=0}^{k} = x(h)$, para $k = 0,1, 2, \cdots\cdots$

$y(k) = x(0) + x(1) + x(2) + \cdots\cdots\cdots + x(k-1) + x(k)$

$y(k-1) = x(0) + x(1) + x(2) + \cdots\cdots\cdots + x(k-1)$, restando estas dos expresiones,

$y(k) - y(k-1) = x(k)$, sacando Transf._Z,

$Y(z) – z^{-1}Y(z) = X(z)$, entonces despejando $Y(z)$ se tiene que:

$$Y(z) = \frac{1}{1-z^{-1}} \cdot X(z)$$

1.3.7 TEOREMA DEL VALOR INICIAL

Si el límite $\lim X(z)$ existe, entonces el valor inicial de $x(k) = x(0)$ es igual a:

$$x(0) = \lim_{z\to\infty} X(z)$$

1.3.8 TEOREMA DEL VALOR FINAL

El valor final de $x(k)$, o sea, cuando k à ∞ (Si $X(z)$ es estable) , es:

$$x(\infty) = \lim_{k\to\infty} x(k) = \lim_{z\to 1} [\,(1 – z^{-1})X(z)\,]$$

EJEMPLO 1-1

Encontrar la transformada Z de una función escalón de amplitud 4 y desfase en atraso de 3 periodos de muestreo.

Solución:

$x(kT) = 4*u(kT- 3T)$, ***asumiendo*** $T = 1$ ***por simplicidad,***

$x(k) = 4u(k-3)$

$Z[4u(k-3)] = 4Z[u(k-3)] = 4\, z^{-3} Z[u(k)]$

aplicando teorema de corrimiento en atraso

$X(z) = 4\, z^{-3}\, (1/\,(1-z^{-1})) = 4 \,/\, (z^{3} - z^{2})$

EJEMPLO 1-2

Obtener la transformada Z de $y(k) = 5^{k-2}$ ***para*** $k = 1, 2, 3, ...$ ***e igual a cero para*** $k \leq 0$.

Solución:

Sea $x(k) = 5^{k}$, ***entonces*** $y(k) = x(k-2) = 5^{k-2}$

$Z[y(k)] = Z[5^{k-2}] = Z[x(k-2)] = z^{-2} Z[x(k)] = z^{-2} Z[5^{k}] = z^{-2} * 1/(1-5\,z^{-1})$

$Z[5^{k-2}] = 1 \,/\, (z^{2} - 5\,z)$

EJEMPLO 1-3

Obtener la transformada Z de $y(k) = k\,e^{-5k}$ ***para*** $k = 1, 2, 3, ...$ ***e igual a cero para*** $k \leq 0$.

Solución:

Sea $x(k) = k$, ***entonces,*** $X(z) = z^{-1} \,/\, (1 - z^{-1})^{2}$, ***y además,*** $y(k) = e^{-5k}\, x(k)$,

Aplicando teorema de traslación,

$Z[y(k)] = Z[k\, e^{-5k}] = X(e^{5k} z)]$, reemplazando z à $e^{5k} z$, en X(z) se tiene:

$$Y(z) = (e^{5k} z)^{-1} / (1 - (e^{5k} z)^{-1})^{2} = e^{-5k} z^{-1} / (1 - e^{-5k} z^{-1})^{2}$$

[***EJEMPLO 1-4***]()

Determinar el valor inicial x(0) de una señal si su transformada Z es igual a :

$$X(z) = \frac{(1-e^{-5k})\, z^{-1}}{(1- z^{-1})(1 - e^{-5k} z^{-1})}$$

Aplicando el Teorema de valor inicial,

$$x(0) = \lim_{z \to \infty} X(z) = \frac{(1-e^{-5k})\, \infty^{-1}}{(1- \infty^{-1})(1 - e^{-5k} \infty^{-1})} = 0$$

[***EJEMPLO 1-5***]()

Determinar el valor final x(∞) de una señal si su transformada Z es igual a:

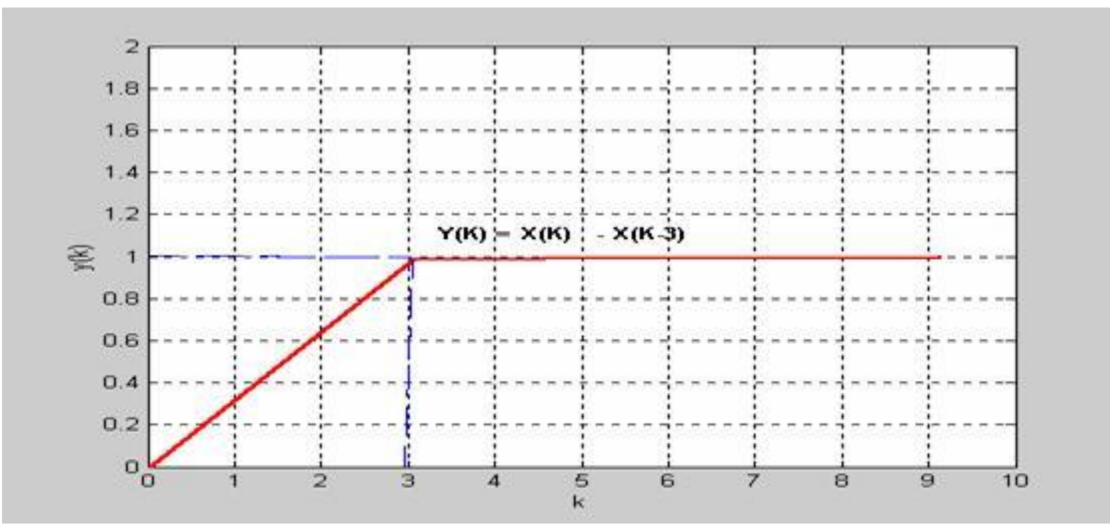
$$X(z) = \frac{1}{(1- z^{-1})} - \frac{1}{(1 - e^{-5} z^{-1})}$$

$$1- z^{-1}$$

$$x(\infty) = \lim_{k\to\infty} x(k) = \lim_{z\to 1} [\,(1 - z^{-1})X(z)\,] = \lim_{z\to 1} \quad 1 - \frac{\qquad}{1 - e^{-5} z^{-1}} = 1$$

EJEMPLO 1-6

Obtener la transformada Z de la figura dada. Tiempo de muestreo = 1.0



Si $x(k) = (1/3)k$ (rampa de pendiente 1/3)

$y(k) = x(k) - x(k-3)$, entonces,

$Y(z) = z[y(k)] = z[(1/3)k] - z^{-3} z[(1/3)k]$

$$Y(z) = \frac{1}{3}\cdot\frac{z^{-1}}{(1-z^{-1})^2} - \frac{1}{3}\cdot\frac{z^{-3}\cdot z^{-1}}{(1-z^{-1})^2} = \frac{1}{3}\cdot\frac{z^{-1}(1-z^{-3})}{(1-z^{-1})^2}$$

1.4 TRANSFORMADA Z INVERSA

Con la transformada Z inversa se obtiene la señal discreta en los instantes de muestreo x(kT). Los siguientes son los métodos más utilizados para obtener la transformada Z inversa.

1.4.1 MÉTODO DE DIVISIÓN DIRECTA

El método consiste en arreglar la función X(z) en términos de z^{-1} tanto el numerador como el denominador, dividir algebraicamente el numerador entre el denominador y su cociente mediante comparación con la definición de X(z) encontrar la señal x(kT).

$$\sum_{K=0}^{\infty} x(k)\, z^{-k} = x(0) + x(1)\, z^{-1} + x(2)\, z^{-1} + x(3)\, z^{-3} + \cdots\cdots + x(k)\, z^{-k}$$

EJEMPLO 1-7

Obtener la transformada Z inversa x(k) de:

$$X(z) = \frac{5z+10}{(z-0.8)(z-0.2)} = \frac{5z+10}{z^2 - z + 0.16}$$

multiplicando numerador y denominador por z^{-2},

$$X(z) = \frac{5z^{-1} + 10z^{-2}}{1 - z^{-1} + 0.16z^{-2}}$$

dividiendo numerador entre denominador, se tiene,

$X(z) = 5\, z^{-1} + 15\, z^{-2} + 14.2\, z^{-3} + 11.8\, z^{-4} + \cdots\cdots\cdots\cdots\cdots$

Comparando con la definición de X(z), se obtiene que:

$X(0) = 0,\; x(1) = 5,\;\; x(2) = 14.2,\;\; x(3) = 11.8, \cdots\cdots\cdots$

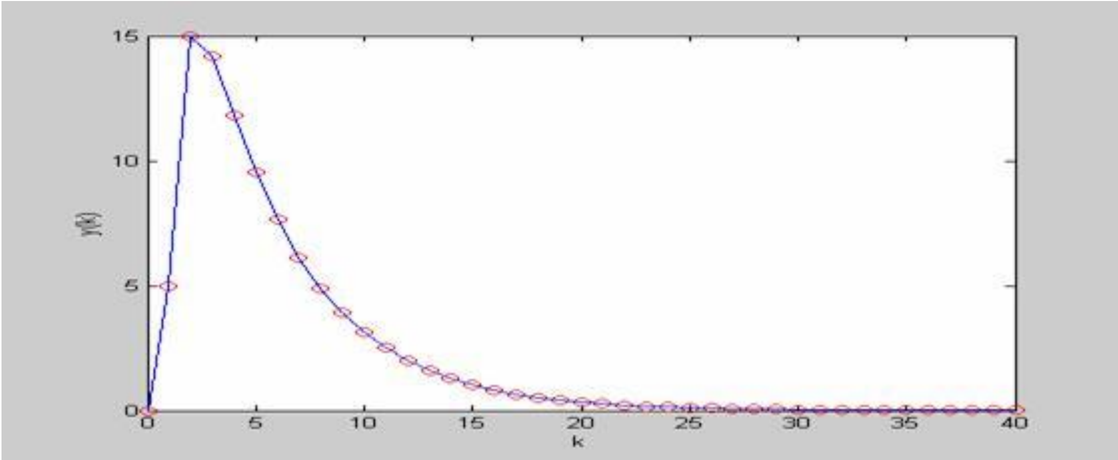
Si se quiere obtener más muestras de la señal, es mejor aplicar Matlab de esta forma:

```
% EJEMPLO 1-6: DE TRANSFORMADA Z INVERSA
x = [1 zeros(1,40)];           % para k = 40 muestras
num = [0 5 10];                % coeficientes del numerador
den = [1 -1 0.16];             % coeficientes del denominador
y = filter(num, den, x)        % obtención de las 40 muestras
k = 0:40;
plot(k, y,'ro',k, y,'-')
xlabel('k')
ylabel('y(k)')
```

Los 40 resultados obtenidos de x(0) hasta x(39) son:

0	5.0000	15.0000	14.2000	11.8000	9.5280	7.6400
6.1155	4.8931	3.9146	3.1317	2.5054	2.0043	1.6035
1.2828	1.0262	0.8210	0.6568	0.5254	0.4203	0.3363
0.2690	0.2152	0.1722	0.1377	0.1102	0.0882	0.0705
0.0564	0.0451	0.0361	0.0289	0.0231	0.0185	0.0148
0.0118	0.0095	0.0076	0.0061	0.0048	0.0039	

Cuya representación es la siguiente:



1.4.2 MÉTODO DE FRACCIONES PARCIALES

Consiste el método en expandir la función X(z) en fracciones parciales con el fin de que queden términos más simples y luego encontrar a cada fracción la transformada Z inversa.

EJEMPLO 1-8

Obtener la transformada Z Inversa de:

$$X(z) = \frac{5z^3 + 26z^2 + 44z + 29}{z^3 + 6z^2 + 11z + 6}$$

Para representar esta función en fracciones parciales usamos el comando Matlab residue que encuentra los valores del vector r, del vector p y del término independiente k según la siguiente expresión:

$$X(z) = \frac{r_1}{z - p_1} + \frac{r_2}{z - p_2} + \ldots\ldots + \frac{r_n}{z - p_n} + k$$

Usando Matlab:

num = [5 26 44 29];

den = [1 6 11   6];

[r, p, k] = residue(num,den)

% Los resultados son:

r = [-2 -5 3], p = [-3 -2 -1], k = 5,

por tanto, las fracciones parciales quedan :

$$X(z) = \frac{-2}{z+3} + \frac{-5}{z+2} + \ldots\ldots + \frac{3}{z+1} + 5$$

Multiplicando por z^{-1} :

$$X(z) = \frac{-2z^{-1}}{1+3z^{-1}} + \frac{-5z^{-1}}{1+2z^{-1}} + \ldots\ldots + \frac{3z^{-1}}{1+z^{-1}} + 5$$

Si se supone que:

$v(k) = a^k$

$Z[a^{k-1}] = Z[v(k-1)] = z^{-1} Z[v(k)] = z^{-1} Z[a^k] = z^{-1} * 1/(1 - a z^{-1})$

entonces ,

$x(k) = -2(-3)^{k-1} - 5(-2)^{k-1} + 3(-1)^{k-1} + 5\delta_k$

donde δ_k es el delta kronecker (delta de Dirac en control continuo) cuya transformada Z es igual a 1.

1.4.3 MÉTODO DE LOS RESIDUOS

El método plantea que:

$x(kT) \sum_{i=1}^{m} =(\text{residuos de } X(z) z^{k-1} \text{ en el polo } z = z_i)$

$x(kT) = k_1 + k_2 + \cdots\cdots + k_m$

(a) Si $X(z) z^{k-1}$ tiene un polo simple en $z = z_i$, entonces,

el residuo es :

$$k_i = \lim_{z \to z_i} \left[(z - z_i) X(z) z^{k-1} \right]$$

(b) Si $X(z)\, z^{k-1}$ tiene un polo múltiple en $z = z_i$ de orden q, entonces,

el residuo es :

$$k_i = \frac{1}{(q-1)!} \cdot \lim_{z \to z_i} \frac{\partial^{q-1}}{\partial z^{q-1}} \left[(z - z_i)^q X(z) z^{k-1} \right]$$

EJEMPLO 1-9

$$X(z) = \frac{3z^2 - 9z}{z^3 + 1.8z^2 + 1.05z - 0.2}$$

```
% EJEMPLO 1-9 : PROGRAMA EN MATLAB
num = [0 3 -9 0];              % coeficientes del numerador
den = [1 -1.8 1.05 -0.2];      % coeficientes del denominador
Xz = tf (num, den, -1);        % función de transferencia en términos de z
pole(Xz)                       % obtención de polos
% la función tiene un polo simple en z = 0.8 y un polo doble en z = 0.5
% se debe reconstruir la función para aplicar el comando limit de Matlab
Xz1 = sum(num.*[z^3 z^2 z 1])/sum(den.*[z^3 z^2 z 1]);
 % a) cálculo del residuo para el polo simple
syms z k
```

*k1 =limit ((z-0.8)*Xz1*z ^(k-1),z,0.8)*

*% k1 = -220/3*4^ k/(5^ k)*

% b) cálculo del residuo para el polo doble

*h=diff((z-0.5)^2*Xz1*z^(k-1),2) % segunda derivada*

k2=limit(h,z,0.5)

% k2 = 20/9(87*k+220+45*k^2)/(2^k)*

*x(k) = k1 + k2 = -220/3*4^ k/(5^ k) + 20/9*(87*k + 220+45*k ^ 2)/(2^ k).*

1.5 ECUACIONES EN DIFERENCIA

Considérese un sistema discreto LTI (Lineal e invariante en el tiempo) dado por la ecuación en diferencias:

$x(k) + a_1\, x(k-1) + a_2\, x(k-2) + \cdots\cdots\cdots +a_n\, x(k-n) =$

$b_0\, u(k) + b_1\, u(k-1) + b_2\, u(k-2) + \cdots\cdots\cdots +b_n\, u(k-n)$

donde u(k) es la entrada al sistema y x(k) es la salida.

La forma de solucionar esta ecuación en diferencia consiste en calcular la transformada Z, luego aplicar las condiciones iniciales dadas y por último obtener la transformada Z inversa. Se debe recordar que:

$Z[x(k-n)T] = z^{-n}\, Z[x(k)] = z^{-n}\, X(z),$ *y,*

$$Z[x(k+n)T] = z^{n} \left[X(z) - \sum_{k=0}^{n-1} x(kT)*z^{-k} \right] =$$

$$z^{n}\, X(z) - z^{n}\, x(0) - z^{n-1}\, x(1) - z^{n-2}\, x(2) - \cdots\cdots - z\, x(n-1)$$

EJEMPLO 1-10

Resolver la siguiente ecuación en diferencias.

$x(k+2) + 5x(k+1) + 6x(k) = 0$

Condiciones iniciales: x(0) = 0, x(1) = 1

Solución :

a) Aplicando Transf._Z, se tiene:

$[z^2 \; X(z) - z^2 x(0) - z \; x(1)] + 5[z \; X(z) - z\,x(0)] + 6\,X(z) = 0$

b) Sustituyendo condiciones iniciales:

$[z^2 \; X(z) - z^2 (0) - z\,(1)] + 5[z \; X(z) - z\,(0)] + 6\,X(z) = 0$

$= z^2 \; X(z) \; - z \; + 5z \; X(z) \; + 6\,X(z) = 0$

despejando X(z):

$$X(z) = \frac{z}{z^2 + 5z + 6} = \frac{z}{z+2} - \frac{z}{z+3} \qquad \textit{(fracciones parciales)}$$

c) Obtener la transf_z inversa :

Sabiendo que, $z[\,a^k\,] = 1/\,(1 - a\,z^{-1})$***, entonces ,***

$$X(z) = \frac{1}{1+2z^{-1}} - \frac{1}{1+3z^{-1}}$$

Por tanto su Transf._Z inversa es :

$x(k) = (-2)^k - (-3)^k$

EJEMPLO 1-11

Resolver la siguiente ecuación en diferencias.

$x(k) + 5x(k-1) + 6x(k-2) = u(k)$, ***donde u(k) es el escalón unitario***

Solución :

a) Aplicando Transf._Z, se tiene:

$X(z) + 5\,z^{-1} X(z) \; + 6\,z^{-2} X(z) = U(z)$, ***pero*** $U(z) = 1\,/\,(1 - z^{-1})$

Despejando X(z) :

$$X(z) = \frac{1}{(1-z^{-1})(1+5z^{-1}+6z^{-2})} = \frac{z^3}{(z-1)(z^2+5z+6)}$$

b) Obtener la transf_z inversa:

% Aplicando Matlab para encontrar las fracciones parciales

Xz = zpk([0 0 0], [1 -2 -3], 1, -1);

Xz = tf(Xz)

pole(Xz)

% tiene polos simples en : -3.0000 -2.0000 1.0000

[num,den]=tfdata(Xz,'v')

[r,p,k] = residue(num,den)

% r = -6.7500 2.6667 0.0833

% p = -3.0000 -2.0000 1.0000

% k = 1

Con base en lo anterior,

$$X(z) = \frac{z+6.75}{z+3} + \frac{z-2.6667}{z+2} + \frac{z-0.0833}{z-1} + 1$$

Multiplicando por z^{-1} :

$$X(z) = \frac{1+6.75z^{-1}}{1+3z^{-1}} + \frac{1-2.6667z^{-1}}{1+2z^{-1}} + \frac{1-0.0833z^{-1}}{1-z^{-1}} + 1$$

Sabiendo que, $z[a^k] = 1/(1 - a\,z^{-1})$, entonces ,

$$x(k) = (-3)^k + 6.75(-3)^{k-1} + (-2)^k - 2.6667(-2)^{k-1} + (1)^k - 0.0833(1)^{k-1} + \delta_k$$

$= (-3)^k + 6.75(-1/3)(-3)^k + (-2)^k – 2.6667(-1/2) (-2)^k + 1 - 0.0833 + \delta_k$

$x(k) = -5/4*(-3)^k + 7/3*(-2)^k + 0.9167 + \delta_k$

EJEMPLO 1-12

Resolver la siguiente ecuación en diferencias.

$x (k + 2) + 0.5x (k + 1) + 0.2x(k) = u(k + 1) + 0.3u(k),$ **(1)**

Condiciones iniciales:

$x(k) = 0$, para $k \le 0$ y $u(k) = 0$, para $k < 0$ y además,

$u(0) = 1.5, u(1) = 0.5, u(2) = -0.5, u(k) =0$ para $k = 3, 4, 5, \cdots\cdots$

Solución :

a) Aplicando Transf._Z, se tiene:

$[z^2 X(z) – z^2 x(0) – z x(1)] + 0.5[z X(z) – z x(0)] + 0.2 X(z)$

$= [z U(z) –z u(0)] + 0.3 U(z)$ **(2)**

b) Sustituyendo condiciones iniciales:

Como no se conoce x(1), se debe encontrar reemplazando k por -1 en la ecuación (1),

$x(-1+2) + 0.5x(-1+1) + 0.2x(-1) = u(-1+1) + 0.3u(-1),$ entonces,

$x(1) + 0.5 x(0) + 0.2 x(-1) = u(0) + 0.3 u(-1),$

de las condiciones iniciales, $x(0)=0, x(-1)= 0, u(0) = 1.5, u(-1) = 0,$

reemplazando se tiene que,

$x(1) = 1.5$

encontrado x(1) se sustituyen condiciones iniciales en (2),

$[z^2 X(z) – z (1.5)] + 0.5[z X(z)] + 0.2 X(z) = [z U(z) - z (1.5)] + 0.3 U(z)$ **(2)**

Despejando X(z):

$$X(z) = \frac{z + 0.3}{z^2 + 0..5\, z + 0.2}\, U(z),$$

$$U(z) = \sum_{k=0}^{\infty} u(k)\, z^{-k} = u(0) + u(1)\, z^{-1} + u(2)\, z^{-2} + u(3)\, z^{-3} + \cdots\cdots +$$

$= 1.5 + 0.5\, z^{-1} - 0.5\, z^{-2}$, reemplazando,

$$X(z) = \frac{(z + 0.3)\,(1.5\, z^{2} + 0.5\, z - 0.5)}{z^{2}\,(z^2 + 0.5\, z + 0.2)} = \frac{1.5\, z^{3} + 0.95\, z^{2} - 0.35\, z - 0.15}{z^4 + 0.5\, z^3 + 0.2\, z^2}$$

Para obtener la señal discreta por Matlab:

```
% OBTENER DIEZ VALORES DE LA SEÑAL
num = [0 1.5 0.95 -0.35 -0.15];
den = [1 0.5 0.2 0 0];
u = [1 zeros(1,10)];
xk = filter(num,den,u)
k = 0:10;
stem(xk, k)          % grafica la señal muestreada
```

Valores x(k) dados por Matlab:

x(0)= 0, x(1)= 1.5000, x(2)= 0.2000, x(3)=-0.7500, x(4)= 0.1850, x(5)= 0.0575

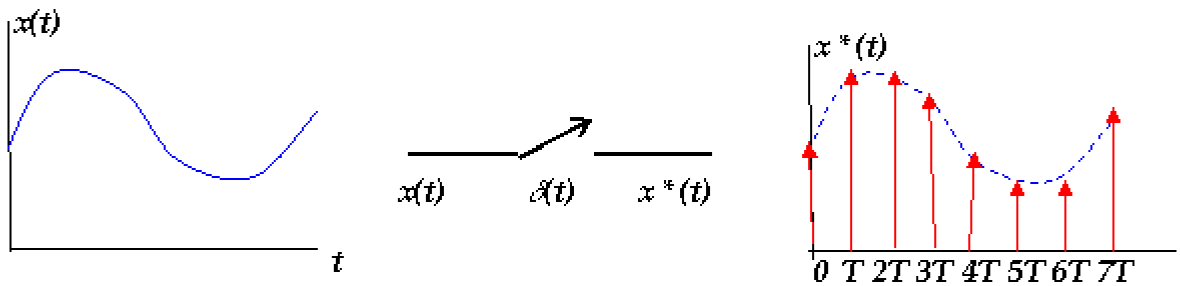
x(6)= -0.0658, x(7)= 0.0214, x(8)= 0.0025, x(9)= -0.0055, x(10)= 0.0023

2 FUNCIÓN DE TRANSFERENCIA

2.1 MUESTREO Y RETENCIÓN

2.1.1 MUESTREO DE UNA SEÑAL

Una señal continua se puede convertir a una señal discreta mediante un muestreador que convierte la señal en un tren de impulsos a una rata o periodo de muestreo T y magnitud de cada impulso igual al valor uestreado de la señal continua en el instante de muestreo.



Señal continua **Muestreador** **Señal**
muestreada

El tren de impulsos unitarios se puede representar mediante la ecuación:

$$\delta_T(T) = \sum_{k=0}^{\infty} \delta(t-kT) = x(0)\,\delta(T) + x(T)\,\delta(t-T) + x(2T)\,\delta(t-2T) + \cdots\cdots$$

La señal muestreada se puede representar como la magnitud de la señal en el instante muestreado multiplicada por el tren de impulsos de la siguiente manera

$$x^*(t) = \sum_{k=0}^{\infty} x(kT)\,\delta(t-kT) = x(0)\,\delta(T) + x(T)\,\delta(t-T) + x(2T)\,\delta(t-2T) + \cdots\cdots$$

Sacando la Transformada de Laplace a la señal muestreada :

$$X^*(s) = L[x^*(t)] = x(0)L[\delta(t)] + x(T)L[\delta(t-T)] + x(2T)\,L[\delta[t-2T)] + \cdots\cdots$$
$$= x(0) + x(T)e^{-Ts} + x(2T)e^{-2Ts} + \cdots\cdots$$

$$X^*(s) = \sum_{k=0}^{\infty} x(kT)\, e^{-kTs}$$

Si $z = e^{Ts}$, esto es, $s = (1/T)\ \ln(z)$, entonces,

$$X^*(s) = \sum_{k=0}^{\infty} x(kT)\, z^{-k} = X(z)$$

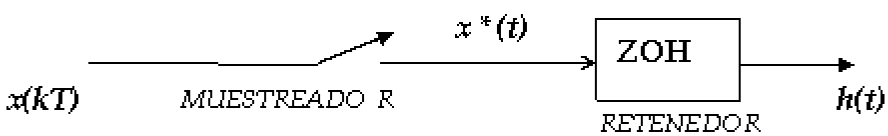
<u>Conclusión :</u> La transformada de Laplace de una señal muestreada es la misma transformada Z si : $z = e^{Ts}$

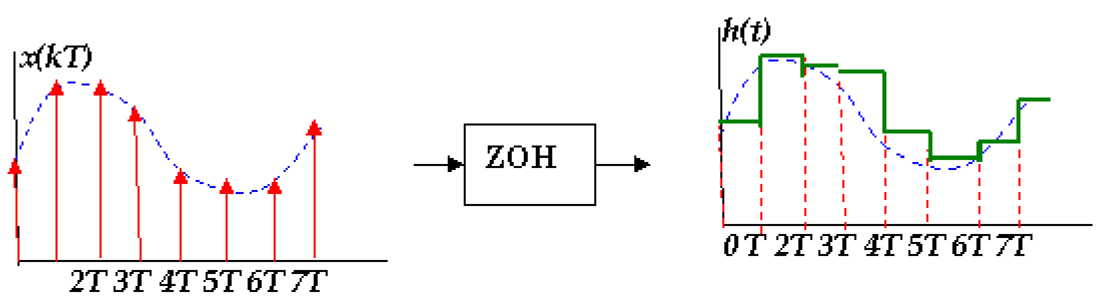
2.1.2 RETENCIÓN DE DATOS

Es el proceso de recuperación de la señal continua a partir de la señal discreta. El retenedor utiliza las muestras anteriores para extrapolar la señal continua entre el instante de muestreo presente y el siguiente.

El retenedor más utilizado es el retenedor de orden cero ZOH (zero order hold). Este retenedor genera una señal continua h(t) manteniendo o reteniendo cada valor de la muestra cada periodo de muestreo. Esto es:

$$h(kT+t) = x(kT), \quad \text{para } kT \le t \le (k+1)T$$





h(t) = x(0)u(t) + [x(T)-x(0)] u(t-T) + [x(2T)-x(T)] u(t-2T) + ·············

h(t) = x(0)[u(t)-u(t-T)] + x(T)[u(t-T)-u(t-2T)] + x(2T)[u(t-2T)-u(t-3T)] + ··········

$$h(t) = \sum_{k=0}^{\infty} x(kT)\,[u(t-kT) - u(t-(k+1)T)],$$ *aplicando transf_Laplace,*

$$H(s) = \sum_{k=0}^{\infty} x(kT)\left[\frac{e^{-kTs}}{s} - \frac{e^{-(k+1)Ts}}{s}\right] = \frac{1-e^{-Ts}}{s}\sum_{k=0}^{\infty} x(kT)e^{-kTs}$$

Como, H(s) = G_{ZOH}(s) X*(s), se tiene que la Función de transferencia del retenedor de orden cero es:

$$\boxed{G_{ZOH}(s) = \frac{1-e^{-Ts}}{s}}$$

Existe y se aplica también el retenedor de orden primero FOH (First Order Hold) cuya interpolación entre periodos de muestreo se hace en forma triangular. Para este retenedor,

$$h(kT+t) = x(kT) + \frac{t - kT}{T}\,[\,x(k+1)T - x(kT)] \quad , \quad \text{para } kT \text{ £ } t \text{ £ } (k+1)T$$

2.2 FUNCIONES DE TRANSFERENCIA

2.2.1 SISTEMA EN LAZO ABIERTO

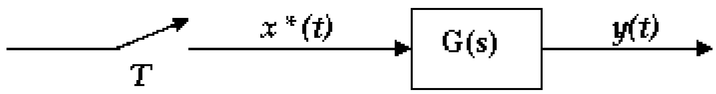
La función de transferencia relaciona la salida de un sistema en los instantes de muestreo con la correspondiente entrada muestreada.

$$X(z) \longrightarrow \boxed{G(z)} \longrightarrow Y(z)$$

Y(z) = G(z) X(z) , entonces,

$$G(z) = \frac{Y(z)}{X(z)}$$

Demostración :



Y(s) = G(s) X(s), entonces,*

Y*(s) = G*(s)X*(s), por tanto,

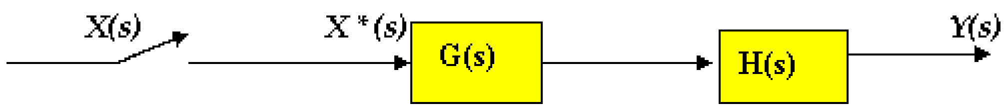
Sacando Transf_Z a la seña muestreada :

Y(z) = G(z) X(z), o sea que:

$$\boxed{G(z) = \frac{Y(z)}{X(z)}}$$

2.2.2 SISTEMAS EN CASCADA

(a) CON UN MUESTREADOR



Y(s) = G(s)H(s)X*(s), discretizando la ecuación,

Y*(s) = [G(s)H(s)]*X*(s), si se simboliza [G(s)H(s)]* =[GH(s)]*

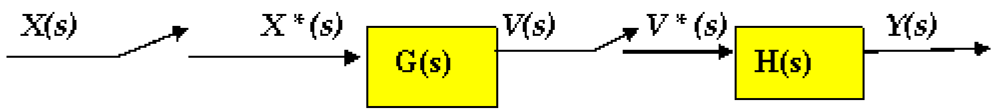
Y*(s) = [GH(s)]*X*(s), sacando transformada z :

Y(z) = [GH(z)]X(z), donde, GH(z) à z[GH(s)] = z[G(s)H(s)]

Por tanto, la función de transferencia para este sistema es igual a :

$$\boxed{\frac{Y(z)}{X(z)} = GH(z)}$$

(b) CON DOS MUESTREADORES SINCRONIZADOS



Y(s) = H(s)V*(s), y V(s) = G(s)X*(s)

discretizando las ecuaciónes,

Y*(s) = H*(s)V*(s), o sea que , Y(z) = H(z)V(z) (1)

V*(s) = G*(s)X*(s), o sea que , V(z) = G(z)X(z)

Reemplazando V(z) en (1), se tiene :

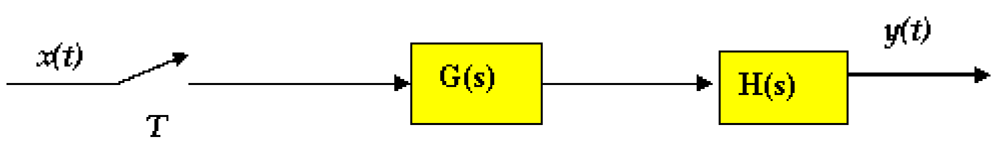
Y(z) = H(z) G(z)X(z)

Por tanto la función de transferencia del sistema es:

$$\frac{Y(z)}{X(z)} = G(z)H(z) \neq GH(z)$$

EJEMPLO 2-1

(a) Obtener la función de transferencia de (T = 1.0) :



$$G(s) = \frac{1}{s+2} \quad H(s) = \frac{1}{s+5}$$

$$F(s) = G(s)H(s) = \frac{1}{s+2} \cdot \frac{1}{s+5} = \frac{1/3}{s+2} - \frac{1/3}{s+5}$$

Por tanto su transformada inversa de Laplace es igual a :

f(t) = L^{-1} [F(s)] = (1/3) e^{-2t} – (1/3) e^{-5t}

La cual corresponde a una función discreta a:

f(kT) = (1/3) e^{-2kT} – (1/3) e^{-5kT},

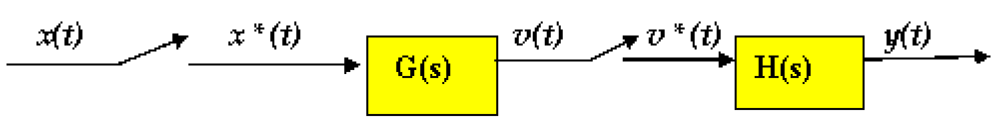
que tiene una transformada Z igual a :

$$F(z) = \frac{Y(z)}{X(z)} = GH(z) = \frac{1/3}{1 - e^{-2T} z^{-1}} - \frac{1/3}{1 - e^{-5T} z^{-1}}$$

Para un periodo de muestreo T = 1, se tiene:

$$F(z) = \frac{Y(z)}{X(z)} = GH(z) = \frac{0.3333}{1 - 0.1353 z^{-1}} - \frac{0.3333}{1 - 0.0067 z^{-1}} = \frac{0.0429 z^{-1}}{\left(1 - 0.1353 z^{-1}\right)\left(1 - 0.0067 z^{-1}\right)}$$

(b) Obtener la función de transferencia de :



$$G(s) = \frac{1}{s+2} \Rightarrow g(t) = e^{-2t} \Rightarrow g(kT) = e^{-2kT} \Rightarrow G(z) = \frac{1}{1 - e^{-2T} z^{-1}}$$

$$H(s) = \frac{1}{s+5} \Rightarrow g(t) = e^{-5t} \Rightarrow g(kT) = e^{-5kT} \Rightarrow G(z) = \frac{1}{1 - e^{-5T} z^{-1}}$$

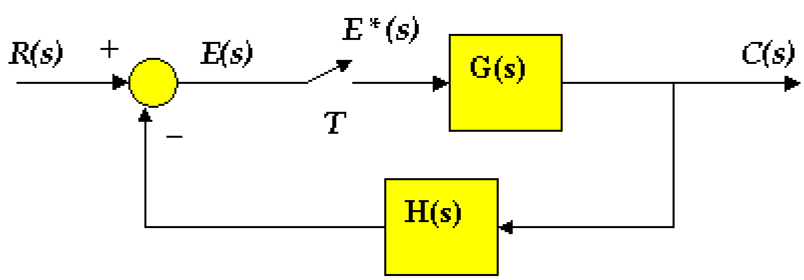
Entonces la función de transferencia del sistema para T = 1 es igual a:

$$F(z) = G(z)H(z) = \frac{1}{1-e^{-2T}z^{-1}} \cdot \frac{1}{1-e^{-5T}z^{-1}} = \frac{1}{(1-0.1353z^{-1})(1-0.0067z^{-1})}$$

Se comprueba que G(z)H(z) ≠ GH(z)

2.2.3 SISTEMA EN LAZO CERRADO

(a) CON UN MUESTREADOR



E(s) : es el error

E(s) = R(s) – H(s)C(s) , como , C(s) = G(s) E*(s), entonces,

E(s) = R(s) – G(s)H(s)E*(s), tomando señales muestreadas,

E*(s) = R*(s) – [G(s)H(s)]* E*(s), despejando E*(s),

$$E^*(s) = \frac{R^*(s)}{1+[G(s)H(s)]^*}$$

Como C*(s) = G*(s)E*(s), entonces,

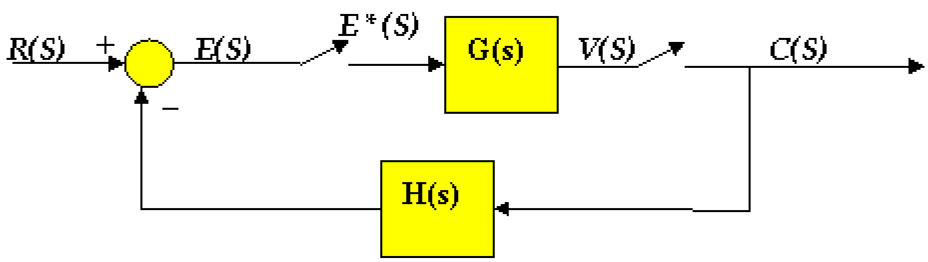
$$C^*(s) = \frac{G^*(s)R^*(s)}{1+[G(s)H(s)]^*}$$

Tomando transf_Z, se tiene :

$$C(z) = \frac{G(z)\,R(z)}{1+[GH(z)]}$$ *, por tanto su función de transferencia es :*

$$\boxed{F(z) = \frac{C(z)}{R(z)} = \frac{G(z)}{1+[GH(z)]}}$$

(b) CON DOS MUESTREADORES SINCRONIZADOS



E(s) = R(s) – H(s)C(s) , como , C(s) =V(s) y V(s) = G(s) E*(s), entonces,*

C(s) = G*(s)E*(s)

E(s) = R(s) – H(s)G*(s)E*(s), tomando señales muestreadas,

E*(s) = R*(s) – H*(s)G*(s) E*(s), despejando E*(s),

$$E^*(s) = \frac{R^*(s)}{1+[G^*(s)H^*(s)]}$$

Como *C*(s) = G*(s)E*(s)****, entonces,***

$$C^*(s) = \frac{G^*(s)\,R^*(s)}{1+[G^*(s)H^*(s)]}$$

Tomando transf_Z, se tiene :

$$C(z) = \frac{G(z)\,R(z)}{1+G(z)H(z)}$$

, *por tanto su función de transferencia es* :

$$\boxed{F(z)=\frac{C(z)}{R(z)}=\frac{G(z)}{1+G(z)H(z)}}$$

EJEMPLO 2-2

Calcular la función de transferencia de un sistema en lazo cerrado con un muestreador que tiene,

$$G(s)=\frac{1}{s+2} \qquad \textit{y} \qquad H(s)=\frac{2}{s}$$

$$F(z)=\frac{C(z)}{R(z)}=\frac{G(z)}{1+\left[GH(z)\right]}$$ ***donde, GH(z) = z[G(s)H(s)]***

$$G(s)H(s)=\left[\frac{2}{s(s+2)}\right]=\frac{1}{s}-\frac{1}{s+2}\Rightarrow GH(z)=\frac{1}{1-z^{-1}}-\frac{1}{1-e^{-2T}z^{-1}}$$

Para T = 0.1 seg,

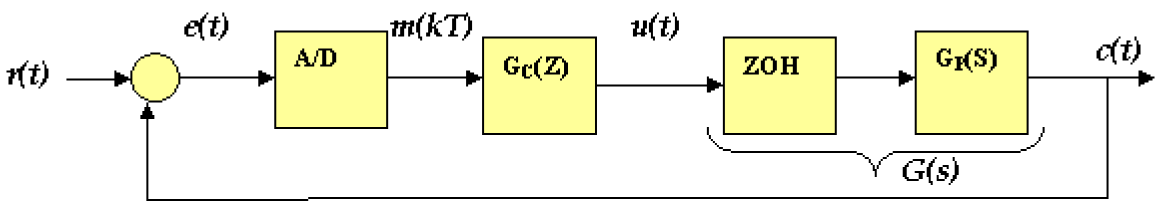
$$GH(z)=\frac{z}{z-1}-\frac{z}{z-e^{-0.2}}=\frac{0.1813z}{(z-1)(z-0.8187)}=\frac{0.1813z}{z^2-0.1813z+0.8187}$$

$$G(z)=\frac{1}{1-e^{-2T}z^{-1}}=\frac{z}{z-0.8187}$$

$$F(z)=\frac{z/(z-0.8187)}{1+0.1813z/((z-1)(z-0.8187))}=\frac{z(z-1)}{(z-1)(z-0.8187)+0.1813z}$$

$$F(z)=\frac{z^2-z}{z^2-0.1813z+0.8187+0.1813z}=\frac{z^2-z}{z^2+0.8187}$$

2.2.4 SISTEMA DE CONTROL DIGITAL



En la figura A/D es el muestreador, Gc(z) es el controlador digital, ZOH es el (convertidor D/A) y Gp(s) es la función de transferencia de la planta.

e(t) es la señal de error que se muestrea, e(kT) es la señal discreta del error, m(kT) es la salida del controlador digital que se obtiene al resolver la ecuación en diferencias de la función de transferencia del controlador, y u(t) es la señal de control que se debe aplicar a la planta.

G(s) = Gzoh(s) Gp(s) =

$$\frac{1-e^{-Ts}}{s}\cdot Gp(s) = (1-e^{-Ts})\frac{Gp(s)}{s}$$

$$G(z) = (1-z^{-1}) * Z\left[\frac{Gp(s)}{s}\right]$$

Del diagrama en bloques se tiene que,

C(z) = [Gc(z)G(z)] E(z) y que E(z) = R(z) – C(z), entonces,

C(z) = [Gc(z)G(z)] [R(z) – C(z)] = Gc(z)G(z)R(z) – Gc(z)G(z)C(z)

Luego entonces la función de transferencia es igual a :

$$\boxed{F(z) = \frac{C(z)}{R(z)} = \frac{Gc(z)G(z)}{1+Gc(z)G(z)}}$$

Como la función de transferencia en lazo abierto de este sistema es :

$F_{lazo\text{-}abierto}(z) = Gc(z)G(z),$

entonces, cuando la realimentación es igual a 1 :

$$\boxed{F_{lazo-cerrado} = \frac{F_{lazo-abierto}(z)}{1 + F_{lazo-abierto}(z)}}$$

2.2.5 CONTROLADOR DIGITAL PID

Un controlador Proporcional-Integral-Derivativo continuo o analógico tiene como respuesta de salida la siguiente ecuación:

$$m(t) = K\left[e(t) + \frac{1}{Ti}\int_0^t e(t)\,dt + Td\frac{de(t)}{dt}\right] \qquad (1)$$

Donde: K = Ganancia proporcional

Ti = Tiempo integral o de ajuste

Td = tiempo derivativo o de adelanto

Discretizando la anterior ecuación, se deben reemplazar los términos continuos a discretos como se indica a continuación :

m(t) → m(kT),

e(t) → e(kT),

$$\frac{1}{Ti}\int_0^t e(t)\,dt \rightarrow \frac{T}{Ti}\sum_{h=1}^{k}\frac{e(h-1)T + e(hT)}{2}$$

$$Td\frac{de(t)}{dt} \rightarrow Td\frac{e(kT) - e(k-1)T}{T}$$

Que quedará de la siguiente forma:

$$m(kT) = K\left[e(kt) + \frac{T}{Ti}\sum_{h=1}^{k}\frac{e(h-1)T + e(hT)}{2} + \frac{Td}{T}\left[e(kT) - e(k-1)T\right]\right] \qquad (2)$$

Ahora se tiene que encontrar la Transf._Z M(z). De la anterior ecuación:

La Transf._Z de la sumatoria se obtiene aplicando la propiedad de la suma de funciones , si :

$$x(hT) = \frac{e(h-1)T + e(hT)}{2} \Rightarrow X(z) = \frac{z^{-1}E(z) + E(z)}{2}$$

$$y(kT) = \sum_{h=1}^{k} x(hT) \Rightarrow Y(z) = \frac{1}{1-z^{-1}} X(z) = \frac{(z^{-1}+1)}{2(1-z^{-1})} E(z)$$

Reemplazando en (2) :

$$\frac{T}{Ti} \cdot \frac{1+z^{-1}}{2(1+z^{-1})} = \frac{T}{Ti}\left[-\frac{1}{2} + \frac{1}{1-z^{-1}}\right] = -\frac{T}{2Ti} + \frac{T}{Ti} \cdot \frac{1}{1-z^{-1}}$$

$$M(z) = K\left[E(z) + \frac{T}{Ti} \cdot \frac{1+z^{-1}}{2(1+z^{-1})} E(z) + \frac{Td}{T}(1-z^{-1})E(z)\right]$$

$$M(z) = K\left[1 - \frac{T}{2Ti} + \frac{T}{Ti} \cdot \frac{1}{1-z^{-1})} + \frac{Td}{T}(1-z^{-1})\right]E(z)$$

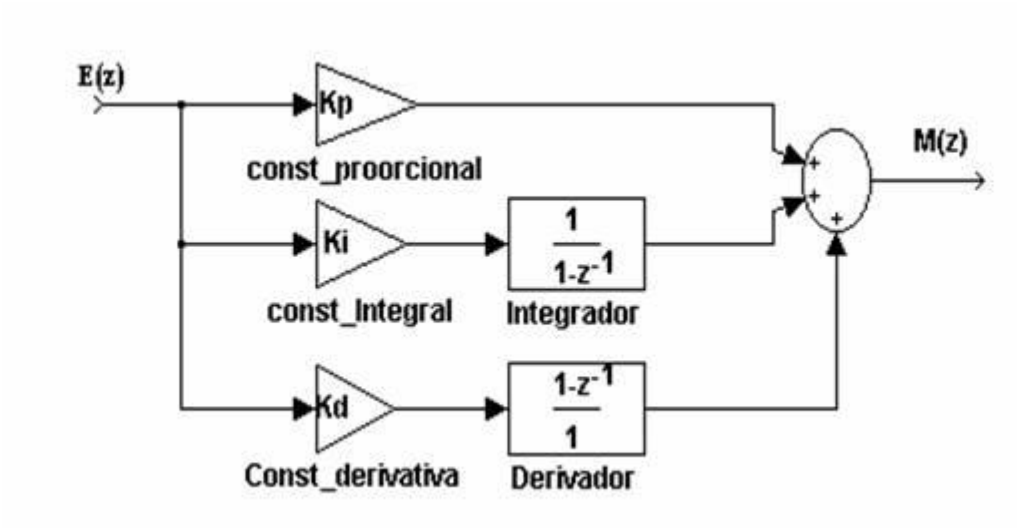
o sea que :

$$G_{PID}(z) = \frac{M(z)}{E(z)} = K\left[1 - \frac{T}{2Ti}\right] + K\left[\frac{T}{Ti} \cdot\right]\frac{1}{1-z^{-1}} + K\left[\frac{Td}{T}\right](1-z^{-1})$$

Por tanto la función de transferencia de un controlador PID discreto es :

$$\boxed{G_{PID}(z) = \frac{M(z)}{E(z)} = Kp + Ki\frac{1}{1-z^{-1}} + Kd(1-z^{-1})}$$

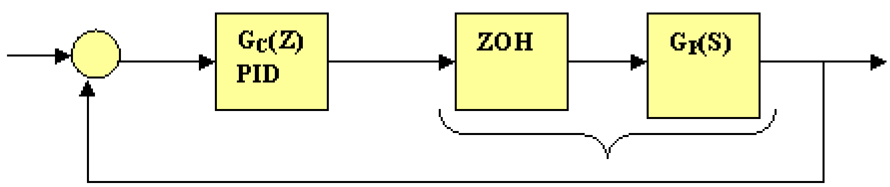
Cuyo diagrama en bloques es el siguiente:



EJEMPLO 2-3

Encontrar la función de transferencia en lazo abierto y lazo cerrado del sistema dado, si el controlador PID tiene como parámetros Kp=1, Ki = 0.2 y Kd = 0.5. La planta tiene una función de transferencia igual a :

$$Gp(s) = \frac{1}{s(s+2)}$$



Como vimos anteriormente, la función de transferencia de la planta junto con el retenedor es igual a:

$$G(z) = (1-z^{-1})\, \mathcal{Z}\left[\frac{Gp(s)}{s}\right]$$

$$Gp(s) = \frac{1}{s(s+2)} \Rightarrow G(z) = (1-z^{-1})\, \mathcal{Z}\left[\frac{1}{s^2(s+2)}\right]$$

Resolviendo en fracciones parciales,

$$\left[\frac{1}{s^2(s+2)}\right]=\frac{A}{s^2}+\frac{B}{s}+\frac{C}{s+2}\Rightarrow A(s+2)+Bs(s+2)+Cs^2=1\Rightarrow A=0.5, B=-0.25, C=0.25$$

$$\text{Si}\quad V(s)=\left[\frac{1}{s^2(s+2)}\right]=\frac{0.5}{s^2}-\frac{0.25}{s}+\frac{0.25}{s+2}\Rightarrow v(t)=0.5t-0.25+0.25e^{-2t}$$

$$v(kT)=0.5kT-0.25+0.25e^{-2kT}\big|_{T=1}=0.5k-0.25+0.25e^{-2k}$$

$$V(z)=\frac{0.5z^{-1}}{(1-z^{-1})^2}-\frac{0.25}{1-z^{-1}}+\frac{0.25}{1-e^{-2}z^{-1}}\Rightarrow G(z)=(1-z^{-1})\cdot\left[\frac{0.5z^{-1}}{(1-z^{-1})^2}-\frac{0.25}{1-z^{-1}}+\frac{0.25}{1-e^{-2}z^{-1}}\right]$$

La función de transferencia de la planta discretizada G(z) es :

$$G(z)=\frac{0.2838(z+0.5232)}{(z-1)(z-0.1353)}=\frac{0.2838z+0.1485}{z^2-1.1353z+0.1353}$$

La función de transferencia del controlador PID es :

$$G_{PID}(z)=Kp+Ki\frac{1}{1-z^{-1}}+Kd(1-z^{-1})=\frac{Kp+Ki+Kd-(Kp+2Kd)z^{-1}+Kdz^{-2}}{1-z^{-1}}$$

$$=1+\frac{0.2}{1-z^{-1}}+0.5(1-z^{-1})=\frac{1.7-2z^{-1}+0.5z^{-2}}{1-z^{-1}}=\frac{1.7z^2-2z+0.5}{z-1}$$

(a) Función de transferencia en lazo abierto :

$$F_{lazo-abierto}=\frac{C(z)}{E(z)}$$

$$F_{lazo-abierto}=G_{PID}(z)G(z)=\left[\frac{1.7z^2-2z+0.5}{z-1}\right]\cdot\left[\frac{0.2838z+0.1485}{z^2-1.1353z+0.1353}\right]$$

$$F_{lazo-abierto}=\frac{0.4825z^3-0.3152z+0.07425}{z^3-2.135z^2+1.271z-0.1353}$$

(b) Función de transferencia en lazo cerrado :

$$F_{\text{lazo-cerrado}} = \frac{F_{\text{lazo-abierto}}}{1 + F_{\text{lazo-abierto}}} = \frac{0.4825z^3 - 0.3152z^2 - 0.1551z + 0.07425}{1.483z^3 - 2.451z^2 + 1.116z - 0.06109}$$

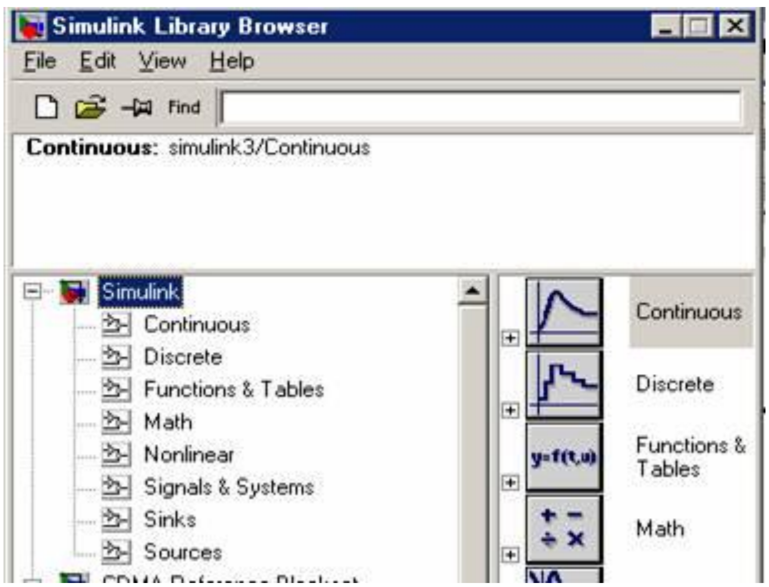
PROGRAMA EN MATLAB

```
% OBTENER FUNCION DE TRANSFERENCIA

% tiempo de muestreo: T = 1 seg

T = 1;

% parámetros del controlador PID

Kp=1; Ki = 0.2; Kd = 0.5;

% Función de transferencia de la planta

Gps = zpk([ ], [0 -2], 1)

% Función de transferencia de planta + retenedor

Gz =c2d(Gps, T,'zoh')

Gz = tf (Gz)

% Función de transferencia del controlador PID

GPID =tf([ Kp+Ki+Kd  -(Kp +2*Kd)  Kd], [1 -1], T)

% FUNCION DE TRANSFERENCIA DEL SISTEMA EN LAZO ABIERTO

Flazo_abierto = GPID*Gz

% FUNCION DE TRANSFERENCIA DEL SISTEMA EN LAZO CERRADO

Flazo_cerrado =feedback(Flazo_abierto,1)
```

3. SIMULINK

3.1 INTRODUCCIÓN

Simulink es una extensión de Matlab utilizado en el modelamiento y simulación de sistemas. Para arrancar Simulink se puede hacer desde el prompt de Matlab digitando el comando >>Simulink o utilizando el icono . Se abre la ventana Simulink Library Browser como se indica abajo y se puede diagramar un nuevo modelo activando el botón New Model , o sea el icono o de File → New→Model

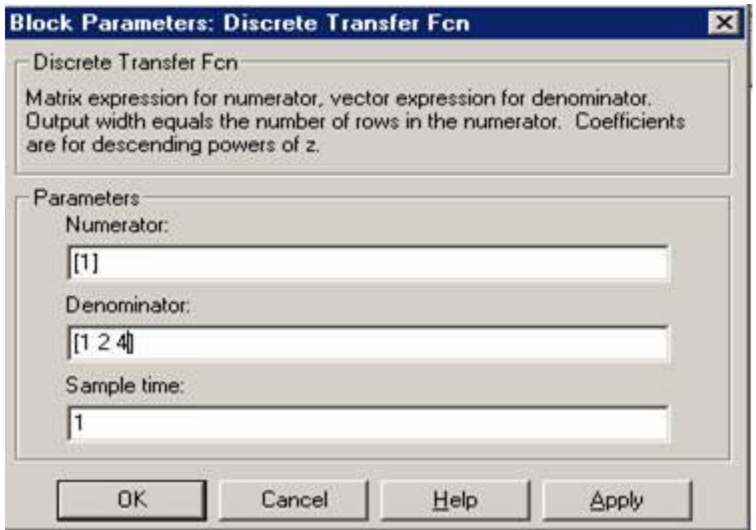


Un modelo es un conjunto de bloques que representa un sistema y como archivo tiene extensión *.mdl

3.2 ELEMENTOS BÁSICOS

Los elementos básicos son líneas y bloques. Los bloques están agrupados en: Sources, Links, Discrete, Continuos, Math, etc., tal como aparecen en la ventana anterior. Cada bloque tiene entradas y salida para realizar su interconexión. Por ejemplo, haga clic en Discrete y luego clic en Discrete Transfer Fcn y arrastre el bloque a la ventana en blanco. Si quiere modificar la función de transferencia del bloque haga doble clic en él y digite los

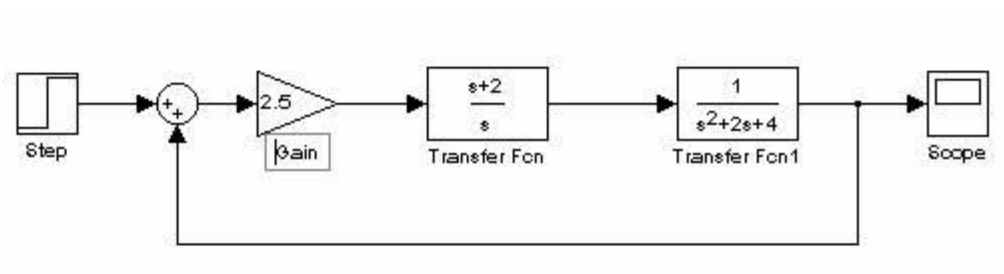
coeficientes del numerador y denominador en la nueva ventana que aparece. Para la función 1/(z^2 +2z +4) con tiempo de muestreo de 1 seg, quedaría:



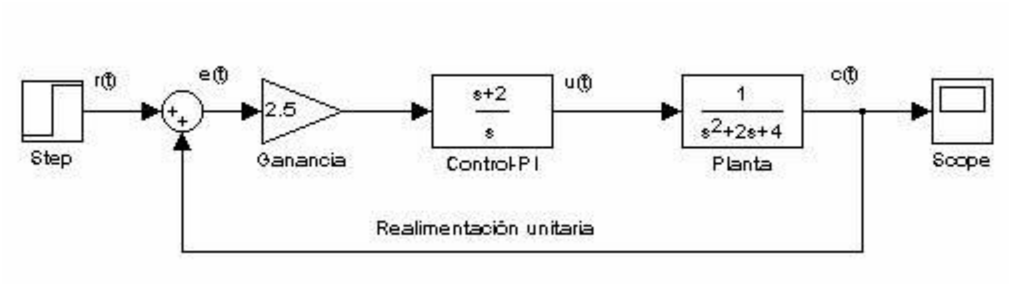
3.3 SISTEMAS DE CONTROL

Realizar el diagrama en bloques del siguiente sistema de control:

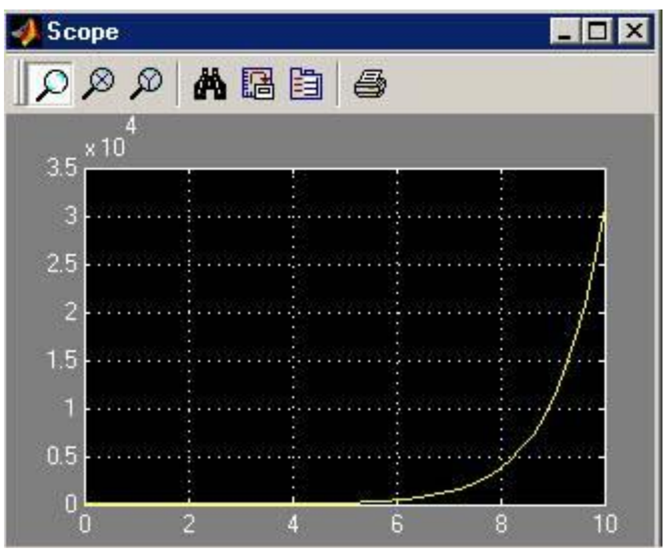
Lo primero es arrastrar los bloques a la página en blanco de forma que, Step es la función paso o escalón que se obtiene de Sources, Scope es el osciloscopio que se obtiene de Sinks, Transfer Fcn se obtiene de Continuos, Sum y Gain se obtienen de Math. Modifique los bloques dando doble clic sobre cada uno de ellos para cambiar sus parámetros o valores e interconéctelos.



Lo segundo es cambiar los nombres a los bloques y asignar las variables o señales haciendo doble clic en el lugar en que se van a colocar y salvar el modelo especificándole un nombre, por ejemplo ejem1.mdl



Por** **último** **se debe simular el sistema. Para ello se configura la señal de entrada, en este caso la función paso. Dar doble clic y asignar los siguientes →parámetros: Step time=0, Inicial value=0, Final value=1, Sample time=0. Para simular el sistema de control se escoge del menú Simulation Start o el icono  ***.y luego se hace doble clic en Scope para ver su respuesta o salida del sistema. Para observar además la entrada se puede colocar otro Scope a la salida de Step y se puede probar para varios pasos variando su amplitud, tiempo de inicio y tiempo de iniciación del paso. Para observar mejor la respuesta se usa el botón Autoscale (binoculares***  ***) de la ventana del Scope. Si quiere observar mejor la respuesta o parte de ella se pueden cambiar los parámetros de simulación, Simulation→ Simulation parameters. Por ejemplo cambiar el Start time y el Stop time y correr nuevamente la simulación.***

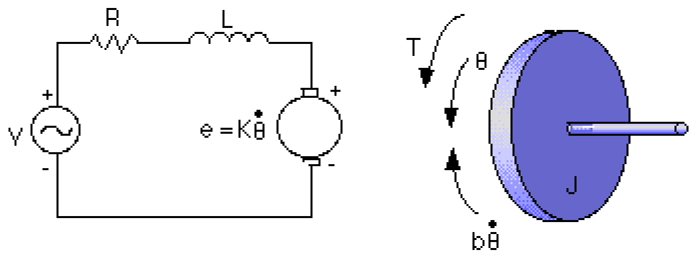


3.4 MODELANDO UN MOTOR DC

Un actuador común en sistemas de control es el motor DC. Provee directamente movimiento rotatorio y acoplado con poleas o correas puede proveer movimiento transnacional.

3.4.1 ECUACIONES DINÁMICAS

El circuito eléctrico de la armadura y el diagrama de cuerpo libre del rotor es mostrado en la figura con sus ecuaciones dinámicas.



(1) Leyes de Newton

$$J\frac{d^2\theta}{dt^2} = T - b\frac{d\theta}{dt} \Rightarrow \frac{d^2\theta}{dt^2} = \frac{1}{J}\left(k_t i - b\frac{d\theta}{dt}\right) \;, \quad \text{debido a que}, T = k_t i$$

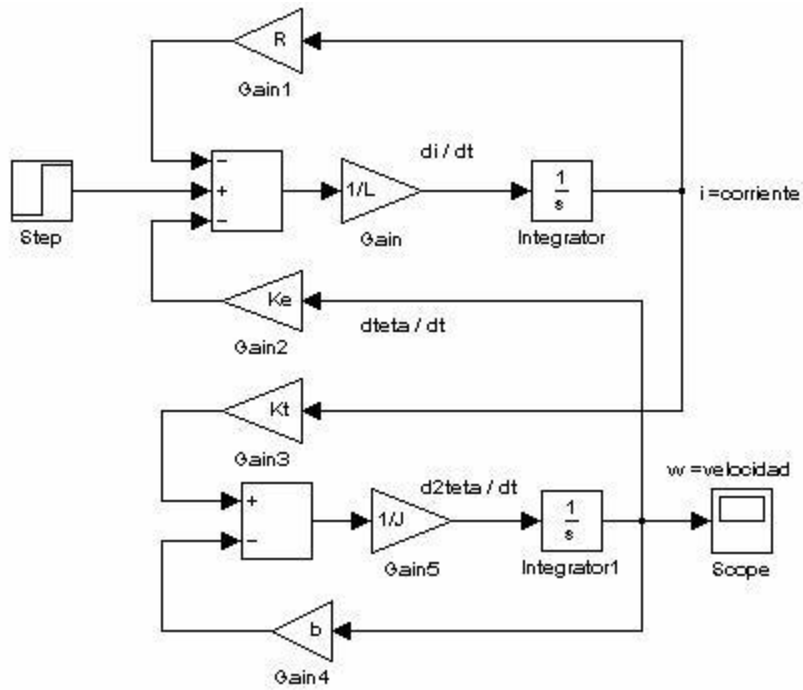
(2) Leyes de Kirchhoffs

$$L\frac{di}{dt} = -Ri + V - e \Rightarrow \frac{di}{dt} = \frac{1}{L}\left(-Ri + V - k_e\frac{d\theta}{dt}\right) \;, \quad \text{debido a que}, e = k_e\frac{d\theta}{dt}$$

Los parámetros físicos tienen los siguiente valores :

Momento de inercia del rotor : J = 0.01kg.m^2/sg^2
Rata de amortiguamiento del sistema mecánico: b = 0.1 N.m.sg
Constante de la fuerza electromotriz: Ke = Kt = 0.01 Nm/Amp
Resistencia eléctrica: R = 1 ohm
Inductancia eléctrica: L =0.5H
Fuente de voltaje de entrada: V
Posición angular: θ
Se asume que el rotor y el eje son rígidos

3.4.2 MODELADO DEL MOTOR EN VELOCIDAD



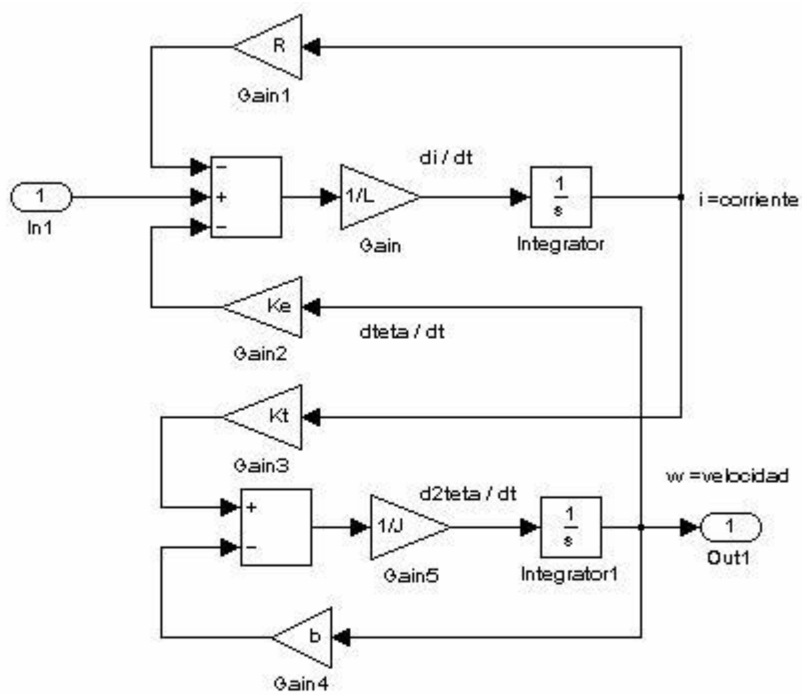
3.5 EXTRAER MODELO LINEAL

Para obtener la función de transferencia del motor primero se trasladan los parámetros del motor al modelo creando un archivo en Matlab (*.m) de la siguiente forma:

% VALORES DE LOS PARÁMETROS DEL MOTOR
J = 0.01;
b = 0.1;
Ke = 0.01;
Kt = 0.01;
R = 1;
L = 0.5;

Se ejecuta este archivo y se simula el modelo para una entrada de paso unitario de valor V = 0.01, con los siguientes parámetros de simulación: Stop time = 3 sg. Arranque la simulación y observe la salida (velocidad del motor).

Como segundo paso se debe obtener el modelo lineal de Matlab del motor. Para esto, borre el bloque Scope y cámbielo por Out obtenido de la librería de Signals&Systems. Haga lo mismo para Step cambiándolo por In de esta misma librería. Los bloques In y Out definen la entrada y salida del sistema que le gustaría extraer. Salve este modelo. El sistema quedará así:



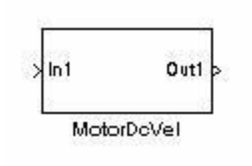
Como tercero y último paso, después de desarrollado el modelo y salvarlo por ejemplo con el nombre MotorDcVel.mdl se ejecutan los siguientes comandos:

% OBTENER EL MODELO LINEAL DEL SISTEMA
[num, den] = linmod('MotorDcVel')
Gps = tf(num, den)
La respuesta es :

$$Gp(s) = \frac{2}{s^2 + 12s + 20.02}$$

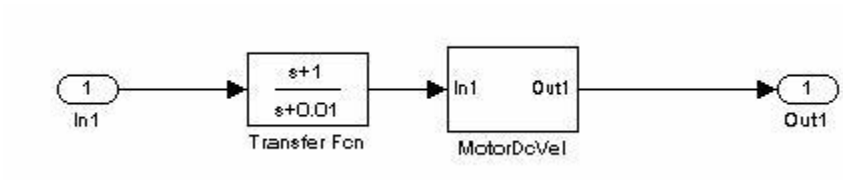
3.6 CREAR UN SUBSISTEMA

Abra una nueva ventana y arrastre de la librería Signals&Systems el bloque SubSystem , haga doble clic en este bloque, abra el modelo MotorDcVel.mdl (el que tiene In y Out como terminales) cópielo y péguelo en la nueva ventana de subsistema anterior. Cierre ventanas y aparece una nueva con el bloque con los terminales del subsistema creado. Déle el nombre MotorDcVel. Si a este bloque de subsistema se le da doble clic aparece el modelo completo diseñado anteriormente.



3.6.1 IMPLEMENTAR SISTEMA EN LAZO ABIERTO

Al subsistema creado que constituye la planta de un sistema de control se le va a adicionar un controlador y obtendremos la función de transferencia en lazo abierto y lazo cerrado.

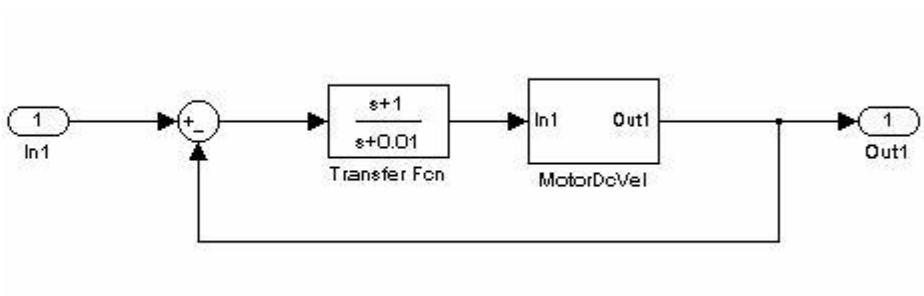


% CONTROL DE UN MOTOR DC
[num, den]=linmod('ControlMotor')
Glazo_abierto = tf(num, den)

Respuesta:

$$Glazo\_abierto = \frac{2s+2}{s^3+12.01s^2+20.14s+0.2002}$$

3.6.2 IMPLEMENTAR SISTEMA EN LAZO CERRADO



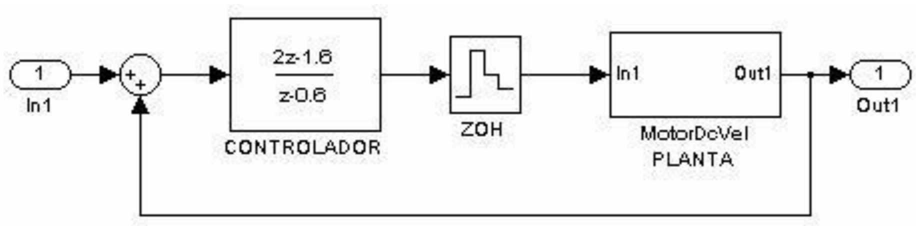
% CONTROL DE UN MOTOR DC
[num, den]=linmod('ControlMotor')
Glazo_cerrado= tf(num, den)

Respuesta:

$$Glazo\_cerrado = \frac{2s+2}{s^3+12.01s^2+22.14s+2.2}$$

3.7 SISTEMA DISCRETO

3.7.1 DIAGRAMA EN SIMULINK



3.7.2 PROGRAMA MATLAB

```
% SISTEMA DISCRETO DISCRETO
T=0.1;
[num,den]=dlinmod('MotorDigital',T)
Glazo_cerradoz=tf(num,den,T)
```

Respuesta:

$$Glazo\_cerrado(z) = \frac{0.01371z^2 - 0.001764z - 0.007364}{z^3 - 1.8z^2 + 1.015z - 0.1734}$$

4 ESTABILIDAD

4.1 INTRODUCCIÓN

Se analizará en este capítulo la estabilidad de un sistema de control discreto lineal e invariante en el tiempo de una entrada y una salida.

Recordemos que la relación entre variables de un sistema continuo a uno discreto se define por :

$$z = e^{Ts} = e^{T(\sigma + jw)}$$

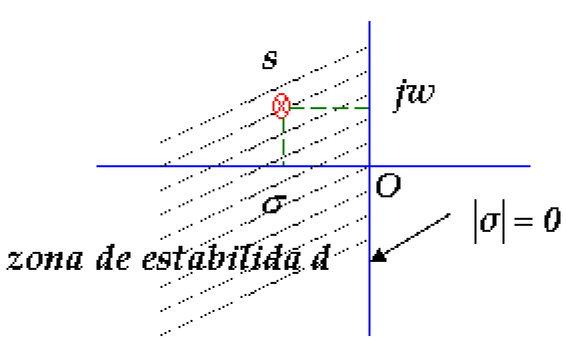
El límite de la estabilidad de un sistema continuo es el eje imaginario, o sea, en σ = 0, entonces,

$$z = e^{T(jw)} = coswT + jsenwT$$

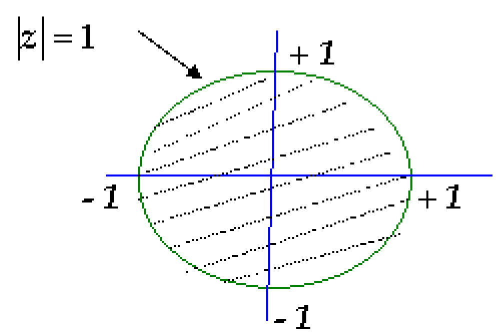
Por tanto, su magnitud es igual a :

$$|z| = cos^2 wT + sen^2 wT = 1$$

Se puede concluir que un sistema discreto es estable cuando sus polos están dentro el círculo unitario.



Sistema continuo

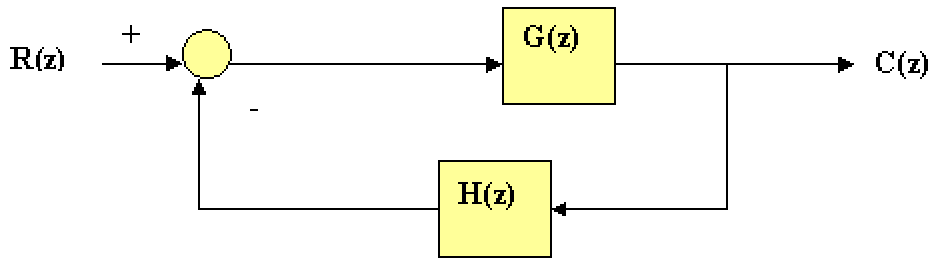


Sistema discreto

4.2 CRITERIOS DE ESTABILIDAD

4.2.1 ECUACIÓN CARACTERÍSTICA

El análisis de estabilidad de un sistema en lazo cerrado se hace de una manera rápida encontrando las raíces de la ecuación característica que a la vez son los polos del sistema en lazo cerrado. Para la figura :



Su función de transferencia es :

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1+G(z)H(z)}$$

Y su ecuación característica es :

F(z) = 1 + G(z) H(z) = 0

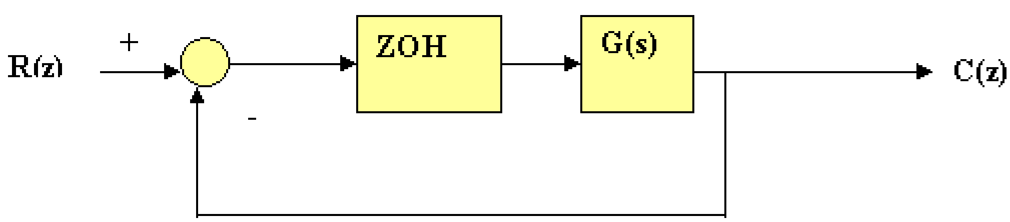
4.2.2 CRITERIOS DE ESTABILIDAD

La estabilidad puede determinarse por las localizaciones de los polos de la ecuación característica, de la siguiente manera :

1. ***Para que el sistema sea estable, los polos en lazo cerrado deben presentarse en el plano-z dentro del círculo unitario. Cualquier polo fuera de este círculo hace el sistema inestable.***
2. ***Si un polo simple se presenta en z =1 o si un par de polos complejos conjugados se presentan sobre el círculo unitario el sistema es críticamente estable. Cualquier polo múltiple sobre el círculo unitario hace inestable el sistema.***
3. ***Los ceros en lazo cerrado no afectan la estabilidad absoluta y por tanto pueden estar ubicados en cualquier parte del plano-z.***

EJEMPLO 4-1 :

Demostrar que el siguiente sistema con G(s) = 1/ (s(s +1)) es estable para un a) Tiempo de muestreo de 1 sg y b) No es estable si T = 10 sg.



Solución :

$$G(z) = (1 - z^{-1})\,\mathcal{Z}\left[\frac{G(s)}{s)}\right] \Rightarrow$$

$$G(z) = \frac{0.3679z + 0.2642z}{(z - 0.3679)(z - 1)}$$

$$G(z) = (1 - z^{-1})\,\mathcal{Z}\left[\frac{1}{s^2(s+1)}\right]$$

La función de transferencia en lazo cerrado es igual a :

$$G(z) = \frac{G(z)}{1 + G(z)} = \frac{0.36788(z + 0.7183)}{z^2 - z + 0.3621}$$

Su ecuación característica es :

$z^2 - z + 0.3621 = 0$

Que tiene como raíces o polos del sistema , los polos conjugados :

$P1 = 0.5 + j\,0.6181$
$P2 = 0.5 - j\,0.6181$

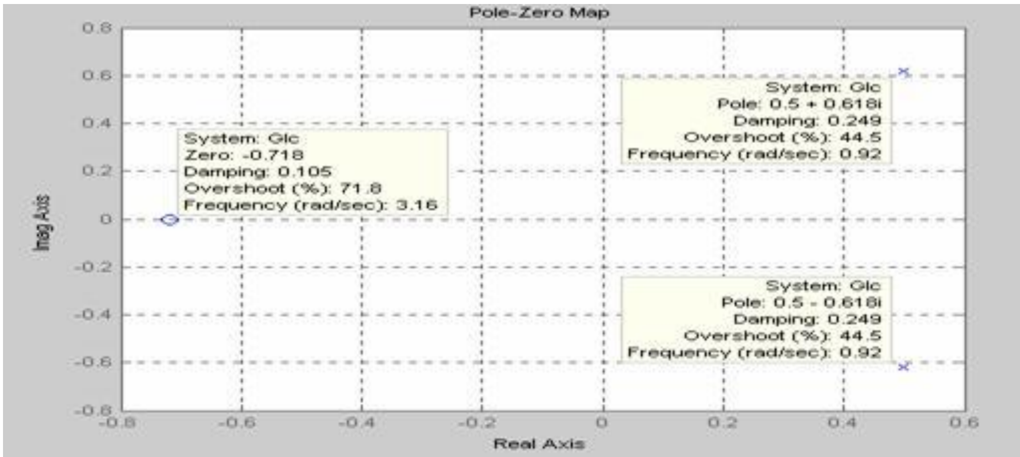
Cuya magnitud y ángulos con respecto al eje de las x's son :

$$|P1| = |P2| = \sqrt{0.5^2 + 0.6181^2} = 0.7951$$

$\Phi_1 = \tan^{-1}(0.6181/0.5) = 51.03^o$,

$\Phi_2 = \tan^{-1}(-0.6181/0.5) = -51.03^o$,

La ubicación de estos polos en el plano-z es la siguiente :



Como la magnitud de los polos es menor que 1, están dentro el círculo unitario y por lo tanto el sistema es estable si T = 1 sg.

Para un periodo de muestreo de T =10 sg el sistema no es estable. Compruébelo

Conclusión : La estabilidad de un sistema de control en lazo cerrado se puede perder al aumentar el periodo de muestreo.

Simulación :

Digite y corra el siguiente programa en Matlab :

```
% EJEMPLO 4-1 : ESTABILIDAD DE UN SISTEMA DISCRETO
clc
disp(' ')
disp('EJEMPLO 4-1 : ESTABILIDAD');
T = input('ENTRE TIEMPO DE MUESTREO : T = ');
num = input('Entre numerador de la planta : num = ');
den = input('Entre denominador de la planta : den = ');
disp(' ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : G(s) = ');
Gs = tf(num,den);
Gs = zpk(Gs)
disp(' ');
disp('PRESIONE TECLA ENTER');
pause
clc
disp(' ');
disp('LA FUNCION DE TRANSFERENCIA DISCRETIZADA ES : G(z) = ');
Gz = c2d(Gs,T,'zoh')
Gla = Gz;
```

```
disp(' ');
disp('PRESIONE TECLA ENTER');
pause
disp(' ');
disp('LA FUNCION DE TRANSF.  EN LAZO CERRADO ES : Glc(z) = ');
Glc = feedback(Gla,1)
disp(' ');
disp('PRESIONE TECLA ENTER');
pause
clc
disp('LOS POLOS DEL SISTEMA EN LAZO CERRADO SON : ');
Polos = pole(Glc)
disp(' ');
disp('SUS MAGNITUDES Y ANGULOS SON : ');
Mag = abs(Polos)
Ang = angle(Polos)*180/pi
disp(' ');
disp('EL SISTEMA ES ESTABLE ?');
disp(' ');
if Mag<1
disp('RESPUESTA : EL SISTEMA ES ESTABLE');
else
disp('RESPUESTA : EL SISTEMA NO ES ESTABLE');
end
disp(' ');
disp('GRAFICA DE LO POLOS Y CEROS EN EL PLANO-Z');
pzmap(Glc)
grid
```

4.3 MÉTODOS DE ESTABILIDAD

Son métodos que se utilizan para probar la estabilidad de un sistema de control sin necesidad de calcular las raíces de la ecuación característica. Los métodos más conocidos para sistemas analizados con funciones de transferencia son el Método de Jury y el Método de Routh - Hurwitz.

4.3.1 MÉTODO DE JURY

Este método consiste en realizar una tabla en donde las filas están conformadas por coeficientes del polinomio de la ecuación característica arreglados de la siguiente forma :

Ecuación característica :

$P(z) = a_0 z^n + a_1 z^{n-1} + a_2 z^{n-2} + a_3 z^{n-3} + \ldots + a_{n-2} z^{n-2} + a_{n-1} z + a_n$

Filas	Coeficientes							
1	a_n	a_{n-1}	a_{n-2}	a_{n-3}	. . .	a_2	a_1	a_0
2	a_0	a_1	a_2	a_3	. . .	a_{n-2}	a_{n-1}	a_n
3	b_{n-1}	b_{n-2}	b_{n-3}	. . .	b_2	b_1	b_0	
4	b_0	b_1	b_2	. . .	b_{n-3}	b_{n-2}	b_{n-1}	
5	c_{n-2}	c_{n-3}	c_{n-4}	. . .	c_1	c_0		
6	c_0	c_1	c_2	. . .	c_{n-3}	c_{n-2}		
. . .	. . .	. . .	. . .	. . .				
n -2	p_3	p_2	p_1	p_0				
n -1	p_0	p_1	p_2	p_3				
n	q_2	q_1	q_0	última fila				

Nota : El último renglón sólo debe tener tres elementos.

Conocidos los coeficientes de la ecuación característica los demás coeficientes se calculan de la siguiente forma :

$$b_k = \begin{bmatrix} a_n & a_{n-1-k} \\ a_0 & a_{k+1} \end{bmatrix} \qquad k = 0, 1, 2, 3, \ldots, n-2, n-1$$

$$c_k = \begin{bmatrix} b_{n-1} & b_{n-2-k} \\ b_0 & b_{k+1} \end{bmatrix} \qquad k = 0, 1, 2, 3, \ldots, n-3, n-2$$

. . . .

$$q_k = \begin{bmatrix} p_3 & p_{2-k} \\ p_0 & p_{k+1} \end{bmatrix} \qquad k = 0, 1, 2$$

Según Jury, el sistema es estable si se cumplen las siguientes condiciones :

1. $|a_n| < a_0$

2. $P(z)\big|_{z=1} > 0$

3. $P(z)\big|_{z=-1} \begin{cases} > 0, & si \ n \ es \ par \\ < 0, & si \ n \ es \ impar \end{cases}$

4. $|b_{n-1}| > |b_0|,\ |c_{n-2}| > |c_0|,\ \ldots\ldots\ldots, |q_2| > |q_0|$

EJEMPLO 4-2 :

Aplicar el método de Jury para determinar la estabilidad de los sistemas :

(a) Cuya planta es : Gp(s) = 10/ (s+1)(s+2), T = 0.1 sg
(b) Cuya ecuación característica es : $P(z) = z^3 - 1.1 z^2 – 0.1 z + 0.2$
(c) Cuya ecuación característica es : $P(z) = z^4 - 1.2 z^3 + 0.07 z^2 + 0.3 z - 0.08$

Solución :

```
% EJEMPLO 4-2 : ESTABILIDAD SEGUN METODO DE JURY
clc
disp('EJEMPLO 4-2 : ESTABILIDAD SEGUN METODO DE JURY ');
disp(' ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
disp(' 3: ECUACION CARACTERISTICA DEL SISTEMA');
n=input('SELECCIONE LA OPCION :  ');
disp(' ');
switch n
case 1
num = input('Entre coeficientes del numerador : num = ');
den = input('Entre coeficientes del denominador : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den)
clc
disp(' ');
T = input('Enter tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')
disp(' ');
disp('LA FUNCION DE TRANSF. LAZO CERRADO ES : Glc(z) = ');
Glc = feedback(Gz,1)
[numz,denz] = tfdata(Glc,'v');

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
```

```
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K)
clc
disp(' ');
T = input('Enter tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')
disp(' ');
disp('LA FUNCION DE TRANSF. LAZO CERRADO ES : Glc(z) = ');
Glc = feedback(Gz,1)
[numz,denz] = tfdata(Glc,'v');

case 3
denz = input('COEFICIENTES DE LA ECUACION CARACTERISTICA : denz = '
);
end
disp(' ');
clc
disp('EL ORDEN DE LA ECUACION CARACTERISTICA ES : n = ')
n = length(denz)-1

% OBTENCION DE MATRIZ DE JURY
disp(' ');
m = n + 1;
for i=1:m
   a(i)=denz(i);
   end
pz11 = polyval(denz,1);
if abs(pz11) < 1.0e-5
   pz1=0;
else
   pz1=pz11;
end
pz2 = polyval(denz,-1);
x = n - 1;
switch x

% SISTEMA DE ORDEN 2
case 1
   disp(' ');
   disp('LA MATRIZ DE JURY ES : a2 a1 a0  = ');
   [a(3) a(2) a(1)]
% PRUEBA DE ESTABILIDAD
   if (pz1 == 0)&(abs(a(m)) < abs(a(1)))
      disp('EL SISTEMA ES CRITICAMENTE ESTABLE')
   end
```

```
  if abs(a(m)) >= a(1)
   disp('EL SISTEMA ES INESTABLE ');
   elseif pz1 <= 0
      disp('EL SISTEMA ES INESTABLE');
   elseif  pz2 >0
      disp('EL SISTEMA ES ESTABLE');
   else
      disp('EL SISTEMA ES INESTABLE');
   end

% SISTEMA DE ORDEN 3
case 2
   disp(' ');
   disp('LA MATRIZ DE JURY ES : ');
   disp(' ');
   disp('a3 a2 a1 a0');
   disp('a0 a1 a2 a3');
   disp('b2 b1 b0   ');
    for k=1:m-1
    b(k)=det([a(m) a(m-k); a(1) a(k+1)]);
    end
   A1 = [a(4) a(3) a(2) a(1)];
   A2 = [a(1) a(2) a(3) a(4)];
   B  = [b(3) b(2) b(1)   0 ];
   [ A1; A2; B]
   pause
   clc
   disp(' ');
disp('PRUEBA DE ESTABILIDAD : ');
disp('a(n) = ');
a(m)
disp('a(0) = ');
a(1)
disp('Para z =1, P(z) = ');
[pz1]
if (pz1 == 0)&(abs(a(m)) < abs(a(1)))
      disp('EL SISTEMA ES CRITICAMENTE ESTABLE')
end
disp('Para z = -1, P(z) = ');
[pz2]
pause
clc
disp(' ');
if abs(a(m)) >= a(1)
   disp('EL SISTEMA ES INESTABLE ');
   elseif pz1 < 0
      disp('EL SISTEMA ES INESTABLE');
```

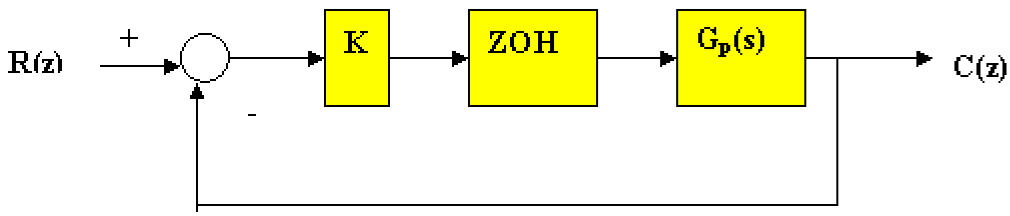
```
  elseif  pz2 >=0
     disp('EL SISTEMA ES INESTABLE');
  elseif (abs(b(m-1)) > abs(b(1)))
     disp(' b(n-1) = ');
     b(m-1)
     disp(' b(0) = ');
     b(1)
     disp('EL SISTEMA ES ESTABLE');
  else
     disp('EL SISTEMA ES INESTABLE');
  end
% SISTEMA DE ORDEN 4
case 3
  disp(' ');
  disp('LA MATRIZ DE JURY ES :');
  disp(' ');
  disp('a4 a3 a2 a1 a0 ');
  disp('a0 a1 a2 a3 a4 ');
  disp('b3 b2 b1 b0    ');
  disp('b0 b1 b2 b3    ');
  disp('c2 c1 c0       ');
   for k=1:m-1
   b(k)=det([a(m) a(m-k); a(1) a(k+1)]);
   end
   for k=1:m-2
   c(k)=det([b(m-1) b(m-1-k); b(1) b(k+1)]);
   end
  A1 = [a(5) a(4) a(3) a(2) a(1)];
  A2 = [a(1) a(2) a(3) a(4) a(5)];
  B1 = [b(4) b(3) b(2) b(1)   0 ];
  B2 = [b(1) b(2) b(3) b(4)   0 ];
  C  = [c(3) c(2) c(1)  0     0 ];
  [A1; A2; B1; B2; C]
%PRUEBA DE ESTABILIDAD
  if (pz1 == 0)&(abs(a(m)) < abs(a(1)))
     disp('EL SISTEMA ES CRITICAMENTE ESTABLE')
  end
if abs(a(m)) >= a(1)
  disp('EL SISTEMA ES INESTABLE ');
  elseif pz1 < 0
     disp('EL SISTEMA ES INESTABLE');
  elseif  pz2 <=0
     disp('EL SISTEMA ES INESTABLE');
  elseif (abs(b(m-1)) <= abs(b(1)))
     disp('EL SISTEMA ES INESTABLE');
  elseif (abs(c(m-2)) > abs(c(1)))
     disp('EL SISTEMA ES ESTABLE');
```

```
   else
      disp('EL SISTEMA ES INESTABLE');
   end

end
```

EJEMPLO 4-3 :

Para la figura determinar por el método de Jury el rango de la ganancia K de tal forma que el sistema siga siendo estable. El tiempo de muestreo T = 0.1 sg y la función de transferencia de la planta es igual a :

$$G_p(s) = \frac{10}{(s+1)(s+2)}$$




```
% EJEMPLO 4-3: DETERMINAR EL RANGO DE ESTABILIDAD DE UN SISTEMA
clc
disp('CON ESTE EJEMPLO SE DETERMINARA EL RANGO DE');
disp('ESTABILIDAD DE UN SISTEMA CON REALIMENTACION UNITARIA ');
disp(' ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
disp(' 3: FUNCION DE TRANSFERENCIA G(z)');
n=input('SELECCIONE LA OPCION :  ');
disp(' ');
switch n
case 1
num = input('Entre coeficientes del numerador : num = ');
den = input('Entre coeficientes del denominador : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den)
clc
disp(' ');
T = input('Entre tiempo de muestreo : T = ');
```

```
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K)
clc
disp(' ');
T = input('Enter tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')

case 3
T = input('Entre periodo de muestreo : T = ');
num = input('Entre coeficientes del numerador : num = ');
den = input('Entre coeficientes del denominador : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : G(z) = ');
Gz = tf(num,den,T)
end
[numz,denz] = tfdata(Gz,'v');
syms K
Pz = denz + K*numz;
EcuaCaract = poly2sym(Pz,'z')
n = length(Pz);
an = Pz(n)
a0 = Pz(1)
disp('SEGUN PRIMER CRITERIO ');
Limite1 = solve(abs(an)-a0,'K');
Max1 = numeric(Limite1(1))
Min1 = numeric(Limite1(2))
disp(' ');
pause
clc
disp(' ');
disp('SEGUN SEGUNDO CRITERIO ');
z=1;
Pz1 = eval(EcuaCaract)
Min2=solve(Pz1,'K');
Min2 = numeric(Min2)
pause
clc
disp(' ');
```

```
disp('SEGUN TERCER CRITERIO ');
z= - 1;
Pz2 = eval(EcuaCaract)
Max2 = solve(Pz2,'K');
Max2 = numeric(Max2)
pause
clc
disp(' ');
disp('EL RANGO DE LA GANANCIA  K PARA QUE HAYA ESTABILIDAD ES :
');
if Max1 <= Max2
   Max = Max1
else
   Max = Max2
end
if Min1 <= Min2
   Min = Min2;
else
   Min = Min1;
end
if Min <= 0
   Min = 0
else
   Min = Min
end
```

4.3.2 MÉTODO DE RUTH - HURWITZ

El método requiere de la transformación de un plano-z discreto a otro plano-w continuo utilizando la transformación bilineal :

$$z = \frac{1 + (T/2)w}{1 - (T/2)w} \quad , \; T : \text{Tiempo de muestreo}$$

entonces G(z) se convierte en G(w) y se aplica este método como si fuera un sistema continuo.

Si se tiene directamente la ecuación característica del sistema en lazo cerrado se prefiere el reemplazo :

$$z = \frac{w + 1}{w - 1}$$

La ecuación característica en términos de w es de la forma :

$Q(w) = a_0\, w^{n} + a_1\, w^{n-1} + a_2\, w^{n-2} + \ldots\ldots + a_{n-1}\, w + a_n$

Para un sistema de orden 4 el arreglo de Ruth es :

Fila	W^{n} :	a_0	a_2	a_4	a_6	....
Fila	W^{n-1} :	a_0	a_3	a_5	a_7	....
Fila	W^{n-2} :	b_1	b_2	b_3	b_4	....
Fila	W^{n-3} :	c_1	c_2	c_3	c_4	....
Fila	W^{n-4} :	d_1	d_2	d_3	d_4	....
....	....	....	....	....	....	....
Fila	W^{2} :	e_1	e_2			
Fila	W^{1} :	f_1				
Fila	W^{0} :	g_1				

Los coeficientes del arreglo se calculan de la siguiente forma :

$$b_1 = \frac{a_1 a_2 - a_0 a_3}{a_1}$$

$$b_2 = \frac{a_1 a_4 - a_0 a_5}{a_1}$$

$$b_3 = \frac{a_1 a_6 - a_0 a_7}{a_1}$$

. . . .

$$c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1}$$

$$c_2 = \frac{b_1 a_5 - a_1 b_3}{b_1}$$

. . . .

Casos especiales :

1. ***Si antes de completar la tabla, el primer elemento de cualquier renglón es cero, pero no los demás, se reemplaza el cero por una constante***

arbitrariamente pequeña epsilon ε y se continua con el arreglo de Routh.

2. ***Si antes de completar la tabla todos los elementos de un renglón son ceros, se forma una ecuación auxiliar con los elementos del renglón anterior como coeficientes y se halla su derivada. Los coeficientes de esta derivada son os nuevos elementos del arreglo que se rremplaza por el renglón de ceros y se continua con la construcción del arreglo.***

Prueba de estabilidad de Routh-Hurwitz :

El número de raíces con partes reales positivas es igual al número de cambios de signo de los coeficientes de la primera columna del arreglo.

EJEMPLO 4-4 :

Aplicar el método de Routh – Hurwitz para determinar la estabilidad del sistema en lazo cerrado con realimentación unitaria formado por :

$$G_p(s) = \frac{10}{(s+1)(s+2)} \quad , \quad T = 0.1 \text{ sg}$$

Solución:

1. ***Encontrar la función de transferencia del sistema en lazo cerrado Glc(z)***
2. ***Encontrar la ecuación característica del sistema P(z)***
3. ***Reemplazar la variable 'z' por el valor z = f(w) de la transformación bilineal***
4. ***Encontrar la ecuación característica continua en términos de w, Q(w)***
5. ***Obtener el arreglo de Routh aplicando las fórmulas para obtener sus coeficientes o elementos del arreglo. Tener en cuenta si se produce en la construcción un caso especial.***
6. ***Aplicar el criterio de estabilidad***

Simulación:

```
% EJEMPLO 4-4 : ESTABILIDAD SEGUN METODO DE ROUTH-HURWITZ
% POR : M.I. JORGE ANTONIO POLANIA PUENTES
clc
disp(' RANGO DE ESTABILIDAD SEGUN METODO DE ROUTH-HURWITZ ');
disp(' ');
disp('POR : M. I. JORGE ANTONIO POLANIA PUENTES');
disp(' ');
```

```
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION :  ');
disp(' ');
disp('1. ENCONTRAR LA FUNCION DE TRANSF. LAZO CERRADO');

switch n
case 1
num = input('Entre coeficientes del numerador : num = ');
den = input('Entre coeficientes del denominador : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den)
clc
disp(' ');
T = input('Enter tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')
disp(' ');
disp('LA FUNCION DE TRANSF. LAZO CERRADO ES : Glc(z) = ');
Glc = feedback(Gz,1)
[numz,denz] = tfdata(Glc,'v');

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K)
clc
disp(' ');
T = input('Entre tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')
disp(' ');
disp('LA FUNCION DE TRANSF. LAZO CERRADO ES : Glc(z) = ');
Glc = feedback(Gz,1)
[numz,denz] = tfdata(Glc,'v');
end
pause

clc
disp(' ');
disp('2. ENCONTRAR G(w) CON TRANSF. BILINEAL');
disp(' ');
```

```
disp(' a) VECTOR DE CAMBIO DE SIGNO ES :');
i=1;
a1=[ ];
while i<= n+1
  a(i)=1;
  if i==n+1
    a1=[a(i) a1];
  else
  a(i+1)=-1;
  a1= [a(i+1) a(i) a1];
  end
i=i+2;
end
display(a1)
% Fin de rutina
disp(' ');
disp(' b) LA ECUACION CARACTERISTICA DE G(w) ES:  ');
num=numz.*a1;
den=denz.*a1;
num=numeric(num);
den=numeric(den);
[numv,denv]=bilinear(num,den,0.5);

v1=[ ];
% v = -(T/2)w
v2 = -T/2;
for i=1:(n+1)
    v(i)= v2^(i-1);
    v1= [ v(i) v1];
end
display(v1)
numw=numv.*v1;
denw=denv.*v1;
Gpw=tf(numw,denw);
denwsym = poly2sym(denw,'w');
pretty(denwsym)
pause

clc
disp(' ');
if denw(1)<0
   denw = denw*(-1);
else
   denw = denw;
end
disp('LOS COEFICIENTES DE LA ECUACION DE Q(w) SON :');
denw = denw/denw(1)
```

```
disp('3. OBTENCION DE LOS COEFICIENTES DE ROUTH  = ')
clc
disp(' ');
n = length(denz)-1;
disp(' ');
x = n-1;
switch x
case 1
   clc
   disp('LA MATRIZ DE ROUTH ES : ');
   disp(' W2 = a0  a2');
   disp(' W1 = a1   0');
   disp(' W0 = b1   0');
   disp(' ');
   W2 = [denw(1) denw(3)];
   W1 = [denw(2)   0 ];
   disp(' RENGLONES : W2 y W1');
   [W2; W1]
   pause
   if W1 ==0
      disp('COMO W1 = 0, SE DERIVA RENGLON ANTERIOR');
      Az=poly2sym([denw(1) 0 den(3)],'w')
      dAz=diff(Az)
      den=sym2poly(dAz)
      W1= [den(1) 0]
   else
      W1=W1
   end
   b1 = denw(3);
   W0 = [b1  0 ];
   disp(' ');
   disp('EL ARREGLO DE ROUTH ES : ')
   [W2; W1; W0]
   disp(' ');
   disp('PRUEBA DE ESTABILIDAD');
   if (denw(1)>0)&(denw(2)<0)
      disp('EL SISTEMA ES INESTABLE');
   elseif (denw(2)>0)&(b1<0)
      disp('EL SISTEMA ES INESTABLE');
   else
      disp('EL SISTEMA ES ESTABLE');
   end
   pause
   disp(' ');

case 2
   pause
```

```
clc
disp('LA MATRIZ DE ROUTH ES : ');
disp(' W3 =   a0   a2');
disp(' W2 =   a1   a3');
disp(' W1 =   b1    0');
disp(' W0 =   a3    0');
disp(' ');
disp(' LAS FILAS W3 y W2 SON :');
W3 = [denw(1) denw(3)];
W2 = [denw(2) denw(4)];
[W3; W2]
if W2==0
   disp(' W2 = 0, SE DEBE DERIVAR W3');
   Az=poly2sym([denw(1)  0 denw(3) 0],'w')
   dAz=diff(Az)
   den=sym2poly(dAz)
   denw(2)=den(1)
   denw(4)=den(3)
elseif denw(2)==0
     disp(' SI a1 = 0, ENTONCES a1 = EPSILON');
     denw(2)= 1.0e-6
  else
   denw(2)=denw(2);
end
W2 = [denw(2) denw(4)];
b1 = (denw(2)*denw(3) - denw(1)*denw(4))/ denw(2);
W1 = [b1 0];
disp('LAS FILAS W3, W2, W1 SON :');
[W3; W2; W1]
if b1 ==0
   disp(' W1 = 0, SE DEBE DERIVAR W2');
   Az=poly2sym([denw(2) 0 denw(4)],'w')
   Az=diff(Az)
   den=sym2poly(Az)
   b1=den(1);
else
   b1=b1;
end

W1 = [  b1    0 ];
W0 = [denw(4) 0 ];
disp('EL ARREGLO DE ROUTH ES :');
[W3; W2; W1; W0]
disp('PRUEBA DE ESTABILIDAD DE ROUTH-HURWITZ :');
disp(' ');
if (denw(1)>0)&(denw(2)<0)
   disp('EL SISTEMA ES INESTABLE');
```

```
   elseif (denw(2)>0)&(b1<0)
      disp('EL SISTEMA ES INESTABLE');
   elseif (b1>0)&(denw(3)<0)
      disp('EL SISTEMA ES INESTABLE');
   else
      disp('EL SISTEMA ES ESTABLE');
   end
end
```

EJEMPLO 4-5 :

Determinar si los sistemas con la ecuación característica dada es estable :

(a) $F(z) = z^3 - 1.25\,z^2 - 1.375\,z - 0.25$
(b) $F(z) = z^3 + 3.3\,z^2 - 3\,z + 0.8$
(c) $F(z) = z^3 - 1.3\,z^2 - 0.08\,z + 0.24$
(d) $F(z) = z^3 + 2\,z^2 + z + 2$
(e) $F(z) = z^2 - 0.25$

Solución:

1. ***Reemplazar la variable 'z' por $z = (w + 1) / (w - 1)$ con el fin de obtener la ecuación característica en términos de w, Q(w)***
2. ***Obtener el arreglo de Routh aplicando las fórmulas para obtener sus coeficientes o elementos del arreglo. Tener en cuenta si se produce en la construcción un caso especial.***
3. ***Aplicar el criterio de estabilidad***

Simulación:

```
% EJEMPLO 4-5 : ESTABILIDAD SEGÚN MÉTODO DE ROUTH-HURWITZ
% POR : M.I. JORGE ANTONIO POLANÍA PUENTES
clc
disp('EJEMPLO 4-5 : ESTABILIDAD SEGUN METODO DE ROUTH-HURWITZ ');
disp('POR : M.I. JORGE ANTONIO POLANIA PUENTES '),
disp(' ');
   denz = input('COEFICIENTES DE LA ECUACION CARACTERISTICA : denz = ' );
   n = length(denz)-1;
   syms w
   disp('1. OBTENER COEFICIENTES DE LA ECUACION Q(w) ');
   z=(w+1)/(w-1)
   denzsym = poly2sym(denz,'z');
   denwsym =eval(denzsym);
```

```
   denwsym =simplify(denwsym);
   denwsym =denwsym*(w-1)^n;
   denw = sym2poly(denwsym)
   m = length(denw)-1;
   if m==n-1
     denw(n+1) =0;
   else
     denw=denw;
   end

   if denw(1)<0
      denw = denw*(-1);
   else
      denw = denw;
   end

   denw = denw/denw(1)

pause

clc
disp(' ');
disp(' OBTENCION DEL ARREGLO DE ROUTH');
disp(' ');
x = n-1;
switch x
case 1
   clc
   disp('LA MATRIZ DE ROUTH ES : ');
   disp(' W2 = a0  a2');
   disp(' W1 = a1   0');
   disp(' W0 = b1   0');
   disp(' ');
   W2 = [denw(1) denw(3)];
   W1 = [denw(2)   0 ];
   disp(' RENGLONES : W2 y W1');
   [W2; W1]
   pause
   if W1 ==0
      disp('COMO W1 = 0, SE DERIVA RENGLON ANTERIOR');
      Az=poly2sym([denw(1) 0 den(3)],'w')
      Az=diff(Az)
      den=sym2poly(Az)
      W1= [den(1) 0]
   else
      W1=W1
   end
```

```
   b1 = denw(3);
   W0 = [b1  0 ];
   disp(' ');
   disp('EL ARREGLO DE ROUTH ES : ')
   [W2; W1; W0]
   disp(' ');
   disp('PRUEBA DE ESTABILIDAD');
   if (denw(1)>0)&(denw(2)<0)
      disp('EL SISTEMA ES INESTABLE');
   elseif (denw(2)>0)&(b1<0)
      disp('EL SISTEMA ES INESTABLE');
   else
      disp('EL SISTEMA ES ESTABLE');
   end
   pause
   disp(' ');
case 2
   pause

   clc
   disp('LA MATRIZ DE ROUTH ES : ');
   disp(' W3 =   a0   a2');
   disp(' W2 =   a1   a3');
   disp(' W1 =   b1    0');
   disp(' W0 =   a3    0');
   disp(' ');
   disp(' LAS FILAS W3 y W2 SON :');
   W3 = [denw(1) denw(3)];
   W2 = [denw(2) denw(4)];
   [W3; W2]
   if W2==0
      disp(' W2 = 0, SE DEBE DERIVAR W3');
      Az=poly2sym([denw(1)  0 denw(3) 0],'w')
      dAz=diff(Az)
      den=sym2poly(dAz)
      denw(2)=den(1)
      denw(4)=den(3)
   elseif denw(2)==0
        disp(' SI a1 = 0, ENTONCES a1 = EPSILON');
        denw(2)= 1.0e-6
     else
      denw(2)=denw(2);
   end
   W2 = [denw(2) denw(4)];
   b1 = (denw(2)*denw(3) - denw(1)*denw(4))/ denw(2);
   W1 = [b1 0];
   pause
```

```
   clc
   disp('LAS FILAS W3, W2, W1 SON :');
   [W3; W2; W1]
   if b1 ==0
      disp(' W1 = 0, SE DEBE DERIVAR W2');
      Az=poly2sym([denw(2) 0 denw(4)],'w')
      Az=diff(Az)
      den=sym2poly(Az)
      b1=den(1);
   else
      b1=b1;
   end

   W1 = [  b1    0 ];
   W0 = [denw(4) 0 ];
   disp('EL ARREGLO DE ROUTH ES :');
   [W3; W2; W1; W0]
   disp('PRUEBA DE ESTABILIDAD DE ROUTH-HURWITZ :');
   disp(' ');
   if (denw(1)>0)&(denw(2)<0)
      disp('EL SISTEMA ES INESTABLE');
   elseif (denw(2)>0)&(b1<0)
      disp('EL SISTEMA ES INESTABLE');
   elseif (b1>0)&(denw(3)<0)
      disp('EL SISTEMA ES INESTABLE');
   else
      disp('EL SISTEMA ES ESTABLE');
   end
end
```

EJEMPLO 4-6 :

Encontrar el rango de estabilidad de un sistema en lazo cerrado si :

$$Gp(s) = \frac{k}{s(s+1)} \quad , \quad T = 0.1sg$$

Simulación :

```
% EJEMPLO 4-6 : RANGO DE ESTABILIDAD SEGUN METODO DE ROUTH-
    HURWITZ
% POR : M.I. JORGE ANTONIO POLANIA PUENTES
clc
disp(' RANGO DE ESTABILIDAD SEGUN METODO DE ROUTH-HURWITZ ');
disp(' ');
```

```
disp('POR : M. I. JORGE ANTONIO POLANIA PUENTES');
disp(' ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION :  ');
disp(' ');

switch n
case 1
num = input('Entre coeficientes del numerador : num = ');
den = input('Entre coeficientes del denominador : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den)
disp(' ');
T = input('Enter tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K)
clc
disp(' ');
T = input('Entre tiempo de muestreo : T = ');
disp(' ');
disp('LA FUNCION DE TRANSF. DISCRETA DE LA PLANTA : G(z) = ');
Gz = c2d(Gp,T,'zoh')
end
pause
clc
disp('COEFICIENTES DE LA ECUACION CARACTERISTICA SON :');
[numz,denz] = tfdata(Gz,'v');
syms K
denz = denz + K*numz
EcuaCaract = poly2sym(denz,'z')
n = length(denz)-1;
ESTABLE = 1;
K =0;

while ESTABLE == 1
syms w
 disp('1. OBTENER COEFICIENTES DE LA ECUACION Q(w) ');
```

```
   z=(w+1)/(w-1)
   denzsym = poly2sym(denz,'z');
   denwsym =eval(denzsym);
   denwsym =simplify(denwsym);
   denwsym =denwsym*(w-1)^n;
   denw = sym2poly(denwsym);
   m = length(denw)-1;
   if m==n-1
     denw(n+1) =0;
   else
     denw=denw;
   end

   if denw(1)<0
      denw = denw*(-1);
   else
      denw = denw;
   end

   denw = denw/denw(1);

disp(' ');
if denw(1)<0
   denw = denw*(-1);
else
   denw = denw;
end
disp('LOS COEFICIENTES DE LA ECUACION DE Q(w) SON :');
denw = denw/denw(1);
disp('3. OBTENCION DE LOS COEFICIENTES DE ROUTH  = ')
disp(' ');
n = length(denz)-1;
disp(' ');
x = n-1;
switch x
% Sistema de segundo orden
case 1
clc
disp('LA MATRIZ DE ROUTH ES : ');
disp(' W2 = a0  a2');
disp(' W1 = a1   0');
disp(' W0 = b1   0');
disp(' ');
   W2 = [denw(1) denw(3)];
   W1 = [denw(2)   0 ];
disp(' RENGLONES : W2 y W1');
   [W2; W1];
```

```
  if W1 ==0
     disp('COMO W1 = 0, SE DERIVA RENGLON ANTERIOR');
     Az=poly2sym([denw(1) 0 den(3)],'w');
     Az=diff(Az);
     den=sym2poly(Az);
     W1= [den(1) 0];
  else
     W1=W1;
  end
  b1 = denw(3);
  W0 = [b1  0 ];
  disp(' ');
  disp('EL ARREGLO DE ROUTH ES : ')
  [W2; W1; W0]
  disp(' ');
  if (denw(1)>0)&(denw(2)<0)
     disp('EL SISTEMA ES INESTABLE PARA K = ');
     display(K)
     ESTABLE = 0;
  elseif (denw(2)>0)&(b1<0)
     disp('EL SISTEMA ES INESTABLE PARA K = ');
     display(K)
     ESTABLE = 0;
  else
     disp('EL SISTEMA ES ESTABLE PARA UN K = ');
     display(K)
     ESTABLE = 1;
  end

  disp(' ');

 % Sistema de tercer orden
case 2
clc
disp('LA MATRIZ DE ROUTH ES : ');
disp(' W3 =   a0   a2');
disp(' W2 =   a1   a3');
disp(' W1 =   b1    0');
disp(' W0 =   a3    0');
disp(' ');
disp(' LAS FILAS W3 y W2 SON :');
  W3 = [denw(1) denw(3)];
  W2 = [denw(2) denw(4)];
  [W3; W2];
  if W2==0
     disp(' W2 = 0, SE DEBE DERIVAR W3');
     Az=poly2sym([denw(1)  0 denw(3)],'w');
```

```
      Az=diff(Az);
      den=sym2poly(Az);
      denw(2)=den(1);
      denw(4)=den(2);
      elseif denw(2)==0
      disp(' SI a1 = 0, ENTONCES a1 = EPSILON');
      denw(2)= 1.0e-6;
     else
      denw(2)=denw(2);
   end
   W2 = [denw(2) denw(4)];
   b1 = (denw(2)*denw(3) - denw(1)*denw(4))/ denw(2);
   W1 = [b1 0];
disp('LAS FILAS W3, W2, W1 SON :');
   [W3; W2; W1];
   if b1 ==0
   disp(' W1 = 0, SE DEBE DERIVAR W2');
      Az=poly2sym([denw(2) 0 denw(4)],'w');
      Az=diff(Az);
      den=sym2poly(Az);
      b1=den(1);
   else
      b1=b1;
   end

   W1 = [  b1    0 ];
   W0 = [denw(4) 0 ];
   disp('EL ARREGLO DE ROUTH ES :');
   [W3; W2; W1; W0]
   disp(' ');
   if (denw(1)>0)&(denw(2)<0)
      disp('EL SISTEMA ES INESTABLE PARA K = ');
      display(K)
      elseif (denw(2)>0)&(b1<0)
      disp('EL SISTEMA ES INESTABLE PARA K = ');
      display(K)
      elseif (b1>0)&(denw(3)<0)
      disp('EL SISTEMA ES INESTABLE PARA K = ');
      display(K);
      else
      disp('EL SISTEMA ES ESTABLE PARA UN K = ');
      display(K)
   end
end

pause
```

K =K+1;
end

4.4 ESTABILIDAD RELATIVA

La estabilidad absoluta es requisito básico en todo sistema de control, pero también se requiere de una buena estabilidad relativa, esto es, de una respuesta transitoria satisfactoria, de una precisión en estado transitorio y de una buena respuesta a la entrada de perturbaciones. Para el análisis de los sistemas de control se utiliza comúnmente entradas como el escalón unitario y la rampa porque son fáciles de generar y proporcionan una clara información en la respuesta del sistema.

El análisis transitorio de un sistema lineal invariante en el tiempo LTI puede realizarse mediante la respuesta transitoria c(t) y la respuesta del estado estacionario $c_{ss}(t)$. La respuesta transitoria es originada por las características dinámicas del sistema y determina el comportamiento del sistema durante la transición de un estado inicial a otro final. La respuesta estacionaria depende fundamentalmente de la excitación del sistema y si el sistema es estable es la respuesta que perdura cuando el tiempo crece infinitamente.

El error en estado estacionario es la diferencia entre la señal de referencia y la señal realimentada en estado estacionario en sistemas estables. En un sistema de control lo importante es minimizar este error, por ello se requiere conocer la respuesta transitoria respecto a entradas fundamentales.

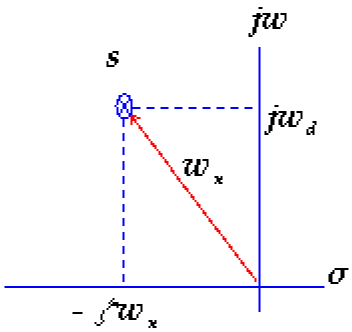
4.4.1 RESPUESTA TRASITORIA

SISTEMA DE SEGUNDO ORDEN

Para un sistema o planta de segundo orden, su función de transferencia es igual a:

$$G(s) = \frac{w_x{}^2}{s^2 + 2\zeta w_x{}^2 s + w_x{}^2} \;, \; s_{1,2}(polos) = -\zeta w_x \pm w_x \sqrt{\zeta^2 - 1}$$

ζ = Factor de amortiguamiento
w_n = Frecuencia natural no amortiguada



$$w_n^{\,2} = w_d^{\,2} + (\zeta\, w_n)^2$$
$$w_d = w_n\sqrt{1-\zeta^2}$$

EJEMPLO 4-7A :

Graficar para una frecuencia natural w_n = 1 rd/sg la respuesta al escalón unitario de un sistema (a) oscilatorio ζ = 0, (b) subamortiguado 0< ζ<1 , ζ = 0.5, (c) amortiguado ζ = 1.0, (d) sobreamortiguado ζ = 1.5

Solución :

Reemplazando w_n = 1 rd/sg en la ecuación de un sistema de segundo orden :

$$G(s) = \frac{1}{s^2 + 2\zeta\, s + 1} \quad , \; s_{1,2}(polos) = -\zeta \pm \sqrt{\zeta^2 - 1}$$

Para una entrada de escalón unitario G(s) = 1/s, su salida es :

$$C(s) = \frac{w_n^{\,2}}{s^2 + 2\zeta\, w_n^{\,2}\, s + 1} \cdot \frac{1}{s},$$

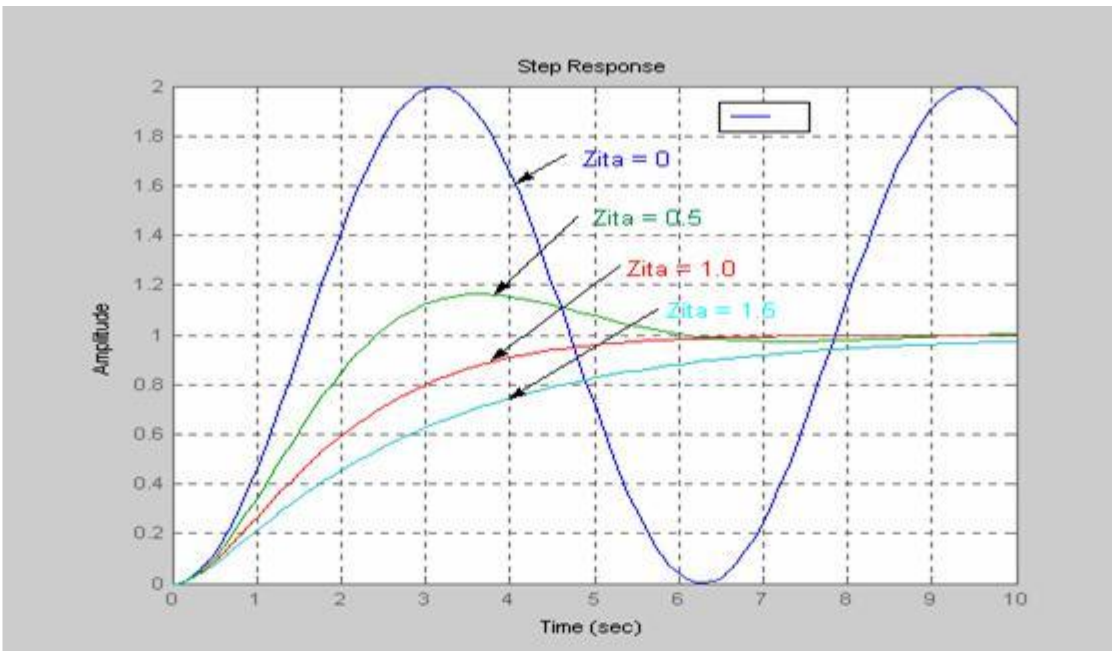
$$c(t) = 1 - \frac{e^{-\zeta\, w_n t}}{\sqrt{1-\zeta^2}} .sen(w_d t + \theta, \qquad \theta = tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)$$

Simulación :

PROGRAMA EN MATLAB :

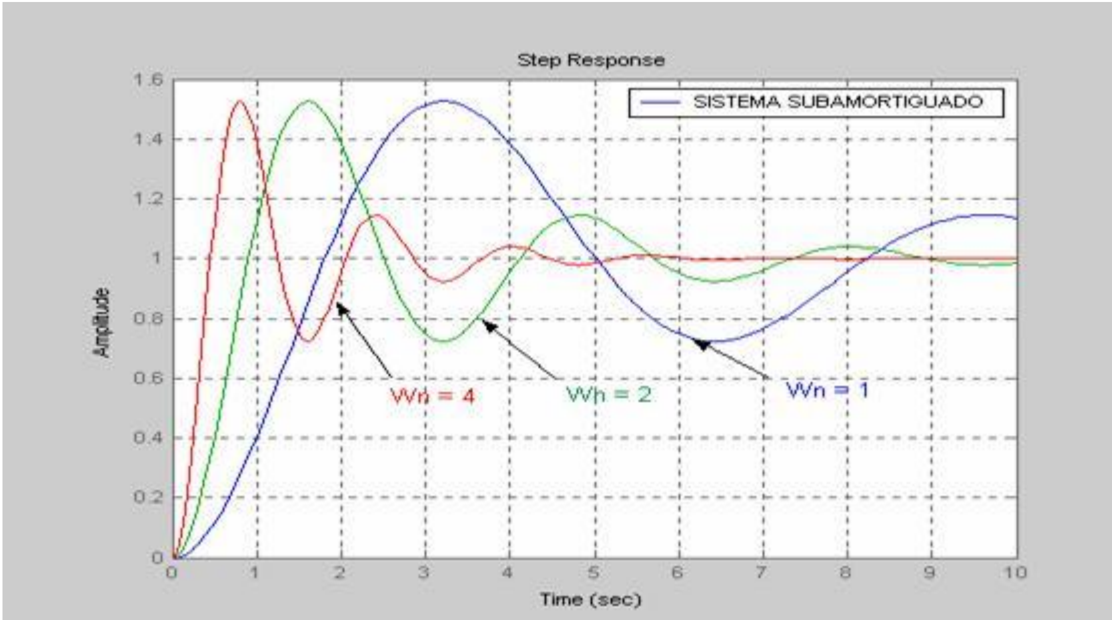
```
clear all
home
disp('  ');
disp('EJEMPLO 4-7: GRAFICA DE SISTEMAS AMORTIGUADOS');
disp('POR : M.I. JORGE ANTONIO POLANIA PUENTES');
disp('  ');
Wn = input('FRECUENCIA NATURAL NO AMORTIGUADA : Wn = ');
Zita = input('FACTOR DE AMORTIGUAMIENTO : Zita = ');
```

```
num = Wn^2;
den = [1 2*Zita*Wn Wn^2];
Gs = tf(num,den)
step(Gs,10)
grid
if Zita == 0
legend('SISTEMA OSCILATORIO')
elseif (Zita >0)&(Zita<1)
legend('SISTEMA SUBAMORTIGUADO')
elseif Zita == 1
legend('SISTEMA AMORTIGUADO')
else
legend('SISTEMA SOBREAMORTIGUADO')
end
hold on
```



EJEMPLO 4-7B :

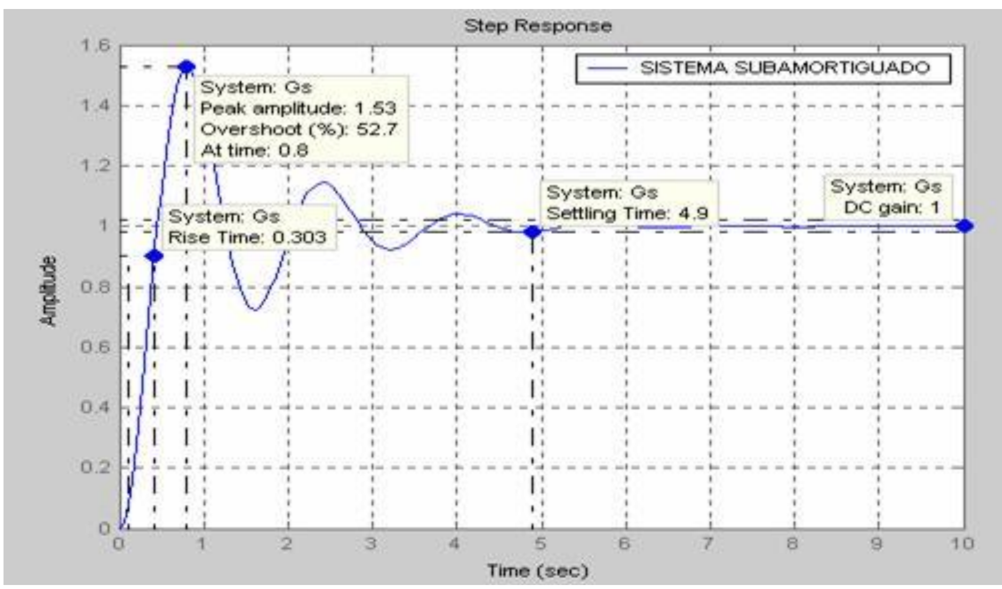
Graficar para un factor de amortiguamiento de ζ = 0.2 la respuesta al escalón unitario de un sistema con (a) wn = 1 rd/sg (b) wn = 2 rd/sg (c) wn = 4 rd/sg ¿Qué concluye ?



ESPECIFICACIONES

La respuesta transitoria a un escalón unitario de entrada tiene las siguientes especificaciones:

Tiempo de retardo : t_d
Tiempo de subida : t_r
Tiempo de pico : t_p
Tiempo de establecimiento : t_s
Máximo sobreimpulso : M_p



ECUACIONES

a) Tiempo de subida

Se calcula teniendo en cuenta que : $c(t_r) = 1$:

$$t_r = \frac{\pi - \theta}{w_d}, \quad \theta = tan^{-1}\left(\frac{w_d}{\sigma}\right) = tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}$$

b) Tiempo de pico

$$\frac{\partial c(t_p)}{\partial t} = 0$$

Se calcula teniendo en cuenta que :

c) Tiempo de asentamiento (setting) :

Se calcula teniendo en cuenta la tolerancia en el error :

$$t_s = \frac{3}{\zeta w_n} = \frac{3}{\sigma}$$

Si la tolerancia es del 5%

$$t_s = \frac{4}{\zeta w_n} = \frac{4}{\sigma}$$

Si la tolerancia es del 2%

d) Sobreimpulso (overshoot) :

Se calcula teniendo en cuenta que : $M_p = c\,(t_p) - 1$

$$M_p = e^{-\sigma \pi / w_d} = e^{-\pi \zeta / \sqrt{1-\zeta^2}}$$

e) Tiempo de muestreo :

Las respuestas temporales de un sistema continuo y de un sistema discreto difieren sustancialmente a medida que aumenta el tiempo de muestreo, dando una peor respuesta transitoria, afectando la estabilidad relativa del sistema y por tanto aumentando el error estacionario.

$$\#\,muestras/ciclo = \frac{w_d}{w_s}, \quad donde \ w_s = \frac{2\pi}{T}, \quad T = Periodo\ de\ muestreo$$

Se consideran suficientes para obtener una buena respuesta transitoria :

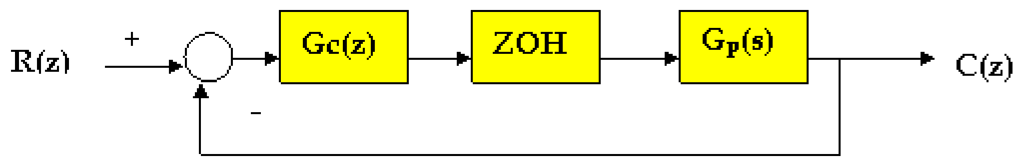
 # muestras / ciclo >= 10.

Si se conoce el ángulo de ubicación del polo dominante, el #muestras /ciclo se puede calcular de la siguiente forma :

$$\boxed{\#\, muestras/ciclo = \frac{360^o}{\theta}}$$

EJEMPLO 4-8 :

Analice el comportamiento de la respuesta transitoria del sistema de la figura cuando el tiempo de muestreo cambia de 0.5 sg a 1.0 sg. Encuentre para cada caso el # muestras / ciclo.



$$Para\ el\ controlador\ digital\ :\ Gc(z) = \frac{2z}{z-1} = \frac{2}{1-z^{-1}}$$

$$Para\ la\ planta\ continua\ :\ Gp(s) = \frac{1}{s+2}$$

Solución :

(a) Encontrar función de Transf. de la planta discretizada con ZOH, T = 0.5 sg

G(z) = (1 – z^-1)Z(Gp(s)/s) = (1-z^-1)Z(1 / s(s+2))

(b) Encontrar la función de transferencia en lazo cerrado

Gla(z) = G(z)*Gc (z)

$$Gla(z) = \frac{0.6321z}{z^2 - 1.368z + 0.3679}$$

$$Glc(z) = \frac{Gla(z)}{1 + Gla(z)}$$

$$Glc(z) = \frac{0.6321z}{z^2 - 0.7358z + 0.3679}$$

(c) Encontrar la ubicación de los polos del sistema

Ecuación característica es:
$z^2 - 0.7358z + 0.3679 = 0$

$p1,2 = 0.3679 \pm j0.4822$

$\theta = \tan^{-1}(0.4822 / 0.3679) = 52.7^o$

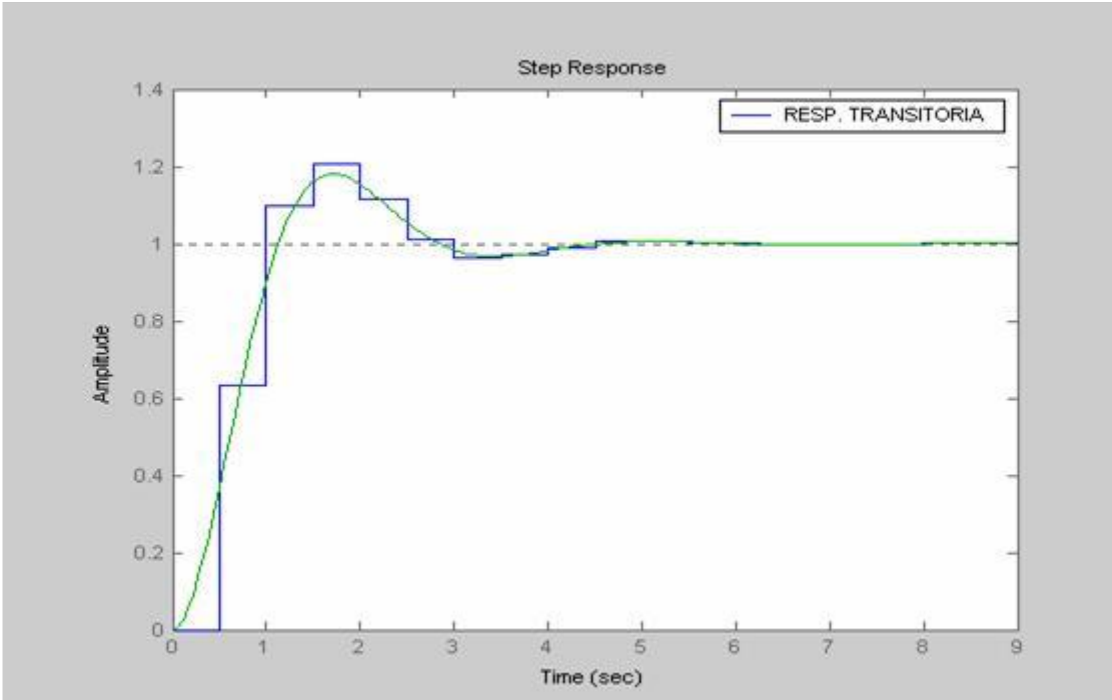
(d) Calcular # muestras / ciclo

 # muestras / ciclo = 360º / θ ***= 6.8***

(e) Encontrar las características del sistema de segundo orden

Wn = 2.1 rd / sg, Wd = 1.8 rd / sg, $\zeta = 0.48$

tr = 1.21 sg, tp = 1.71 sg ts = 4 sg, Mp = 18%

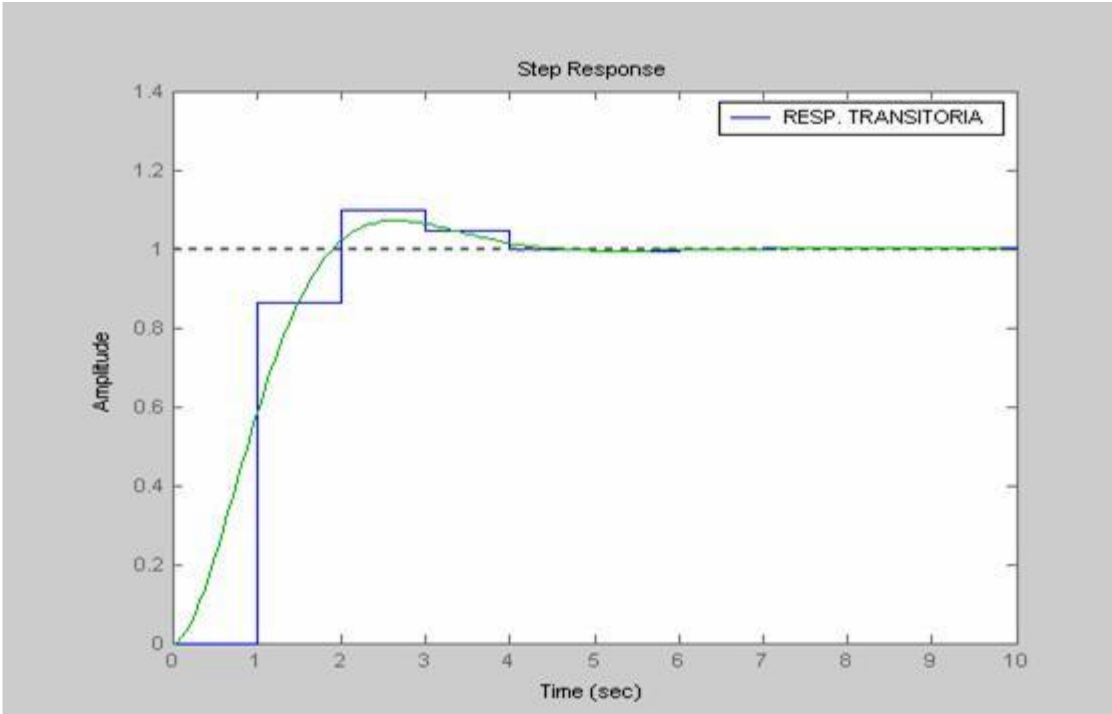


(f) <u>Repetir para T = 1 sg</u>

 # muestras / ciclo = 360º / θ =360º / 68.4º = 5.26

Wn = 2.1 rd / sg, Wd = 1.8 rd / sg, ζ = 0.48

tr = 1.63 sg, tp = 2.63 sg ts = 4 sg, Mp = 7%



Conclusión : ***Al aumentar el tiempo de muestreo disminuye el # muestras / ciclo y empeora su respuesta transitoria.***

Simulación :

```
% EJEMPLO 4-8 : RESPUESTA TRANSITORIA VS. #MUESTRAS/CICLO
% POR : M.I. JORGE ANTONIO POLANIA PUENTES
clc
disp('  ');
disp('EJEMPLO 4-8 : RESPUESTA TRANSITORIA VS. #MUESTRAS/CICLO ');
disp('  ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION : ');
disp('  ');

switch n
case 1
num = input('ENTRE NUMERADOR DE LA PLANTA : num = ');
den = input('ENTRE DENOMINADOR DE LA PLANTA : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den)
disp('  ');
```

```
case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K)

end

numzc = input('ENTRE NUMERADOR DEL CONTROLADOR : numzc = ');
denzc = input('ENTRE DENOMINADOR DEL CONTROLADOR : denzc = ');
SIGA =1;
while SIGA ==1
clc
disp('  ');
T = input('ENTRE TIEMPO DE MUESTREO : T = ');
Gz = c2d(Gp,T,'zoh');
disp('LA FUNCION DE TRANSF. DEL CONTROLADOR ES : ');
Gzc = tf(numzc, denzc,T)
disp('PARA SEGUIR OPRIMA ENTER');
pause

clc
disp('  ');
disp('LA FUNCION DE TRANSF. EN LAZO ABIERTO ES : ');
Gla = Gz*Gzc
disp('  ');
disp('LA FUNCION DE TRANSF. EN LAZO CERRADO ES : ');
Glc = feedback(Gla,1)
disp('PARA SEGUIR OPRIMA ENTER');
pause

clc
disp('  ');
disp('LOS POLOS DEL SISTEMA SON : ');
Polos = pole(Glc)
disp('QUE TIENEN MAGNITUDES Y ANGULOS DE :');
Mag = abs(Polos)
Ang1 = angle(Polos);
Ang = Ang1*180/pi
if (Mag(1)<1)&(Mag(2)<1)
disp('EL SISTEMA ES ESTABLE');
else
disp('EL SISTEMA ES INESTABLE');
end
disp('  ');
disp('PARA SEGUIR OPRIMA ENTER');
```

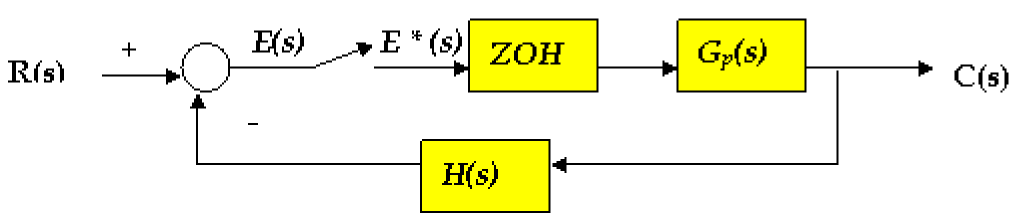
```
pause

clc
disp('  ');
disp('EL # MUESTRAS/CICLO = ');
2*pi/Ang1(1)
disp('LA RESPUESTA TRANSITORIA TIENE ');
disp('LAS SIGUIENTES CARACTERISTICAS');
[Wn,Zita] = damp(Glc);
Wd = Wn(1)*sqrt(1-Zita(1)^2);
tr = (pi-Ang1(1))/Wd
tp = pi/Wd
Mp = exp(-Zita(1)*Wn(1)*pi/Wd)
ts = 4/(Zita(1)*Wn(1))
nums = Wn(1)^2;
dens = [1 2*Zita(1)*Wn(1) Wn(1)^2];
Gs = tf(nums,dens);
SIGA = input (' PRESIONE 1 PARA SEGUIR ');
figure
step(Glc,Gs)
legend('RESP. TRANSITORIA')
end              % fin de while
```

4.4.2 ESTADO ESTACIONARIO

El desempeño de un sistema de control discreto se mide por su error en estado estacionario o permanente 'e_{ss}' y depende de la señal de entrada . Por el Teorema del valor final se tiene que :

$$e_{ss} = \lim_{k \to \infty} e(kT) = \lim_{z \to 1} \left[(1 - z^{-1}) E(z) \right] \qquad (1)$$

Para el siguiente sistema en lazo cerrado :



E(s) = R(s) – H(s) C(s), pero, C(s) = E(s)G_{ZOH}(s)Gp(s)*

$E(S) = R(S) – E^*(s)\, G_{ZOH}(s)Gp(s)H(s) \Rightarrow$

$E(z) = R(z) – E(z)\mathcal{Z}\, [G_{ZOH}(s)Gp(s)H(s)]$

$E(z) = R(z) – E(z)GH(z)$, por tanto,

$$\boxed{E(z) = \frac{R(z)}{1 + GH(z)}, \quad donde\ GH(z) = (1 - z^{-1})\mathcal{Z}\left[\frac{Gp(s)H(s)}{s}\right]}$$

Reemplazando en la ecuación (1) :

$$\boxed{e_{ss} = \lim_{z \to 1}\left[(1 - z^{-1}) \cdot \frac{R(z)}{1 + GH(z)}\right] \qquad (2)}$$

ENTRADA EN ESCALÓN UNITARIO

Si $r(t) = u(t)$, entonces,

$$R(z) = \frac{1}{1 - z^{-1}}$$

Reemplazando R(z) en (2) :

$$e_{ss} = \lim_{z \to 1}\left[\frac{1}{1 + GH(z)}\right], \quad si \quad \lim_{z \to 1} GH(z) = Kp,\ entonces,$$

$$\boxed{e_{ss} = \frac{1}{1 + Kp}, \Rightarrow e_{ss} \to 0, \Leftrightarrow Kp \to \infty}$$

Kp : constante de error de posición estática

Conclusión: ***Para que el error en estado estacionario tienda a cero se requiere que la constante de error de posición tienda a infinito, o sea que la función de transferencia en lazo abierto GH(z) tenga por lo menos un polo en z = 1.***

ENTRADA EN RAMPA UNITARIA

$$Si\ r(t) = t,\ entonces,\ R(z) = \frac{z^{-1}T}{(1 - z^{-1})^2}$$

$$e_{ss} = \lim_{z \to 1}\left[\frac{(1-z^{-1})(z^{-1}T)}{(1+GH(z))(1-z^{-1})^2}\right] = \lim_{z \to 1}\frac{Tz^{-1}}{(1+GH(z))(1-z^{-1})}$$

$$e_{ss} \to 0, \Leftrightarrow \lim_{z \to 1}\frac{(1+GH(z))(1-z^{-1})}{Tz^{-1}} \to \infty$$

$$\lim_{z \to 1}\frac{(1+GH(z))(1-z^{-1})}{Tz^{-1}} = \lim_{z \to 1}\frac{1-z^{-1}+GH(z)-GH(z)*z^{-1}}{T} \to \infty$$

$$\boxed{\lim_{z \to 1}\frac{(1-z^{-1})GH(z)}{T} = Kv \qquad e_{ss} = \frac{1}{Kv}}$$

Kv = Constante de error de velocidad estática

Conclusión: ***Para que el error en estado estacionario tienda a cero se requiere que la constante de error de velocidad tienda a infinito, o sea que la función de transferencia en lazo abierto GH(z) tenga un polo doble en z = 1.***

EJEMPLO 4-9 :

Para el sistema dado en el Ejemplo 4-8, encontrar la constante de error de posición Kp y la constante de error de velocidad Kv. (T = 0.5 sg)

Solución :

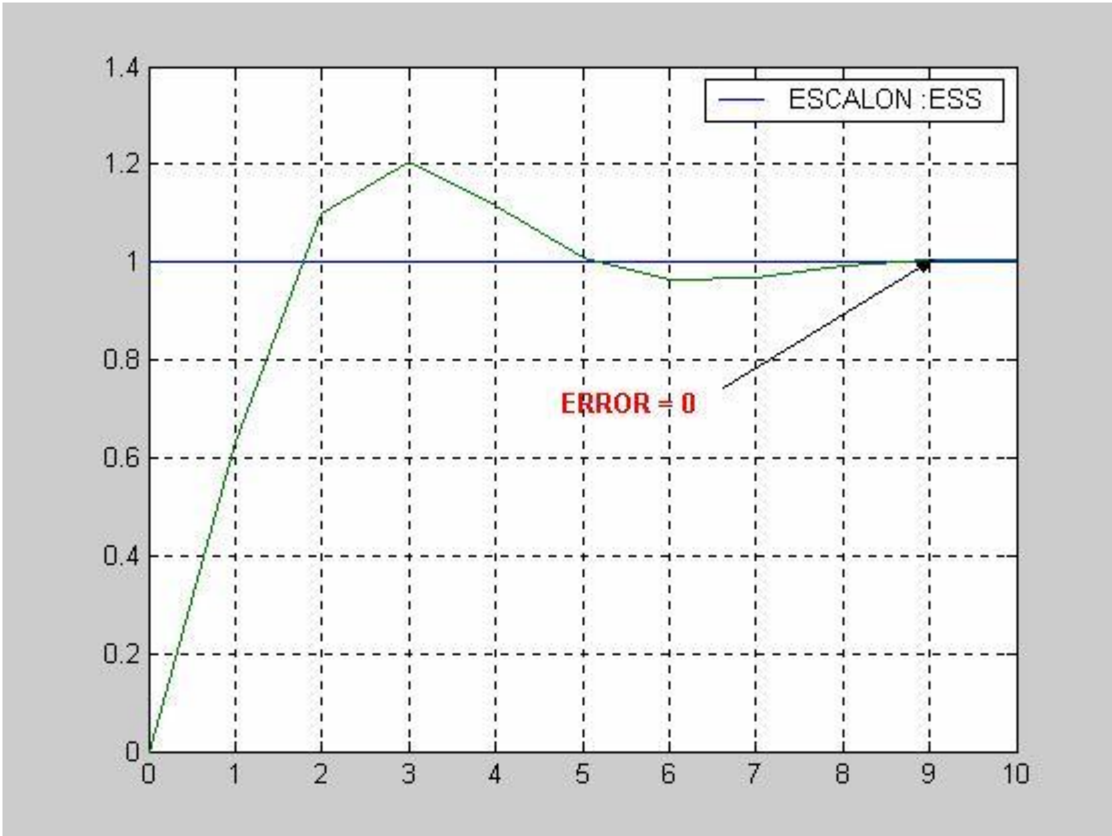
(a) Determinar Kp

$$Gla(z) = \frac{0.6321z}{z^2 - 1.368z + 0.3679} = \frac{0.6321z}{(z-1)(z-0.3679)}$$

$$Kp = \lim_{z \to 1} GH(z) = \lim_{z \to 1} Gla(z) \text{ , entonces,}$$

$$Kp = \lim_{z \to 1}\frac{0.6321z}{z^2 - 1.368z + 0.3679} = \frac{0.6321}{1 - 1.368 + 0.3679} = \infty$$

e_{ss} = 1 /(1+Kp) = 0 (entrada en escalón unitario)

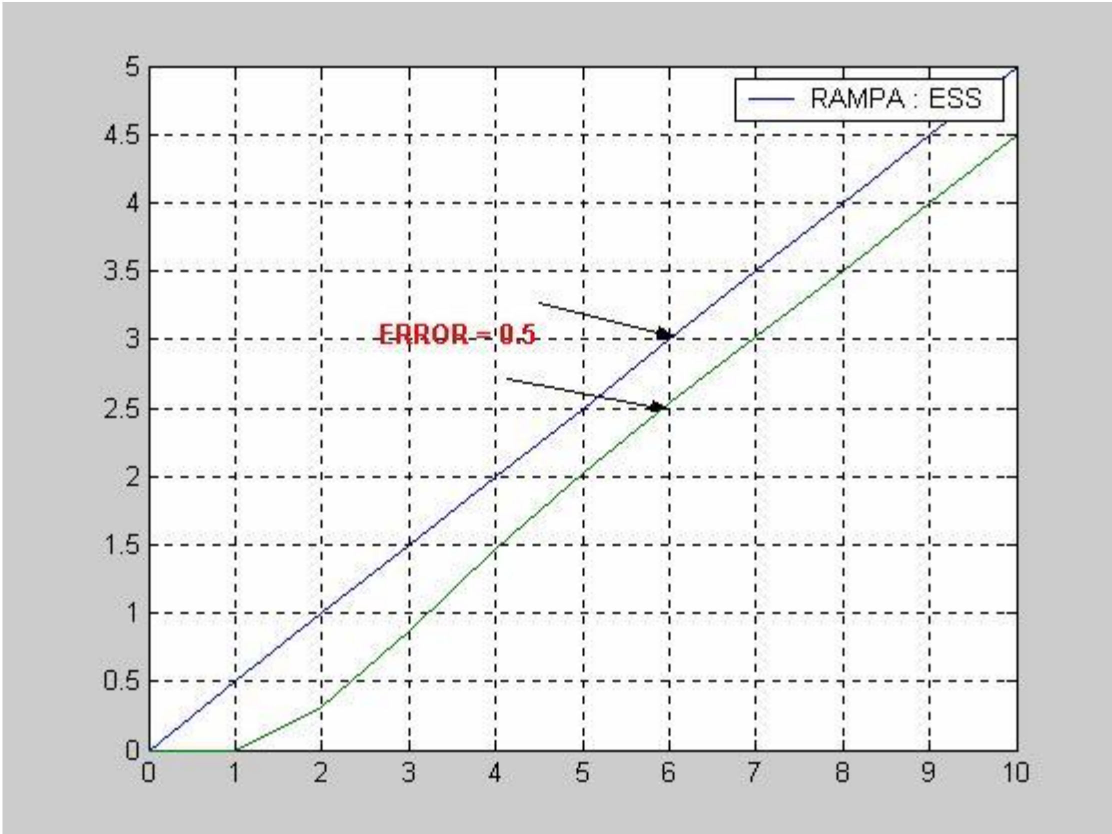


(b) Determinar Kv

$$Kv = \lim_{z \to 1} \frac{(1-z^{-1})Gla(z)}{T} = \lim_{z \to 1} \frac{(z-1)}{Tz} \cdot \frac{0.6321z}{(z-1)(z-0.3679)}$$

$$Kv = \lim_{z \to 1} \frac{(z-1)}{Tz} \cdot \frac{0.6321z}{(z-1)(z-0.3679)} = \frac{0.6321}{0.5(0.6321)} = 2$$

e_{ss} = 1 / Kv = 0.5 sg^{-1} (entrada rampa unitaria)



Simulación :

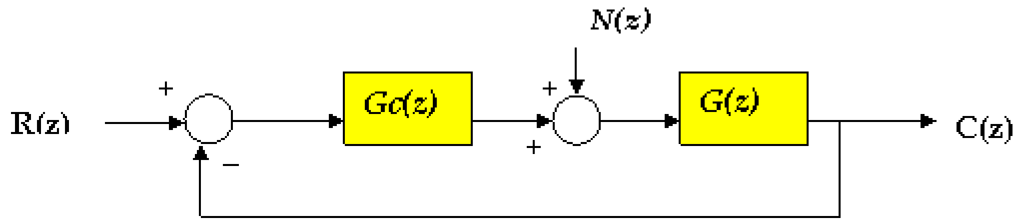
```
% EJEMPLO 4-9 : ERROR EN ESTADO ESTACIONARIO
% POR : M.I. JORGE ANTONIO POLANIA PUENTES
clc
disp('  ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION : ');
disp('  ');

switch n
case 1
num = input('ENTRE NUMERADOR DE LA PLANTA : num = ');
den = input('ENTRE DENOMINADOR DE LA PLANTA : den = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den)
disp('  ');

case 2
Z = input('Entre vector de ceros : Z = ');
```

```
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K)

end

numzc = input('ENTRE NUMERADOR DEL CONTROLADOR : numzc = ');
denzc = input('ENTRE DENOMINADOR DEL CONTROLADOR : denzc = ');
SIGA =1;
while SIGA ==1
clc
disp('  ');
T = input('ENTRE TIEMPO DE MUESTREO : T = ');
Gz = c2d(Gp,T,'zoh');
disp('LA FUNCION DE TRANSF. DEL CONTROLADOR ES : ');
Gzc = tf(numzc, denzc,T)
disp('PARA SEGUIR OPRIMA ENTER');
pause

clc
disp('  ');
disp('LA FUNCION DE TRANSF. EN LAZO ABIERTO ES : ');
Gla = Gz*Gzc
disp('  ');
% (A) OBTENCION DE Kp

disp('PARA SEGUIR OPRIMA ENTER');
pause
clc
disp('  ');

disp('(A) OBTENCION DE Kp');
[numa,dena] = tfdata(Gla,'v');
syms z
Glasym = poly2sym(numa,'z') / poly2sym(dena,'z');
Kp = limit(Glasym,z,1);
if Kp==NaN
disp('EL ess A LA ENTRADA DEL ESCALON ES : ess = ');
ess = 0;
numeric(ess)
else
disp('EL ess A LA ENTRADA DEL ESCALON ES : ess = ');
ess = 1/(1+Kp);
numeric(ess)
end
figure(1)
```

```
Glc = feedback(Gla,1);
k = 0:10;
x = ones(1,11);
c1 = lsim(Glc,x);
plot(k,x,k,c1)
grid
legend(' ESCALON :ESS')

% (B) OBTENCION DE Kv

disp('PARA SEGUIR OPRIMA ENTER');
pause

clc
disp('  ');

disp('(B) OBTENCION DE Kv');
Kv = limit((1-z^(-1))*Glasym/T,z,1);
if Kv==NaN
disp('EL ess A LA ENTRADA DE LA RAMPA ES : ess = ');
ess = 0;
numeric(ess)
else
disp('EL ess A LA ENTRADA DE LA RAMPA ES : ess = ');
ess = 1/Kv;
numeric(ess)
end
figure(2)
k = 0:10;
x = k*T;
c2 = lsim(Glc,x);
plot(k,x,k,c2)
grid
legend('RAMPA : ESS')
SIGA = input (' PRESIONE 1 PARA SEGUIR ');
end             % fin de while
```

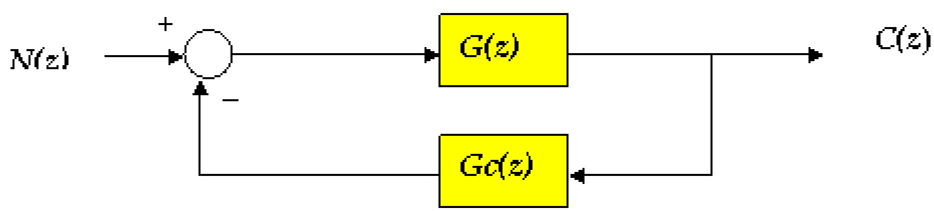
4.4.3 PERTURBACIONES

Las perturbaciones es otra entrada externa al sistema que influyen en la respuesta transitoria y en el estado estacionario. Para analizar el efecto que produce en una salida una perturbación, se hace la entrada al sistema igual a cero. Por ejemplo :



N(z) : Perturbación o ruido

Haciendo R(z) = 0 , el sistema se convierte en :



E(z) = R(z) – C(z) = - C(z)

$$\frac{C(z)}{N(z)} = \frac{G(z)}{1 + Gc(z)G(z)} \Rightarrow E(z) = -\frac{N(z)G(z)}{1 + Gc(z)G(z)}$$

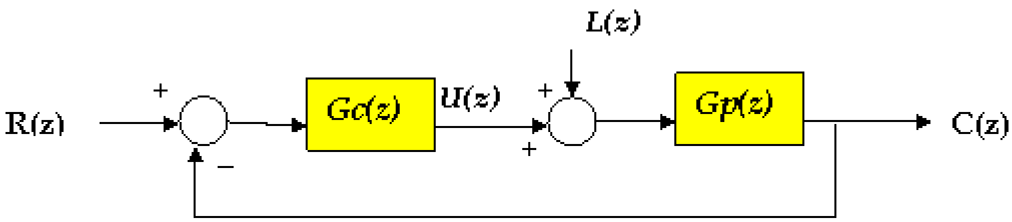
$$e_{ss} = \lim_{z \to 1} (1 - z^{-1})E(z) = \lim_{z \to 1} \left[\frac{-(1 - z^{-1})N(z)G(z)}{1 + Gc(z)G(z)} \right]$$

Si la perturbación es constante (escalón)   n(t) = N, se tiene :

$$N(z) = \frac{N}{1 - z^{-1}}, \quad e_{ss} = -\lim_{z \to 1} \left[\frac{NG(z)}{1 + Gc(z)G(z)} \right]$$

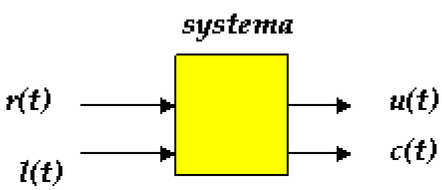
EJEMPLO 4-10 :

Encontrar las funciones de transferencia de la señal de control U(z) y de la salida C(z) con respecto a la entrada (set point) R(z) y a la perturbación en la carga L(z), (T = 2 sg).



Controlador ***Planta***

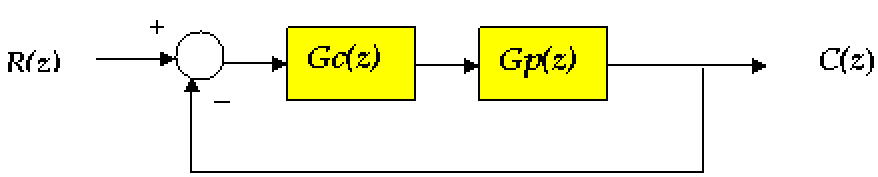
$$Gp(s) = e^{1.5s}\,\frac{1}{6s+1} \Rightarrow G(z) = \frac{0.07996z + 0.2035}{z^2 - 0.7165z}$$



$$Gc(z) = \frac{z - 0.5}{z - 1} \quad (controlad\ or\ PI)$$

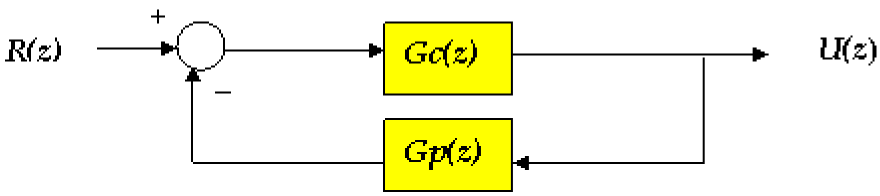
Solución :

(a) $G_{CR}(z)$: $l(t) = 0$



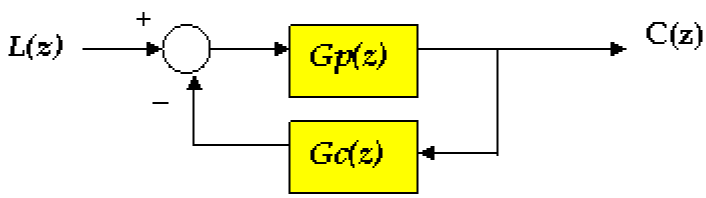
$$G_{CR}(z) = \frac{C(z)}{R(z)} = \frac{Gc(z)Gp(z)}{1 + Gc(z)Gp(z)} = \frac{0.07996z^2 + 0.1635z - 0.1018}{z^3 - 1.637z^2 + 0.8801z - 0.1018}$$

(b) $Gu_R(z)$: $l(t) = 0$



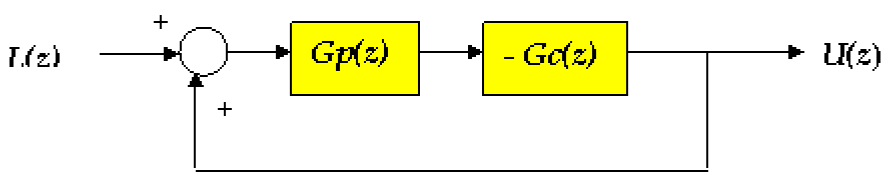
$$G_{UR}(z) = \frac{U(z)}{R(z)} = \frac{Gc(z)}{1 + Gc(z)Gp(z)} = \frac{z^3 - 1.217z^2 + 0.3583z}{z^3 - 1.637z^2 + 0.8801z - 0.1018}$$

(c) $G_{CL}(z)$: $r(t) = 0$

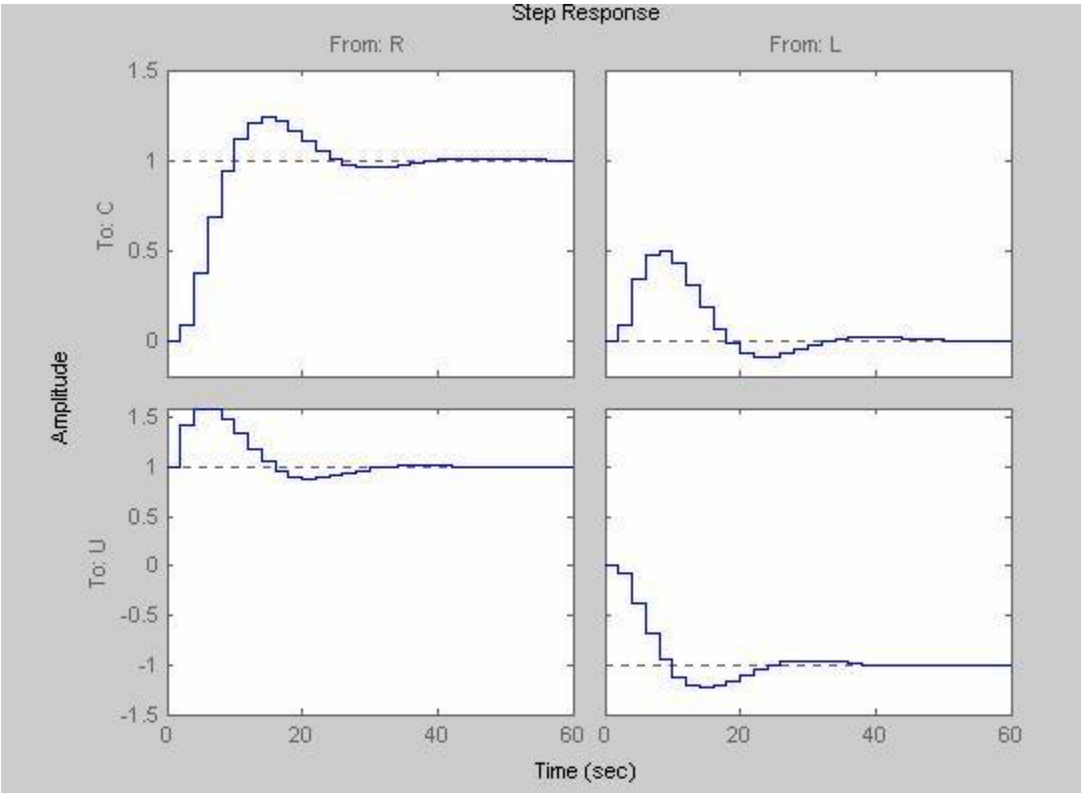


$$G_{CL}(z) = \frac{C(z)}{L(z)} = \frac{Gp(z)}{1 + Gc(z)Gp(z)} = \frac{0.07996z^2 + 0.1236z - 0.2035}{z^3 - 1.637z^2 + 0.8801z - 0.1018}$$

(d) $G_{UL}(z)$: $r(t) = 0$



$$G_{UL}(z) = \frac{U(z)}{L(z)} = \frac{-Gp(z)Gc(z)}{1+Gp(z)Gc(z)} = \frac{-0.07996z^2 - 0.1635z + 0.1018}{z^3 - 1.637z^2 + 0.8801z - 0.1018}$$



(e) Calcular el error de la perturbación sobre la salida c(t)

E(z) = R(z) – C(z) = -C(z)

$$E(z) = -G_{CL}(z) * L(z) = -\frac{0.07996z^2 + 0.1236z - 0.2035}{z^3 - 1.637z^2 + 0.8801z - 0.1018} * L(z)$$

$$G_{CL}(z) = \frac{C(z)}{L(z)} = \frac{Gp(z)}{1+Gc(z)Gp(z)} = \frac{0.07996z^2 + 0.1236z - 0.2035}{z^3 - 1.637z^2 + 0.8801z - 0.1018}$$

Si la perturbación es un escalón unitario, entonces,

$$L(z) = \frac{1}{1 - z^{-1}}$$

$$e_{ss} = \lim_{z \to 1}\left[(1 - z^{-1})E(z)\right] = -\lim_{z \to 1}\left[\frac{0.07996z^2 + 0.1236z - 0.2035}{z^3 - 1.637z^2 + 0.8801z - 0.1018}\right] = 0$$

5. ANÁLISIS : LGR

5.1 INTRODUCCIÓN

El Lugar Geométrico de las Raíces es un enfoque gráfico muy utilizado par investigar los efectos de la ganancia del sistema y del periodo de muestreo sobre la estabilidad de un sistema en lazo cerrado conforme los polos y ceros del lazo abierto se mueven.

Representando la ecuación característica como :

1 + F(z) = 0

se define la condición de magnitud como :

$$F(z) = -1, \text{ entonces, } |F(z)| = 1$$

y la condición de ángulo como :

$$F(z) = -1, \text{ entonces, ángulo}(F(z)) = \pm 180^{o}(2k+1), \text{ para } k = 0,1\ 2,3,\ldots$$

Tanto la condición de magnitud como de ángulo deben cumplirse para las raíces de la ecuación característica o polos del sistema en lazo cerrado.

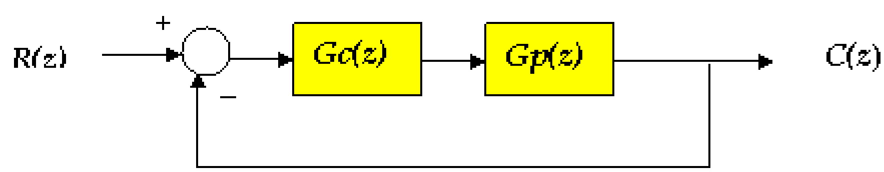
$$si \quad F(z) = K\frac{B(z)}{A(z)}, \Rightarrow 1 + K\frac{B(z)}{A(z)} = 0 \Rightarrow K = -\frac{A(z)}{B(z)}$$

La gráfica de los puntos en el plano complejo que satisfacen la condición de ángulo son los lugares geométricos de las raíces o de los polos en lazo cerrado conforme la ganancia K aumenta de cero a infinito. Cuando K = 0, los polos en lazo cerrado son los polos del sistema en lazo abierto.

5.2 VARIACIÓN DE LA GANANCIA

EJEMPLO 5-1A :

Obtener el lugar geométrico de las raíces del sistema en lazo cerrado mostrado en la figura, para (a) K = 0 , (b) K = 0:0.01:0.1, (c) K = 0:0.01:1, (d) K = 0:0.01:10



Solución :

$$Gp(s) = \frac{1}{s(s+1)}, \quad T = 1\,sg, \quad Gc(z) = \frac{1.4\,z^2 - 1.4\,z + 0.2}{z^2 - z},$$

$$Gla(z) = \frac{0.515\,z^3 - 0.1451\,z^2 - 0.2964z + 0.05285}{z^4 - 2.368\,z^3 + 1.736z^2 - 0.3679z} = \frac{numz}{denz}$$

La ecuación característica es igual a :

$$1 + K\frac{numz}{denz} = 0 \Rightarrow denz + K*numz = 0$$

Simulación :

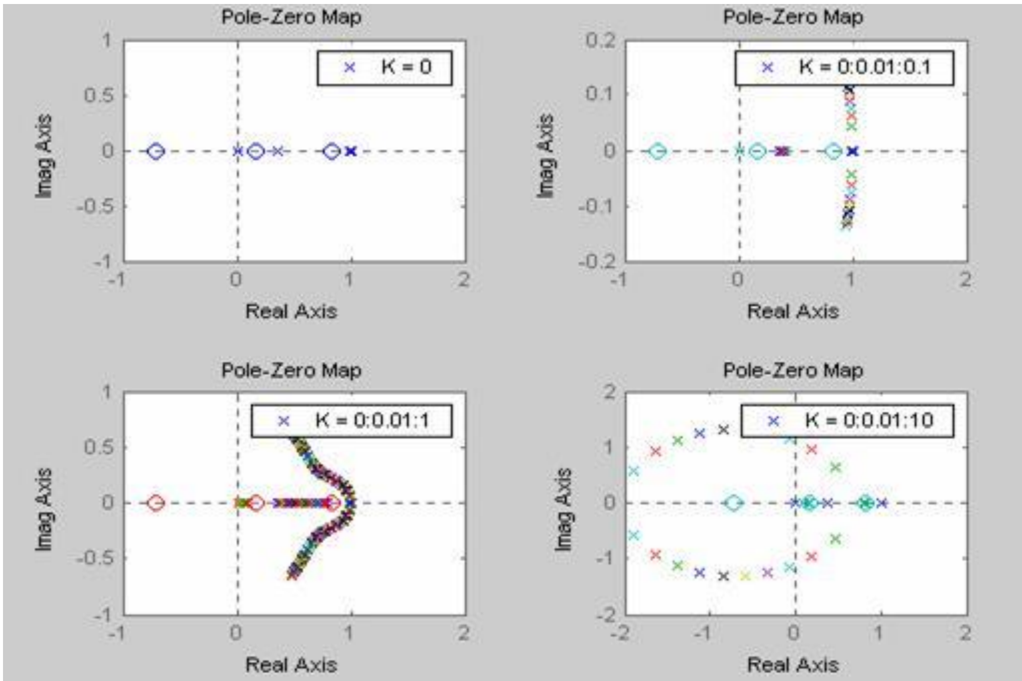
PROGRAMA EN MATLAB :

```
% EJEMPLO 5-1 : LUGAR GEOMETRICO DE LAS RAICES
% POR : M.I. JORGE ANTONIO POLANIA PUENTES
clc
disp('  ');
disp('EJEMPLO 5-1 : LUGAR GEOMETRICO DE LAS RAICES ');
disp('  ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION : ');
disp('  ');

switch n
case 1
num = input('ENTRE NUMERADOR DE LA PLANTA : num = ');
den = input('ENTRE DENOMINADOR DE LA PLANTA : den = ');
Ret =input('RETARDO DE LA ENTRADA : inputDelay = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den,'inputDelay',Ret)
disp('  ');

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
Ret =input('RETARDO DE LA ENTRADA : inputDelay = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K,'inputDelay',Ret)
end

clc
disp('  ');
numzc = input('ENTRE NUMERADOR DEL CONTROLADOR : numzc = ');
denzc = input('ENTRE DENOMINADOR DEL CONTROLADOR : denzc = ');
T = input('ENTRE TIEMPO DE MUESTREO : T = ');
Gc = tf(numzc,denzc,T)
Gpz = c2d(Gp,T,'zoh');
disp('PRESIONE TECLA ENTER');
pause
clc
disp('  ');
Gla = Gpz*Gc
```

```
[numz denz] = tfdata(Gla,'v');

disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA GRAFICA USANDO ECUAC. CARACT. ');
disp(' 2: PARA GRAFICA USANDO RLOCUS ');
disp(' 3: DETERMINAR K CRITICO ' );
m = input('SELECCIONE LA OPCION : ');
disp('  ');

switch m
case 1
%'GRAFICA DE LOS POLOS EN LAZO ABIERTO K = 0 '
subplot(221)
K = 0;
numzc = numz;
denzc = denz + K*numz;
Glc = tf(numzc,denzc,T);
pzmap(Glc)
legend('K = 0 ');
%'GRAFICA DE LOS POLOS EN LAZO CERRADO K = 0:0.1 '
subplot(222)
for K = 0:0.01:0.1
numzc = numz;
denzc = denz + K*numz;
Glc = tf(numzc,denzc,T);
pzmap(Glc)
legend('K = 0:0.01:0.1 ');
hold on
end
%'GRAFICA DE LOS POLOS EN LAZO CERRADO K = 0:1 '
subplot(223)
for K = 0:0.01:1
numzc = numz;
denzc = denz + K*numz;
Glc = tf(numzc,denzc,T);
pzmap(Glc)
legend('K = 0:0.01:1 ');
hold on
end
%'GRAFICA DE LOS POLOS EN LAZO CERRADO K = 0:10 '
subplot(224)
for K = 0:01:10
numzc = numz;
denzc = denz + K*numz;
Glc = tf(numzc,denzc,T);
pzmap(Glc)
legend('K = 0:0.01:10 ');
```

```
hold on
end
hold off

case 2
% EJEMPLO 5-1B : UTILIZANDO RLOCUS
%'GRAFICA DE LOS POLOS EN LAZO CERRADO K = 0:0.1 '
subplot(221)
K = 0:0.01:0.1;
rlocus(Gla,K)
legend('K = 0:0.01:0.1 ');
hold off
%'GRAFICA DE LOS POLOS EN LAZO ABIERTO K = 0:1 '
subplot(222)
K = 0:0.01:1;
rlocus(Gla,K)
legend('K = 0:0.01:1 ');
hold off
%'GRAFICA DE LOS POLOS EN LAZO ABIERTO K = 0:10 '
subplot(223)
K = 0:0.01:10;
rlocus(Gla,K)
legend('K = 0:0.01:10 ');
%'GRAFICA DE LOS POLOS EN LAZO ABIERTO K = 0:INFINITO '
subplot(223)
rlocus(Gla)
legend('K = 0:INFINITO ');

case 3
% EJEMPLO 5-1C : DETERMINAR K CRITICO
disp(' METODO GRAFICO ');
hold off
rlocus(Gla)
hold on
% Graficar circulo unitario
r = 0:0.01:2*pi;
x = sin(r);
y = cos(r);
plot(x,y)
[K,P] = rlocfind(Gla);
Mag = abs(P);
disp('LA GANANCIA CRITICA ES IGUAL A : ');
display(K);
disp('LA MAGNITUD DEL POLO SELECCIONADO ES : ');
display(Mag(1));
end
```

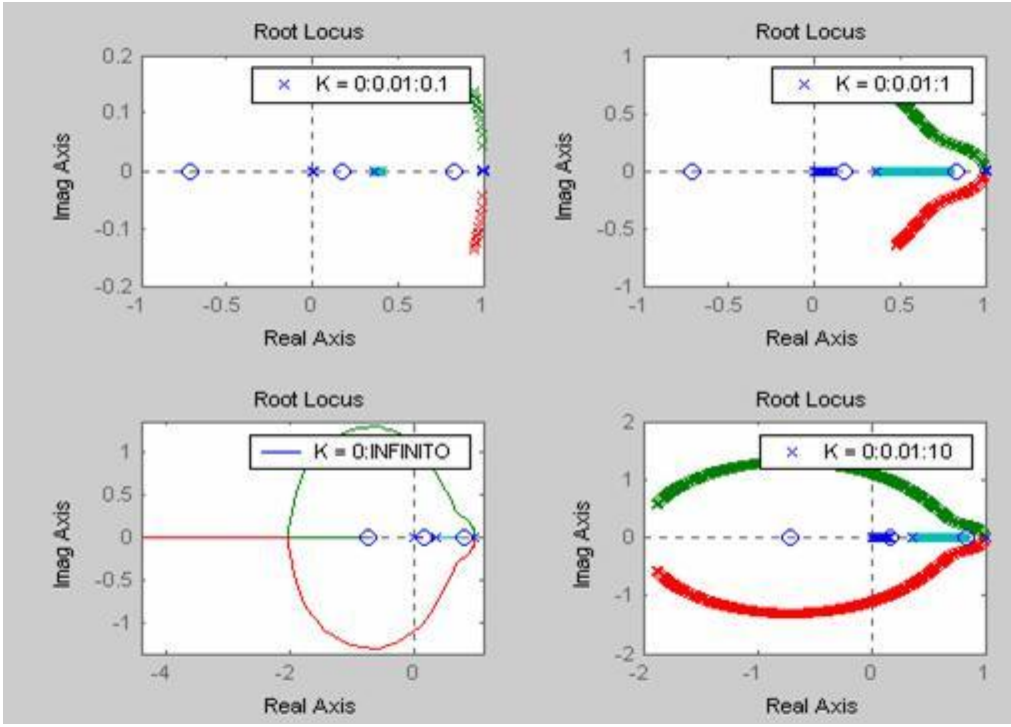
Las gráficas obtenidas son las siguientes:



5.3 USO DE RLOCUS

EJEMPLO 5-1B :

Obtener las gráficas anteriores usando el comando Matlab rlocus.



Simulación :

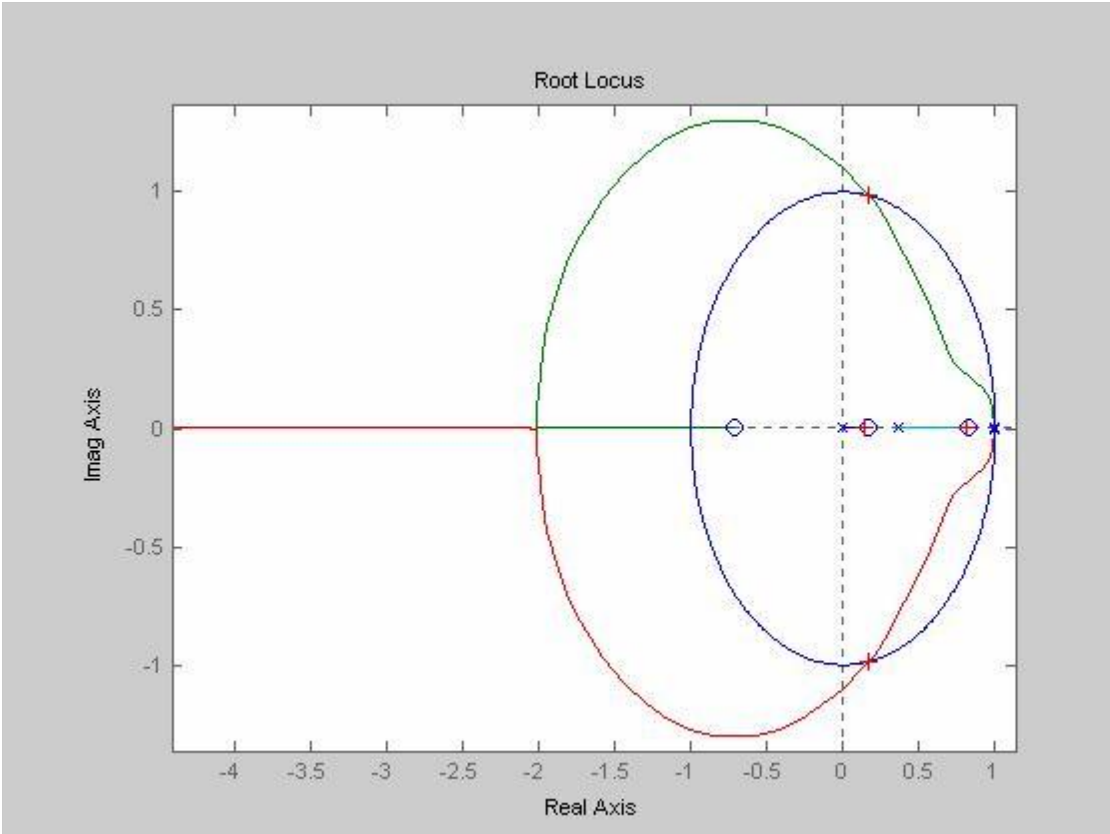
Se encuentra en la sección CAP5_2

5.4 USO DE RLOCFIND

EJEMPLO 5-1C :

Para este ejemplo determinar el valor máximo de K para el cual el sistema en lazo cerrado sea estable.

$$\left|K\,Gla(z)\right| = 1 \Rightarrow \left| K\,\frac{0.515\,z^{3} - 0.1451\,z^{2} - 0.2964z + 0.05285}{z^{4} - 2.368\,z^{3} + 1.736z^{2} - 0.3679z} \right| = 1$$



La ganancia crítica corresponde al punto de corte entre la gráfica del lugar de las raíces y el círculo unitario. Matlab nos da un valor de K = 2.06 en el polo:

z = 0.1793 + 0.9780i

EJEMPLO 5-2 :

Calcular la ganancia crítica del ejemplo anterior para : (a) T = 0.5 seg, T = 1.0 seg, T = 1.5 seg (b) Obtener la respuesta transitoria (c) demostrar que al aumentar T con ganancia constante K = 1.5 el número de muestras/ciclo disminuye.

Simulación:

```
% EJEMPLO 5-2 : LUGAR GEOMETRICO DE LAS RAICES
% POR : M.I. JORGE ANTONIO POLANIA PUENTES
clc
disp('  ');
disp('EJEMPLO 5-1 : LUGAR GEOMETRICO DE LAS RAICES ');
disp('  ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
```

```
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION : ');
disp('  ');

switch n
case 1
num = input('ENTRE NUMERADOR DE LA PLANTA : num = ');
den = input('ENTRE DENOMINADOR DE LA PLANTA : den = ');
Ret =input('RETARDO DE LA ENTRADA : inputDelay = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den,'inputDelay',Ret)
disp('  ');

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
Ret =input('RETARDO DE LA ENTRADA : inputDelay = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K,'inputDelay',Ret)
end

clc
disp('  ');
numzc = input('ENTRE NUMERADOR DEL CONTROLADOR : numzc = ');
denzc = input('ENTRE DENOMINADOR DEL CONTROLADOR : denzc = ');
SIGA = 1;
while SIGA ==1
T = input('ENTRE TIEMPO DE MUESTREO : T = ');
Gc = tf(numzc,denzc,T)
Gpz = c2d(Gp,T,'zoh');
disp('PRESIONE TECLA ENTER');
pause
clc
disp('  ');
Gla = Gpz*Gc
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA OBTENER Kc ');
disp(' 2: PARA OBTENER RESP. TRANSITORIA ');
disp(' 3: PARA OBTENER MUESTRAS/CICLO ' );
m = input('SELECCIONE LA OPCION : ');
disp('  ');
switch m
case 1
% (a) Obtencion de Kc
disp('ESPERE GLR Y POSICIONE CURSOR EN EL PUNTO DE CORTE ');
hold off
```

```
rlocus(Gla)
hold on
r = 0:0.01:2*pi;
x = sin(r);
y = cos(r);
plot(x,y)
[K,P] = rlocfind(Gla);
Mag = abs(P);
disp('LA GANANCIA CRITICA ES IGUAL A : ');
display(K);
case 2
% (b) Respuesta transitoria K = 1
Glc = feedback(Gla,1);
hold off
step(Glc)
case 3
% (c) Muestras/ciclo
K = 1.5;
Gla1 = K*Gla;
Glc = feedback(Gla1,1);
PoloDominante = pole(Glc)
teta = angle(PoloDominante)
Angulo = teta*180/pi
MuestrasCiclo = 360/Angulo
end             % fin de swich
SIGA = input('TECLEE 1 PARA SEGUIR : ');
end
```

Respuesta :

(a)

Para T = 0.5 sg se obtiene una ganancia crítica Kc = 3.6
Para T = 1.0 sg se obtiene una ganancia crítica Kc = 2.1
Para T = 1.5 sg se obtiene una ganancia crítica Kc = 1.5

Conclusión: Al aumentar el periodo de muestreo disminuye la ganancia crítica.

(b)

Al aumentar el periodo de muestreo la respuesta transitoria se va empeorando tendiendo hacia la inestabilidad

(c)

Para T = 0.5 sg se obtiene Muestras/ciclo = 10.8
Para T = 1.0 sg se obtiene Muestras/ciclo = 5.3
Para T = 1.5 sg se obtiene Muestras/ciclo = 3.7

***Conclusión**: Al aumentar el periodo de muestreo disminuye el número de muestras/ciclo tendiendo hacia la inestabilidad.*
Se sugiere en diseño un valor de 10 muestras/ciclo.

EJEMPLO 5-3 :

Hacer un programa tal que conocida la función de transferencia en lazo abierto de un sistema de control se encuentre el valor de la ganancia crítica en forma analítica.

PROGRAMA EN MATLAB :

```
% EJEMPLO 5-3 : HALLAR Kc ANALITICAMENTE
clc
clear
disp('  ');
disp(' EJEMPLO 5-3 : HALLAR Kc ANALITICAMENTE ');
disp('  ');
disp('SELECCIONE PRESIONANDO :');
disp(' 1: PARA FUNCION DE LA PLANTA EN TF ');
disp(' 2: PARA FUNCION DE LA PLANTA EN ZPK ');
n=input('SELECCIONE LA OPCION : ');
disp('  ');

switch n
case 1
num = input('ENTRE NUMERADOR DE LA PLANTA : num = ');
den = input('ENTRE DENOMINADOR DE LA PLANTA : den = ');
Ret =input('RETARDO DE LA ENTRADA : inputDelay = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = tf(num,den,'inputDelay',Ret)
disp('   ');

case 2
Z = input('Entre vector de ceros : Z = ');
P = input('Entre vector de polos : P = ');
K = input('Ganancia es igual a : K = ');
Ret =input('RETARDO DE LA ENTRADA : inputDelay = ');
disp('LA FUNCION DE TRANSFERENCIA DE LA PLANTA ES : Gp(s) = ');
Gp = zpk(Z,P,K,'inputDelay',Ret)
end

clc
disp('  ');
numzc = input('ENTRE NUMERADOR DEL CONTROLADOR : numzc = ');
```

```
denzc = input('ENTRE DENOMINADOR DEL CONTROLADOR : denzc = ');
SIGA = 1;
while SIGA ==1
T = input('ENTRE TIEMPO DE MUESTREO : T = ');
Gc = tf(numzc,denzc,T)
Gpz = c2d(Gp,T,'zoh');
disp('PRESIONE TECLA ENTER');
pause
clc
disp('  ');
Gla = Gpz*Gc
[numz,denz]=tfdata(Gla,'v');
syms K
EcuacionCaract = poly2sym(denz,'z') + K*poly2sym(numz,'z')
z = input('ESCRIBIR solve(EcuacionCaract=0,z : ');
j=1;
for K= 0:0.1:10
% z=solve('z^2-1.6065*z+0.6065+K*0.3935*z=0','z');
z1=eval(z(1));
z2=eval(z(2));
i=1;
a(i,j) = K;
a(i+1,j)= abs(z1)
a(i+2,j)= abs(z2)
j =j+1;
end                     % fin de for
% El Kc se encuentra cuando abs(z) aprox. 1
% pruebe inicialmente un delta = 0.1
delta = input('ENTRE DELTA DE ABS(Z): delta = ');
[i,j]=find(a<(1+delta)&a>(1-delta))
Kc = a(1,j)
n=length(Kc);
Kc=a(1,j(n))
SIGA = input('TECLEE 1 PARA SEGUIR : ');
end              % fin de while
```

6. ANÁLISIS : DOMINIO DEL TIEMPO

6.1 SOBREIMPULSO Y TIEMPO DE PICO

Aunque en el mundo real es muy raro encontrar un verdadero sistema de segundo orden, la mayor parte de parámetros de desempeño están definidos

con respecto a este modelo. Muchos sistemas de orden superior pueden aproximarse a uno de segundo orden si todos los polos de orden superior están localizados de modo que su contribución a la respuesta transitoria es despreciable.

Algunas de las raíces dada su ubicación en el plano-z afectan más la respuesta del sistema que otras. Estas raíces se denominan dominantes. Las raíces dominantes están dentro el círculo unitario y más próximas a éste. Las raíces cercanas al origen son de menor importancia ya que para $z = 0$, $s \to \infty$. El eje real negativo debe evitarse pues su respuesta en el tiempo es oscilatoria.

Las raíces no significantes no pueden descartarse, afectan el desempeño de estado estacionario del sistema. Una manera de simplificar un sistema de orden superior con polos no significativos en la función de transferencia de lazo cerrado es sustituirlos por polos en
$z = 0$.

Un sistema continuo sin ceros al aplicarse la transformada-Z produce en el sistema digital al menos un cero, por eso se puede especificar un sistema de control digital prototipo de segundo orden como:

$$\frac{C(z)}{R(z)} = \frac{K(z - z_1)}{(z - p1)(z - p1^*)}, \text{ donde } p1 \text{ y } p1^* \text{ son polos conjugados}$$

Si se tiene una entrada escalón unitario, $R(z) = z/(z-1)$, entonces,

$$C(z) = \frac{Kz(z - z_1)}{(z - 1)(z - p1)(z - p1^*)}$$

$\zeta = -\cos\theta$, donde

θ = angle(ln (p1))   Wn = abs(ln(p1))

Mp = exp(- ζ □/sqrt(1- ζ^2)

tp = □/Wn*sqrt(1- ζ^2)

EJEMPLO 6-1:

La función de transferencia (planta) de un vehículo espacial es:

$$Gp(s) = \frac{39.45K}{s^2 + 8.871s}$$

Encontrar para el sistema digital con T = 0.1 sg , T = 0.6 y T = 1.0 sg :

- **(a)** ***Factor de amortiguamiento ζ y frecuencia natural Wn***
- **(b)** ***Gráfica de la secuencia de salida c(kT) si la entrada es un escalón unitario***
- **(c)** ***El tiempo de pico: tp, el error de estado estacionario: ess y el máximo sobreimpulso: Mp***
- **(d)** ***La respuesta al paso digital y continuo.***

Simulación:

PROGRAMA EN MATLAB :

```
% EJEMPLO 6-1: HALLAR zita, Wn, tp, Mp
clear
disp('EJEMPLO 6-1 : HALLAR zita, Wn, tp, Mp ');
nump = input('ENTRE NUMERADOR : nump = ');
denp = input('ENTRE DENOMINADOR : denp = ');
Gp = tf(nump,denp)
SIGA = 1;
while SIGA ==1
T = input('TIEMPO DE MUESTREO : T = ');
K = input('VALOR DE GANANCIA: K = ');
disp('Glcz = ');
Gz = c2d(Gp,T,'zoh');
Glaz = K*Gz;
Glcz = feedback(Glaz,1)
P = pole(Glcz);
Mag = abs(P);
disp('*************************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
if Mag<=1
disp('*************************************');
disp('RESPUESTA : EL SISTEMA ES ESTABLE');
disp('POLOS de Glcz (Lazo cerrado): ');
display(Mag)
```

```
disp('***********************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
disp('***********************************');
disp('LOS VALORES DE zita y Wn SON: ');
teta = angle(log(P(1)));
zita = -cos(teta)
Wn = abs(log(P(1)))/T
disp('***********************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
% Tambien se puede encontrar [Wn, zita]= damp(Glaz)
% Para entrada con escalon unitario
% Uz = z/(z-1)
Uz = tf([1 0],[1 -1],T);
disp('*******************************');
disp('Cz = ');
Cz = Glcz*Uz;
Cz1= zpk(Cz)
p = pole(Cz);
[numc,denc]= tfdata(Cz,'v');
syms k z
% Se halla c(kT) por fracciones parciales
[r,p,k]=residue(numc,denc);
N = input('ENTRE NUMERO DE MUESTRAS: N = ');
for k=0:N
   ck1=r(1)*p(1)^(k-1);
   ck2=r(2)*p(2)^(k-1);
   ck3=r(3)*p(3)^(k-1);
   ck=ck1+ck2+ck3;
   c(k+1)=ck;
end
plot(c,'r-')
legend('GRAFICA DE SALIDA c(kT)')
grid
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
disp('********************************');
disp('TIEMPO DE PICO : ');
i=find(c==max(c));
kp = i-1;
```

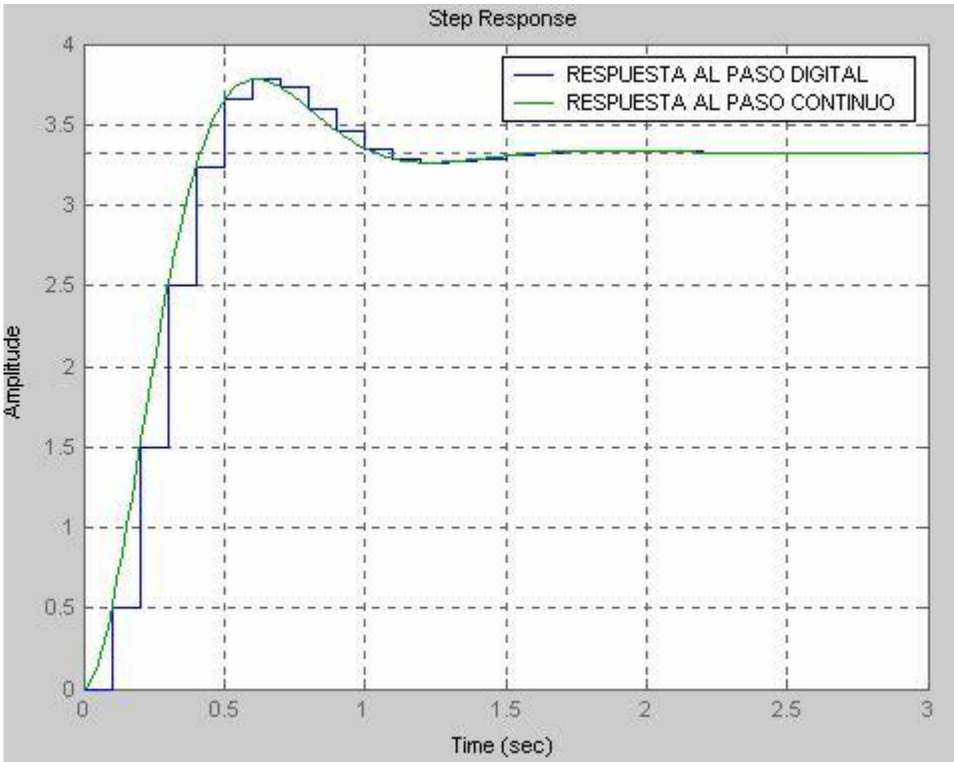
```
tp=kp*T
disp('SOBREIMPULSO:');
Czsym = poly2sym(numc,'z')/poly2sym(denc,'z');
epsilon =1.0e-10;
uno=1+epsilon;
css = limit((1-z^-1)*Czsym,z,uno);
css = numeric(css);
Cpico = c(kp+1)
Mp = Cpico-css;
PorcientoMp = 100*Mp/css
disp('*******************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp('*******************************');
disp(' ');
disp('ERROR ESTADO ESTACIONARIO: ');

% Kp es la consatante de error estacionario
[numz,denz] = tfdata(Glaz,'v');
Glazsym = poly2sym(numz,'z')/poly2sym(denz,'z');
Kp = limit(Glazsym,z,uno);
Kp = numeric(Kp);
if Kp>1.0e5
  Kp=inf
else
  Kp =Kp
end
ess = 1/(1+Kp);
ess=numeric(ess)
disp('*******************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
disp('RESPUESTA AL PASO DIGITAL Y CONTINUO:')

% Respuesta al escalon unitario digital y continuo
Glcs = d2c(Glcz,'zoh');
step(Glcz,Glcs)
legend('RESPUESTA AL PASO DIGITAL','RESPUESTA AL PASO
CONTINUO')
grid
else
 disp('******************************************************');
 disp(' ');
 disp('OJO: EL SISTEMA ES INESTABLE, PRUEBE OTRO T u OTRO K');
```

```
  disp('POLOS de Glcz (Lazo cerrado): ');
  display(Mag)
  disp('*****************************************************');
end                    % fin de if
disp(' ');
SIGA = input('PRESIONE TECLA 1 PARA SEGUIR: ');
end                      % fin de while
```

Respuesta:

(a) ***ζ = 0.5348,   Wn = 6.0510***
(c) ***tp = 0.6 sg, ess = 3.3277, Mp = 13.65%***
(b) ***Gráfica de la secuencia de salida c(kT)***

(d) Respuesta al paso digital y continuo



6.2 ADICIÓN DE POLOS Y CEROS

El diseño de sistemas de sistemas de control se basa principalmente en la adición de polos y ceros a la función de transferencia de lazo abierto en locaciones deseables y la supresión de polos y ceros en locaciones no deseables en el plano z, que es lo que en la práctica realiza la adición del controlador a la planta.

6.2.1 ADICIÓN DE UN CERO A Glaz

EJEMPLO 6-2:

Para el problema del ejemplo anterior del vehículo espacial probar que pasa con el sobreimpulso y con el tiempo de pico al adicionar ceros a la derecha y a la izquierda en la función de transferencia del lazo abierto. Considere T = 0.1 sg, K = 1, y ceros en: z1 = 0, 0.2, 0.5, 0.8, 1.0, - 0.2, - 0.5, - 0.8, - 1.0.

Simulación:

PROGRAMA EN MATLAB :

```
 % EJEMPLO 6-2: ADICION DE UN CERO A Glaz
clear
nump = input('ENTRE NUMERADOR : nump = ');
denp = input('ENTRE DENOMINADOR : denp = ');
Gp = tf(nump,denp)
SIGA = 1;
while SIGA ==1
T = 0.1;
K = input('VALOR DE GANANCIA: K = ');
disp('Glaz = ');
Gz = c2d(Gp,T,'zoh');
Glaz = K*Gz
z1 = input('ENTRE CERO ADICIONAL A Glaz : z1 = ');
Gcero = tf([1 z1],1,T);
Glaz = Glaz*Gcero;
Glcz = feedback(Glaz,1);
P = pole(Glcz);
Mag = abs(P);
disp('***********************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
if Mag<=1
disp('LOS VALORES DE Wn y zita SON : ');
[Wn,zita]= damp(Glaz)

% Para entrada con escalon unitario
% Uz = z/(z-1)
Uz = tf([1 0],[1 -1],T);
Cz = Glcz*Uz;
p = pole(Cz);
[numc,denc]= tfdata(Cz,'v');
syms k z
% Se halla c(kT) por fracciones parciales
[r,p,k]=residue(numc,denc);
for k=0:100
  ck1=r(1)*p(1)^(k-1);
  ck2=r(2)*p(2)^(k-1);
  ck3=r(3)*p(3)^(k-1);
  ck=ck1+ck2+ck3;
  c(k+1)=ck;
end
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
```

```
disp('********************************');
disp('TIEMPO DE PICO : ');
i=find(c==max(c));
kp = i -1;
tp=kp*T
disp('SOBREIMPULSO:');
Czsym = poly2sym(numc,'z')/poly2sym(denc,'z');
epsilon =1.0e-10;
uno=1+epsilon;
css = limit((1-z^-1)*Czsym,z,uno);
css = numeric(css);
Cpico = c(kp+1)
disp('********************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp('********************************');
disp(' ');
disp('ERROR ESTADO ESTACIONARIO: ');

% Kp es la consatante de error estacionario
[numz,denz] = tfdata(Glaz,'v');
Glazsym = poly2sym(numz,'z')/poly2sym(denz,'z');
Kp = limit(Glazsym,z,uno);
Kp = numeric(Kp);
if Kp>1.0e5
  Kp=inf
else
  Kp =Kp
end
ess = 1/(1+Kp);
ess=numeric(ess)
disp('********************************');
disp('PRESIONE UNA TECLA');
pause
step(Glcz)
else
   disp('******************************************************');
   disp(' ');
   disp('OJO: EL SISTEMA ES INESTABLE, PRUEBE OTRO T u OTRO K');
   disp('POLOS de Glcz (Lazo cerrado): ');
   display(Mag)
   disp('******************************************************');
end                % fin de if
disp(' ');
SIGA = input('PRESIONE TECLA 1 PARA SEGUIR: ');
end                % fin de while
```

Conclusión: ***Al adicionar un cero a la izquierda disminuye el sobreimpulso y el tiempo de pico aumenta pero entre más alejado esté estos valores aumentan. Al adicionar un cero a la derecha disminuye el sobreimpulso y el tiempo de pico aumenta y entre más alejado esté el sobreimpulso disminuye y el sobrepico aumenta.***

6.2.2 ADICIÓN DE UN POLO A Glaz

EJEMPLO 6-3:

Repita el ejemplo anterior adicionando ahora polos en el semiplano derecho y en el izquierdo. T = 0.1 sg, K = 1. Adicione polos en p1 = 0, 0.2, 0.5, 0.8, 1.0, - 0.1, - 0.2, - 0.5, - 0.8, 1.0.

Simulación:

```
% EJEMPLO 6-3: ADICION DE UN CERO A Glaz
clear
nump = input('ENTRE NUMERADOR : nump = ');
denp = input('ENTRE DENOMINADOR : denp = ');
Gp = tf(nump,denp)
SIGA = 1;
while SIGA ==1
T = 0.1;
K = input('VALOR DE GANANCIA: K = ');
disp('Glaz = ');
Gz = c2d(Gp,T,'zoh');
Glaz = K*Gz
p1 = input('ENTRE POLO ADICIONAL A Glaz : p1 = ');
Gpolo = tf(1,[1 p1],T);
Glaz = Glaz*Gpolo;
Glcz = feedback(Glaz,1);
P = pole(Glcz);
Mag = abs(P);
disp('**********************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
if Mag<=1
```

```
disp('LOS VALORES DE Wn y zita SON : ');
[Wn,zita]= damp(Glaz)

% Para entrada con escalon unitario
% Uz = z/(z-1)
Uz = tf([1 0],[1 -1],T);
Cz = Glcz*Uz;
p = pole(Cz);
[numc,denc]= tfdata(Cz,'v');
syms k z
% Se halla c(kT) por fracciones parciales
[r,p,k]=residue(numc,denc);
for k=0:100
   ck1=r(1)*p(1)^(k-1);
   ck2=r(2)*p(2)^(k-1);
   ck3=r(3)*p(3)^(k-1);
   ck=ck1+ck2+ck3;
   c(k+1)=ck;
end
disp('PRESIONE UNA TECLA');
pause
clc
disp(' ');
disp('********************************');
disp('TIEMPO DE PICO : ');
i=find(c==max(c));
kp = i -1;
tp=kp*T
disp('SOBREIMPULSO:');
Czsym = poly2sym(numc,'z')/poly2sym(denc,'z');
epsilon =1.0e-10;
uno=1+epsilon;
css = limit((1-z^-1)*Czsym,z,uno);
css = numeric(css);
Cpico = c(kp+1)
disp('********************************');
disp('PRESIONE UNA TECLA');
pause
clc
disp('********************************');
disp(' ');
disp('ERROR ESTADO ESTACIONARIO: ');

% Kp es la consatante de error estacionario
[numz,denz] = tfdata(Glaz,'v');
Glazsym = poly2sym(numz,'z')/poly2sym(denz,'z');
Kp = limit(Glazsym,z,uno);
```

```
Kp = numeric(Kp);
if Kp>1.0e5
   Kp=inf
else
   Kp =Kp
end
ess = 1/(1+Kp);
ess=numeric(ess)
disp('*******************************');
disp('PRESIONE UNA TECLA');
pause
step(Glcz)
else
disp('*****************************************************');
disp(' ');
disp('OJO: EL SISTEMA ES INESTABLE, PRUEBE OTRO T u OTRO K');
disp('POLOS de Glcz (Lazo cerrado): ');
display(Mag)
disp('*****************************************************');
end
disp(' ');
SIGA = input('PRESIONE TECLA 1 PARA SEGUIR: ');
end
```

Conclusión: ***Al adicionar un polo a la izquierda aumenta el sobrepico y el tiempo de pico pero entre más alejado esté estos valores disminuyen. Al adicionar un polo a la derecha aumenta el sobreimpulso y el sobrepico y entre más alejado esté estos valores aumentan.***

7. ANÁLISIS : DOMINIO EN FRECUENCIA

7.1 DIAGRAMA DE NYQUIST

El análisis en el dominio de la frecuencia de un sistema de control digital es muy antiguo y completo para sistemas continuos, pero sus métodos de análisis son aplicables a los sistemas digitales o discretos. El estudio en el dominio de la frecuencia se basa en el diagrama de Nyquist (gráfica polar), en los diagramas de Bode (amplitud-fase) y los diagramas de Nichols.

El análisis se hace con base en entradas de tipo senoidal en estado estacionario, por eso el análisis y diseño en el dominio de la frecuencia un controlador digital se comporta como un filtro digital.

El diagrama de Nyquist de una función de transferencia en lazo abierto es una gráfica polar de los puntos del círculo unitario del plano z sobre el plano Glaz. La estabilidad absoluta y relativa del sistema en lazo cerrado se obtiene con este diagrama.

La gráfica polar se obtiene reemplazando,

$$z = e^{jwT}$$

en la función de transferencia Glaz y variando w desde 0 hasta ∞ se mapean los puntos de Glaz. Como el círculo unitario se recorre una vez para toda w = n*ws, donde n = 1, 2, 3, . . . se requiere solamente graficar desde w= 0, hasta w = ws.

EJEMPLO 7-1:

Obtener el diagrama de Nyquist para una función de transferencia en lazo abierto de un sistema digital cuya frecuencia de muestreo Ws = 4 rad/sg (T = pi / 2) cuya planta es,

$$Gp(s) = \frac{1.57}{s(s+1)}$$

Solución:

$$Gla(z) = \frac{1.2225(z+0.5979)}{(z-1)(z-0.2979)}, \quad reemplazando \quad z = e^{jwT}$$

$$Gla(e^{jwT}) = \frac{1.2225(e^{jwT}+0.5979)}{(e^{jwT}-1)(e^{jwT}-0.2979)}$$

$$Gla(e^{jwT}) = Re\left[Gla(e^{jwT})\right] + jIm\left[Gla(e^{jwT})\right]$$

Para w = 0 :

$$z = e^{jwT} = 1, \Rightarrow Gla(e^{jwT}) = \infty$$

Para w = ws / 2 :

wT = (ws/2)(2π/ws) = π

$$e^{jwT} = cos(wT) + jsen(wT) = cos\pi + jsen\pi = -1$$

$$Gla(-1) = \frac{1.2225(-1 + 0.5979)}{(-1-1)(-1-0.2979)} = -0.2$$

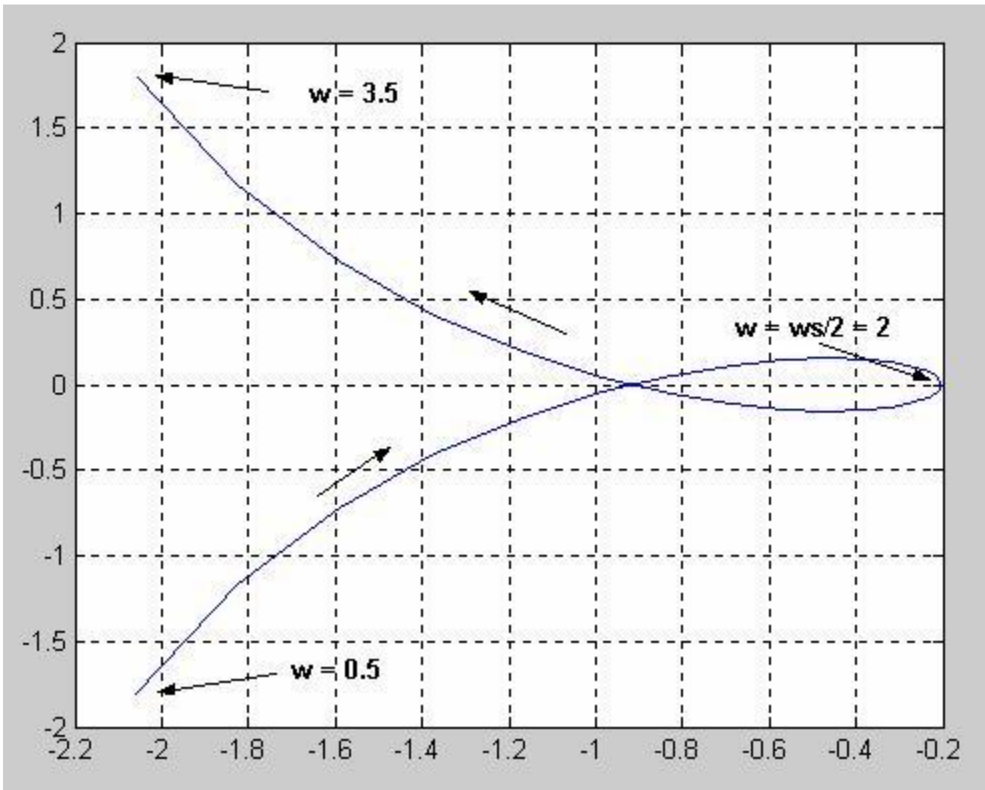
Para w = ∞ = (n*ws) :

$$z = e^{jwT} = 1, \Rightarrow Gla(e^{jwT}) = \infty$$

Simulación:

```
%EJEMPLO 7-1 : GRAFICA DE NYQUIST
disp('GRAFICA DE NYQUIST');
clear
num = input('NUMERADOR DE Glas : num = ');
den = input('DENOMINADOR DE Glas : den = ');
Glas = tf(num,den);
T = input('PERIODO DE MUESTREO: T = ');
disp('FRECUENCIA DE MUESTREO: Ws = ');
Ws = 2*pi/T
Glaz = c2d(Glas,T,'zoh')

% METODO ANALITICO
[numz,denz] = tfdata(Glaz,'v');
Glazsym = poly2sym(numz,'z')/poly2sym(denz,'z');
i = 1;
for w = 0.5:0.1:3.5;
   z = exp(j*w*T);
   G = eval(Glazsym);
   Gx = real(G);
   Gy = imag(G);
   Gr(i) = Gx
   Gi(i) = Gy
   i = i+1;
end
plot(Gr,Gi)
grid

% METODO MATLAB INDIRECTO
w = 0.5:0.1:3.5;
[Real,Imag] = nyquist(Glaz,w)
```

```
Plot(Real,Imag)
grid

% METODO MATLAB DIRECTO
nyquist(Glaz,{0.5,3.5})
grid
```



7.2 CRITERIO DE NYQUIST

El criterio de Nyquist determina la estabilidad de un sistema en lazo cerrado mediante la gráfica polar de la función de transferencia del sistema en lazo abierto.

Según el criterio de Nyquist el sistema de lazo cerrado es estable si para Imag(Glaz) = 0, la parte Real(Glaz) se encuentra dentro del intervalo :

$$-1 < Real(Glaz) < 0$$

En este caso, la ganancia máxima permitida para que el sistema siga siendo estable es :

Kmax = -1 / Real(Glaz)

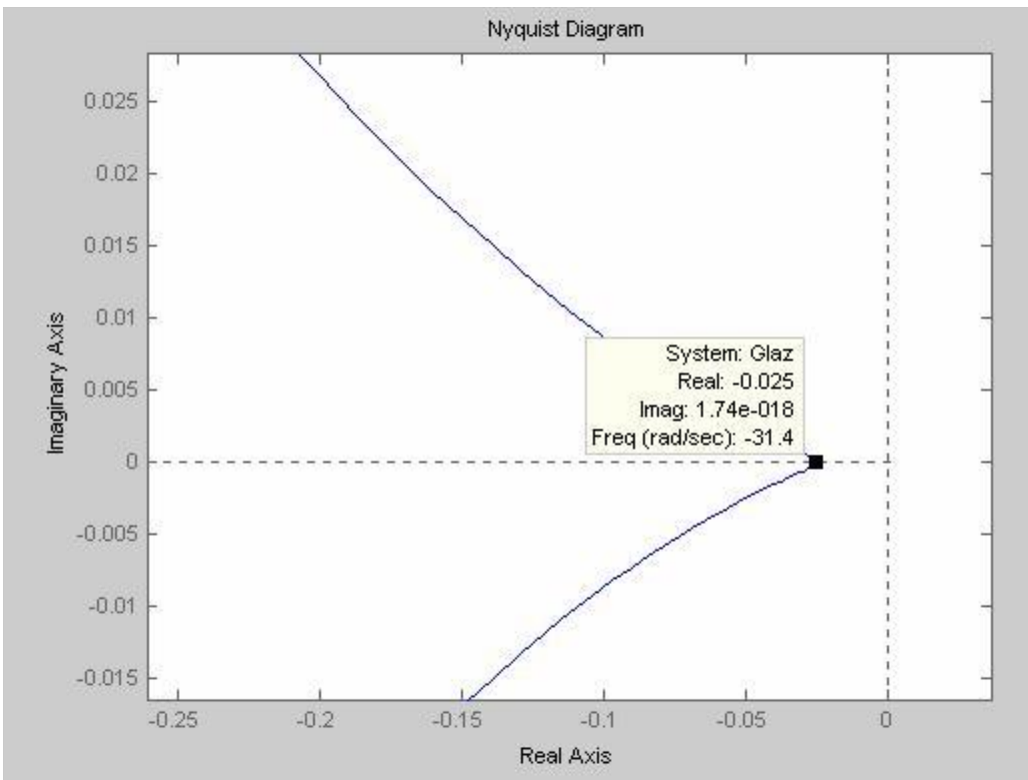
EJEMPLO 7-2:

Para la función de transferencia de lazo abierto, T = 0.1 sg;

$$Gla(z) = \frac{0.0952z}{(z-1)(z-0.905)}$$

ws = 2π / T = 62.8 rad/sg

El diagrama de Nyquist para el intervalo 5 < w < ws/2 es el siguiente:



De la gráfica se observa que para Imag(Glaz) ≈ 0, Real(Glaz) = - 0.025. Como este valor se encuentra en el intervalo de estabilidad, entonces el sistema de lazo cerrado es estable y la ganancia máxima permisible sería igual a :

Kmax = - 1/Real(Glaz) = - 1 / (- 0.025) = 40, entonces el valor de K debe estar :

0 < K < 40

Simulación:

PROGRAMA EN MATLAB :

```
% EJEMPLO 7-2 : CRITERIO DE ESTABILIDAD DE NYQUIST
disp('****************************************');
disp('CRITERIO DE ESTABILIDAD DE NYQUIST ');
clear
Z = input('ENTRE VECTOR DE CEROS: Z = ');
P = input('ENTRE VECTOR DE POLOS: P = ');
K = input('ENTRE VALOR DE GANANCIA: K = ');
T = input('ENTRE TIEMPO DE MUESTREO: T = ');
Glaz = zpk(Z,P,K,T)
% Frecuencia de muestreo
Ws = 2*pi/T
disp('****************************************');
disp('OPRIMA CUALQUIER TECLA ');
pause
clc
disp(' ');
% nyquist opera para 0 > w > wmax
nyquist(Glaz,{5,Ws/2})
disp(' Real DEBE ESTAR A LA DERECHA DE (-1,j0) ');
w = 0.1:0.001:Ws/2;
[Real,Imag] = nyquist(Glaz,w);
delta =1.0e-6;
i=[ ];
while i==[ ]
   i = find(Imag<(delta)&Imag>(-delta));
   delta = delta*1.0e+1;
end
m = length(i);
disp('VALORES REALES CUANDO IMAG APROX = 0 ');
Reales = Real(i)

% Kmax = 1/Real cuando (Imag = 0)
KM=[ ];
for j=1:m
   l=1;
   if Real(i(j))<0 & Real(i(j))>-1
      Km = - 1/Real(i(j));
      KM(l) = Km;
      l=l+1;
   end
```

```
end        % end de for
Kmax = min(KM);
if Kmax>0
   disp(' EL SISTEMA ES ESTABLE PORQUE Kmax > 0 ');
   display(Kmax)
else
   disp(' EL SISTEMA ES INESTABLE PORQUE TODOS LOS VALORES');
   disp(' DE LOS REALES PARA IMG=0 NO ESTA ENTRE -1:0 ');
end
```

7.3 DIAGRAMAS DE BODE

El diagrama de Bode de una función de transferencia de lazo cerrado se realiza graficando en función de la frecuencia (0< w< ws) la magnitud en decibelios y la fase en grados en dos gráficas independientes de la función de transferencia de lazo abierto. Para fines prácticos basta dibujar el diagrama de Bode entre 0 y ws/2.

$$Mag(Glaz(z)) = 20log10 * \left|Glaz(e^{jwT})\right|$$

$$Ang(Glaz(z)) = angle(Glaz(e^{jwT}))$$

EJEMPLO 7-3:

Para el ejemplo anterior realizar el diagrama de Bode.

$$Gla(z) = \frac{0.0952z}{(z-1)(z-0.905)}$$

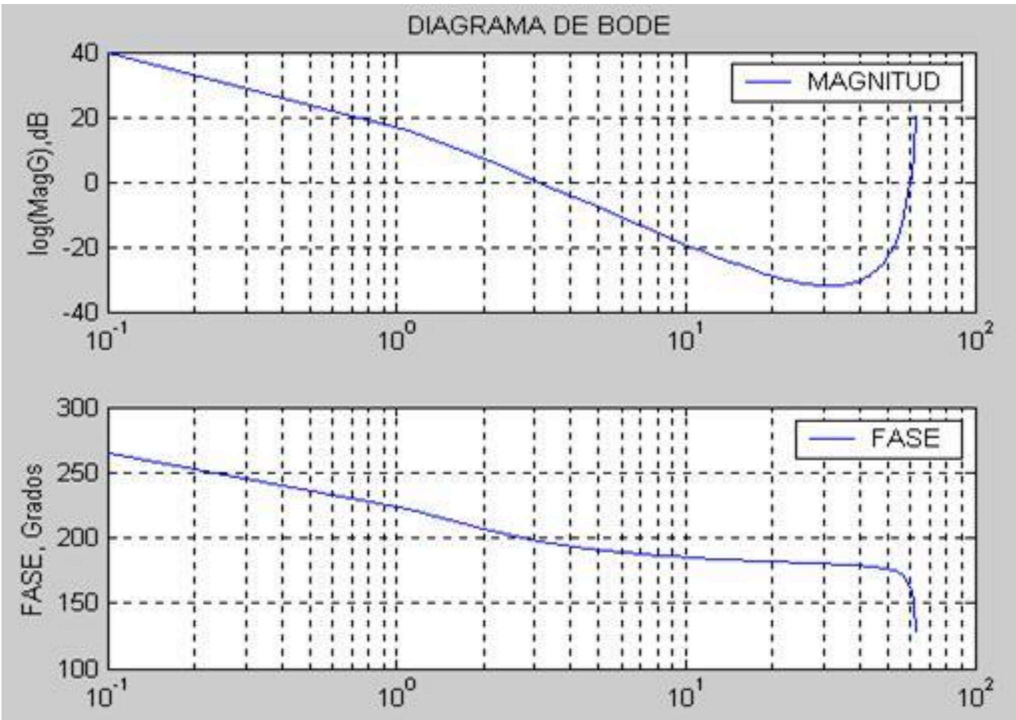
T = 0.1 sg, entonces, Ws = 2□ / T = 62.8 rad/sg

PROGRAMA EN MATLAB :

```
% EJEMPLO 7-3 : DIAGRAMA DE BODE
disp('****************************************');
disp('DIAGRAMA DE BODE ');
clear
Z = input('ENTRE VECTOR DE CEROS: Z = ');
P = input('ENTRE VECTOR DE POLOS: P = ');
K = input('ENTRE VALOR DE GANANCIA: K = ');
T = input('ENTRE TIEMPO DE MUESTREO: T = ');
Glaz = zpk(Z,P,K,T)
% Frecuencia de muestreo
```

```
Ws = 2*pi/T
disp('***************************************');
disp('OPRIMA CUALQUIER TECLA ');
pause
clc
disp(' ');
disp('TECLEE 1: METODO ANALITICO');
disp('TECLEE 2: METODO MATLAB INDIRECTO');
disp('TECLEE 3: METODO MATLAB DIRECTO');
m = input('SELECCIONE LA OPCION: ');
switch m

case 1
% METODO ANALITICO
[numz,denz] = tfdata(Glaz,'v');
Glazsym = poly2sym(numz,'z')/poly2sym(denz,'z');
i = 1;
for w = 0.1:1:Ws;
z = exp(j*w*T);
G = eval(Glazsym);
Mag = 20*log10(abs(G));
Ang = atan(imag(G)/real(G))*180/pi;
if real(G)>0&imag(G)>0
   An(i)=Ang;
elseif
   real(G)<0&imag(G)>0
   An(i)=180+Ang;
elseif
    real(G)<0&imag(G)<0
   An(i)=180+Ang
elseif
    real(i)>0&imag(G)<0
   An(i)=360+Ang
end          % end de if
W(i) = w
Mg(i) = Mag
i = i+1;
end        % end de for

subplot(211)
semilogx(W,Mg)
grid
ylabel('log(MagG),dB')
title('DIAGRAMA DE BODE')
legend('MAGNITUD')
subplot(212)
semilogx(W,An)
```

```
grid
ylabel('FASE, Grados')
legend('FASE');

case 2
% METODO INDIRECTO
w = [0.1:1:Ws];
[Mag,Fase] = Bode(Glaz,w);
for i=1:Ws
   Mg(i)=20*log10(Mag(:,:,i))
   Fs(i)=Fase(:,:,i);
end
subplot(211)
semilogx(w,Mg)
grid
subplot(212)
semilogx(w,Fs)
grid

case 3
% METODO DIRECTO
close figure No. 1
w = [0.1:1:Ws]
bode(Glaz,w)
grid
end             % fin de while
```



Como la curva de magnitud se da en decibelios un cambio en el factor de ganancia se traduce en un movimiento hacia arriba o hacia debajo de la curva de magnitud sin distorsionarla y un cambio en la fase en un factor de retraso de nT se traduce en una disminución de la fase en un factor n wT.

$$20log10\left|G(e^{j\omega T})Gc(e^{j\omega T})\right| = 20log10\left|G(e^{j\omega T})\right| + 20log10\left|Gc(e^{j\omega T})\right|$$

$$\angle\left|G(e^{j\omega T})Gc(e^{j\omega T})\right| = \angle\left|G(e^{j\omega T})\right| + \angle\left|Gc(e^{j\omega T})\right|$$

7.4 MÁRGENES DE GANANCIA Y FASE

El margen de ganancia y el margen de fase son especificaciones de desempeño en el dominio de la frecuencia utilizados en el diseño de sistemas de control digital.

7.4.1 MARGEN DE GANANCIA

Si un sistema de lazo cerrado es estable para un Kmax = 40 se dice que cuando K = 1, el margen de ganancia del sistema es de 40, esto es, en decibelios (diagrama de Bode) = 20 log10(40) = 32 dB. O sea, en K = 1, la magnitud en w = ws/2=31.4 rad/sg cuando la fase = 180º es igual a - 32 dB.

Por definición, margen de magnitud es igual a:

$$MG = 20log10\frac{1}{\left|Glaz(e^{j\omega_{cp}T})\right|} = 0\,dB - dB\left|Glaz\right|_{w=w_{cp}}$$

Donde w_{cp} es la frecuencia de cruce de fase que corresponde a la fecuencia donde la fase es igual a 180º.

7.4.2 MARGEN DE FASE

Se debe tener en cuenta el margen de fase para asegurar la estabilidad relativa de un sistema de control. Un sistema con margen de ganancia grande puede tener una estabilidad relativa pequeña debido a cambios en la fase del sistema.

Por definición, margen de fase es igual a :

$$MP = \angle Glaz(e^{j\omega_{cg} T}) - 180^{o}$$

Donde wcg es la frecuencia de cruce de ganancia que corresponde a la frecuencia donde la magnitud es igual a 0 dB.

En resumen el margen de ganancia se mide en la frecuencia de cruce de fase y el margen de fase en la frecuencia de cruce de ganancia.

EJEMPLO 7-4:

Para el ejemplo anterior determinar el margen de ganancia y el margen de fase.

Simulación:

PROGRAMA EN MATLAB :

```
% EJEMPLO 7-4 : MARGEN DE GANANCIA Y DE FASE
clear
clc
disp('*******************************************');
disp('MARGEN DE GANANCIA Y DE FASE ');
Z = input('ENTRE VECTOR DE CEROS: Z = ');
P = input('ENTRE VECTOR DE POLOS: P = ');
K = input('ENTRE VALOR DE GANANCIA: K = ');
T = input('ENTRE TIEMPO DE MUESTREO: T = ');
Glaz = zpk(Z,P,K,T)
% Frecuencia de muestreo
Ws = 2*pi/T
disp('*****************************************');
disp('OPRIMA CUALQUIER TECLA ');
pause
SIGA = 1;
while SIGA==1
clc
disp(' ');
disp('TECLEE 1: ESTABILIDAD POR LGR');
disp('TECLEE 2: MARGEN DE GANANCIA Y FASE: ANALITICO');
disp('TECLEE 3: MARGEN DE GANANCIA Y FASE:MATLAB');
m = input('SELECCIONE LA OPCION: ');
switch m

case 1
% ESTABILIDAD POR LGR
rlocus(Glaz)
```

```
hold on
r = 0:0.01:2*pi;
x = sin(r);
y = cos(r);
plot(x,y)
[K,P] = rlocfind(Glaz);
Mag = abs(P);
disp('LA GANANCIA CRITICA ES IGUAL A : ');
display(K);

case 2
% MARGEN DE GANANCIA Y FASE ANALITICAMENTE
% (a) Calculo de fase y ganancia en funcion de w
[numz,denz] = tfdata(Glaz,'v');
Glazsym = poly2sym(numz,'z')/poly2sym(denz,'z');
i = 1;
for w = 0.1:0.1:Ws/2;
z = exp(j*w*T);
G = eval(Glazsym);
Mag = 20*log10(abs(G));
Ang = atan(imag(G)/real(G))*180/pi;
if real(G)>0&imag(G)>0
   An(i)=Ang;
elseif real(G)<0&imag(G)>0
   An(i)=180+Ang;
elseif real(G)<0&imag(G)<0
   An(i)=180+Ang
elseif real(i)>0&imag(G)<0
   An(i)=360+Ang
end          % end de if
W(i) = w
Mg(i) = Mag
i = i+1;
end          % end de for

% (b) Margen de ganancia donde fase = 180
delta = 0.1;
Wcp = [ ];
while isempty(Wcp)
j=find(An<(180+delta)&An>(180-delta));
A = An(j);
i=find(An==min(A));
Wcp = W(i);
MGx = Mg(i);
MG = 0 - MGx
delta = delta +0.1;
end          % find de while
```

```
disp('FRECUENCIA DE CRUCE DE FASE: ');
display(Wcp)
disp('MARGEN DE GANANCIA. ');
display(MG)

% (c) Margen de fase donde ganancia = 0 dB
delta = 0.1;
Wcg = [ ];
while isempty(Wcg)
j=find(Mg<(0+delta)&Mg>(0-delta));
M = Mg(j);
i = find(Mg==min(M));
Wcg =W(i);
MPx = An(i);
MP = MPx-180;
delta = delta +0.1;
end            % fin de while
disp('FRECUENCIA DE CRUCE DE GANANCIA: ');
display(Wcg)
disp('MARGEN DE FASE: ');
display(MP)

case 3
% MARGEN DE GANANCIA Y FASE POR MATLAB
clc
[MG,MP,Wcp,Wcg] = margin(Glaz);
disp('MARGEN DE GANANCIA ');
MGdB = 20*log10(MG);
display(MGdB)
disp('MARGEN DE FASE ');
display(MP)
disp('FRECUENCIA DE CRUCE DE FASE ');
display(Wcp)
disp('FRECUENCIA DE CRUCE DE GANANCIA ');
display(Wcg)
end             % end de switch

SIGA = input('TECLE 1 PARA SEGUIR : ');
end             % end de while
```

Respuesta:

Wcp = 31.4 rad/sg, ***MG = 32.04 dB***

Wcg = 3.1 rad/sg, ***MP = 17.7º***

7.5 CARTA DE NICHOLS

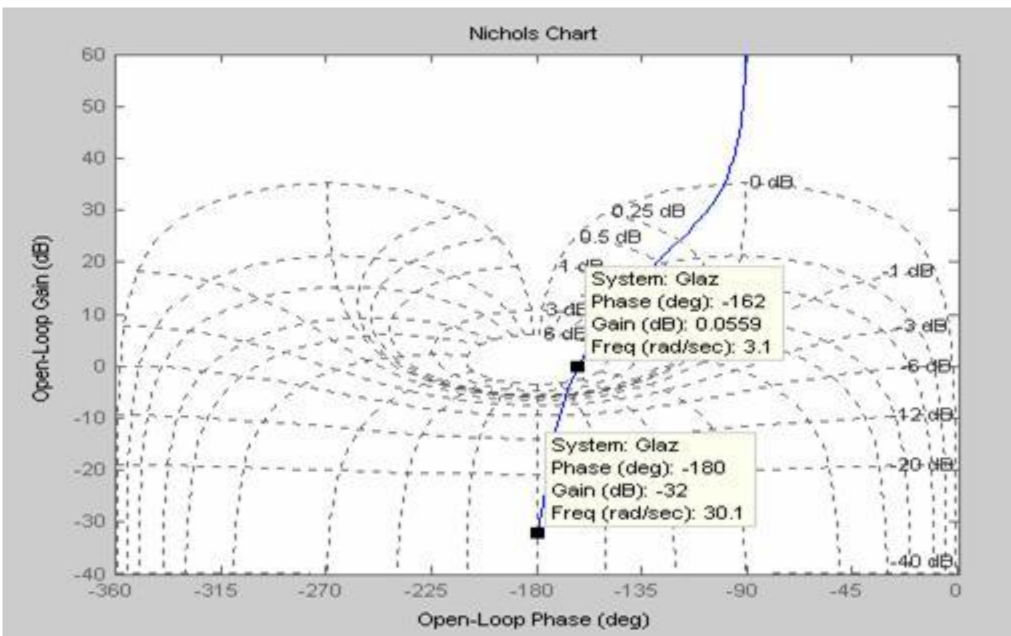
La carta de Nichols proporciona información de la función de transferencia en lazo cerrado para un sistema con retroalimentación unitaria a partir de la gráfica de la ganancia en función de la fase utilizando como parámetro la frecuencia de una función de transferencia en lazo abierto.

Continuando con el mismo ejemplo:

w = 0.1:1:100;
nichols(Glaz)

se obtiene la siguiente gráfica activando posteriormente del submenú la opción grid para dibujar la malla del sistema en lazo cerrado. De la gráfica se observa que se tiene una frecuencia de cruce de ganancia igual a 3.1 rad/sg con un margen de fase aproximado de 180-160 = 18º (ganancia = 0 dB), de forma similar la frecuencia de cruce de fase es de 30.1 rad/sg con un margen de ganancia igual a 0 – (- 32) = 32 dB (fase = 180º).

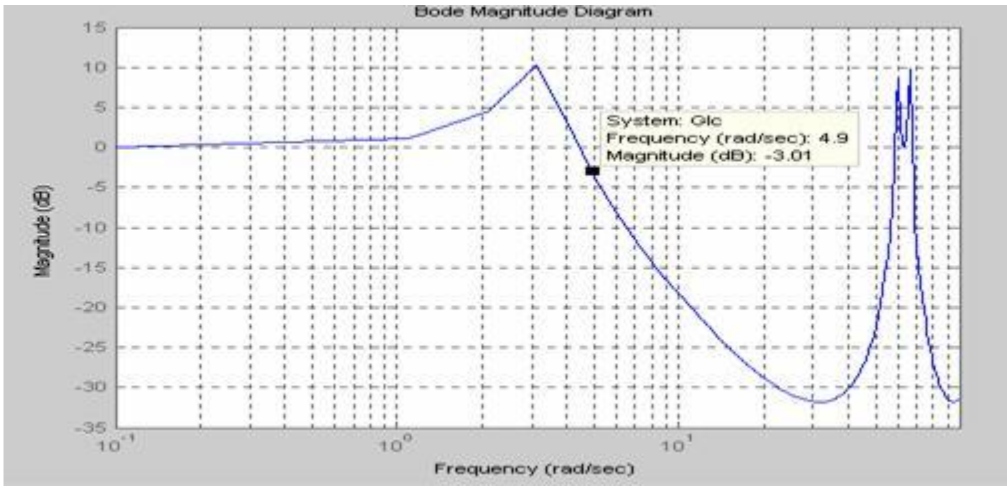
Para encontrar el ancho de banda del sistema en lazo cerrado, se busca la intersección de la carta de Nichols con la ganancia del lazo cerrado en -3.0 dB. Dela curva se obtiene un valor aproximado de 4.9 rad/sg.



Otra manera de encontrar el ancho de banda es graficar la magnitud de la función del lazo cerrado en función de la frecuencia (Bode) y obtener la frecuencia cuando la magnitud es igual a -3.0 dB

w = 0.1:1:100;
Glc = feedback(Glaz,1);
bodemag(Glc,w)

Se obtiene la siguiente gráfica en donde se observa que BW = 4.9 rad/seg



7.6 SENSIBILIDAD

Un sistema de control además de cumplir con los requisitos de estabilidad debe tener un desempeño robusto ante perturbaciones externas y variaciones de los parámetros. Se debe limitar la sensibilidad a cierto valor en un intervalo de frecuencias. La sensibilidad de Glc con respecto a Gla se define como:

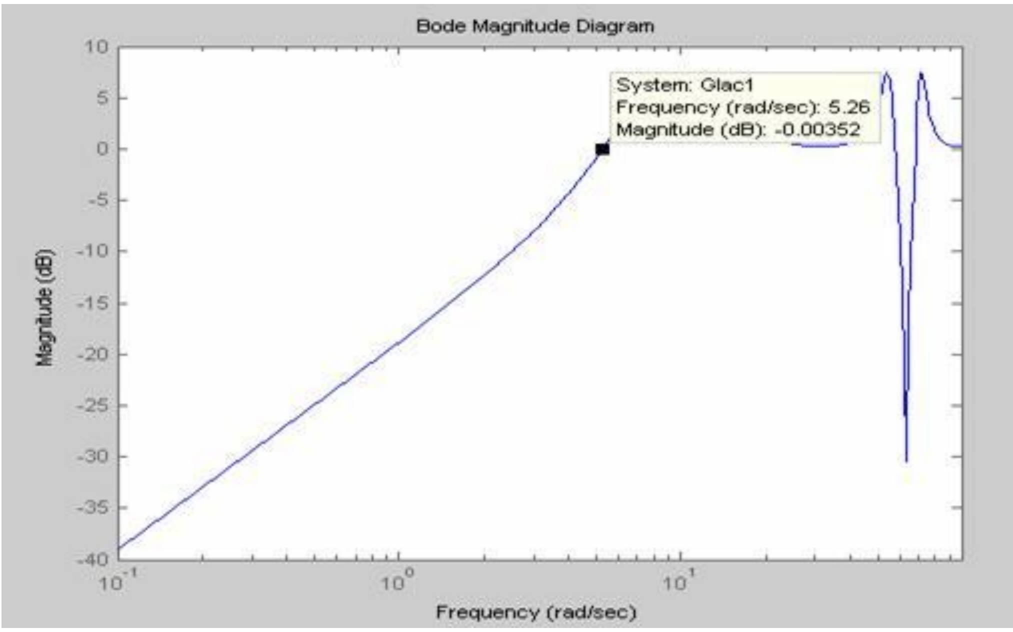
$$S(z) = \frac{dGlc(z)/Glc(z)}{dGla(z)/Gla(z)} = \frac{dGlc(z)}{dGla(z)} \cdot \frac{Gla(z)}{Glc(z)} = \frac{1}{1+Gla(z)} = \frac{1/Gla(z)}{1+\left[1/Gla(z)\right]}$$

La ecuación nos indica que la sensibilidad es análoga a una función de transferencia de lazo cerrado si la función de transferencia de lazo abierto es 1 / Gla(z).

Nótese que la sensibilidad es pequeña si la ganancia de Gla(z) es grande, pero puede traer inestabilidad.

Gla(z) = 0.3(z+0.7433) / (z-1)(z-0.4119)

Glaz = zpk(-0.7433,[1 0.4119], 0.3,0.1);
Glaz1 = 1/ Glaz;
w = 0.1:1:100;
Bodemag(Glaz1,w)



La sensibilidad es menor a 1.0 (0 dB) para w < 5.26 rad/sg

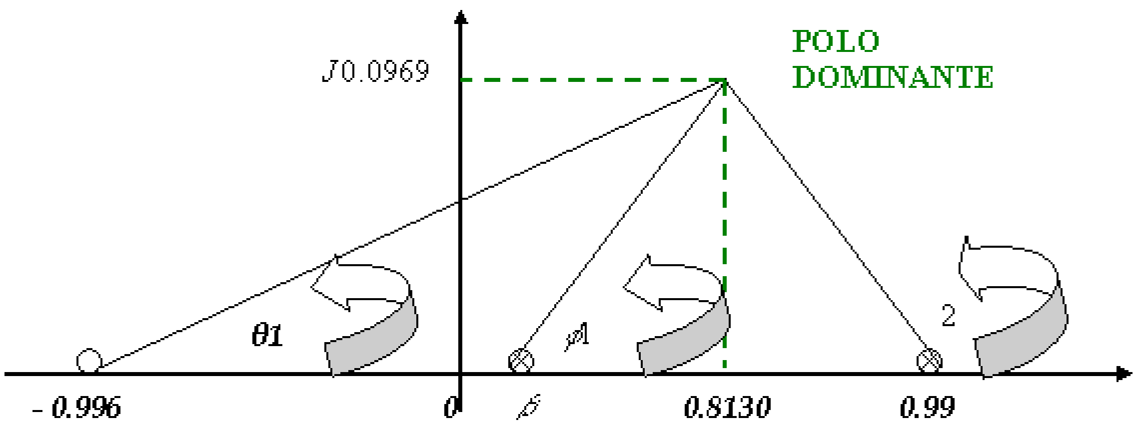
8. DISEÑO: DOMINIO - TIEMPO

8.1 CANCELACIÓN DE POLOS

***%**
% MOTOR DE CORRIENTE CONTINUA CONTROLADO EN VELOCIDAD
% SEGUN MODELO REALIZADO EN EL CAP3_5
% Gp(s) = 2/(s+10)(s+2)
% CONDICIONES DE DISEÑO(CAP6_1) : ts = 0.02 seg, T = 0.001 seg, Mp = 0.5%
%***
clear all
home
disp(' ');***

```
disp('***************************************************************************')
disp('DISEÑO DE SISTEMAS DE CONTROL DISCRETO');
disp('DISEÑO EN EL DOMINIO DEL TIEMPO');
disp('DISEÑO MEDIANTE CANCELACION DE POLOS');
disp('***************************************************************************')
% FUNCION DE TRANSFERENCIA DE LA PLANTA Y DEL CONTROLADOR :
%                  2                                  z - alfa
% Gp(s)= -----------------            Gc(z)= k * -----------
%           (s+10)s(s+2)                              z - beta

disp('  ');
ts=input('Digite el tiempo de establecimiento : ts(seg) = ');
%  ts = 0.02;
disp(' ');
T=input('Digite el tiempo de muestreo "T" (seg): ');
%  T=0.001;
disp(' ');
Mp=input('Digite maximo sobreimpulso : Mp = ');
% Mp = 0.5%;
cita= solve('Mp = 100*exp(-cita*3.1416/sqrt(1-cita^2))');
cita = eval(cita)
% cita = 0.86
disp('  ');
disp('  ');

% (a) calculo del polo dominante en todo el sistema lazo cerrado
wn=4/(cita*ts);              % frecuencia natural para 2% de tolerancia
wd=wn*sqrt(1-(cita^2)); % Frecuencia amortiguada
ws=(2*pi)/T;                 % Frecuencia de muestreo
muestras=ws/wd;               % Muestras por ciclo
% muestras/ciclo = 53
% magnitud del polo dominante
MagPolo=exp(((-2*pi*cita)/sqrt(1-(cita^2))).*(wd/ws));
AngPolorad=(2*pi).*(wd/ws);                  % angulo del polo dominante en
radianes
AngPolograd=(AngPolorad*180)/(pi);      % angulo del polo dominante en
grados
zpole=MagPolo*exp(j*AngPolorad);       % polo dominante
disp('El valor del Polo Dominante es: ')
disp(zpole)
% polo dominante : 0.8130 + 0.0969i
%***************************************************************************
% (b) Discretizar la planta
% Gp(s) = 2/(s+10)(s+2)
gps = zpk([],[-10 -2],2);
gps=tf(gps);
gpztf=c2d(gps,T,'ZOH');   % Función de Transferencia Planta con Retenedor
```

```
gpz=zpk(gpztf);
pgpz=pole(gpz);              % Polos en Gp(z)
zgpz=zero(gpz);              % Zeros en Gp(z)

%            9.9601e-007 (z+0.996)
% Gp(z) = -----------------------------
%              (z-0.998) (z-0.99)

%*************************************************************
% METODO CANCELACION DE POLOS :
% SE HACE EL CERO DEL CONTROLADOR alfa IGUAL
% AL POLO DE LA PLANTA MAYOR CERCANO A UNO
%*************************************************************

% Hallar alfa
alfa=pgpz(1)
disp(' ');
disp('El valor de Alfa es: ')
disp(alfa)
% alfa = 0.998
%*********************************************************
% (C) HALLAR beta POR CONDICION DE ANGULO
```

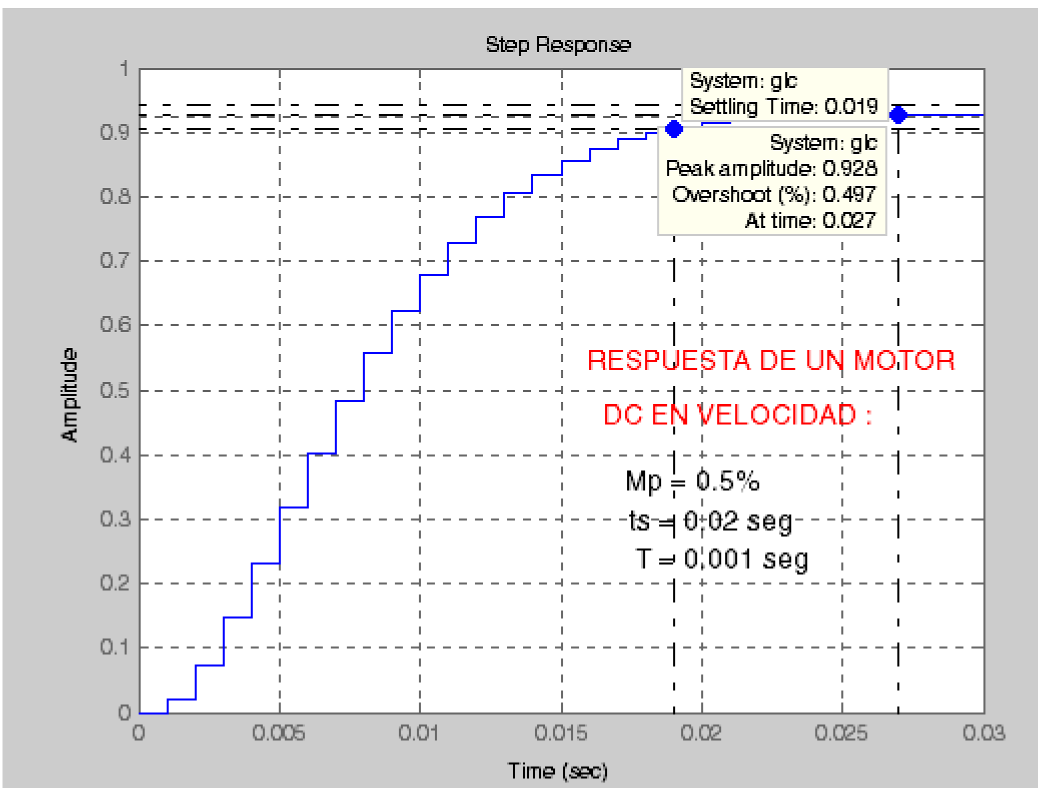


$$\theta - \varphi 1 - \varphi 2 = -180^{o}$$

```
% La funcion de transferencia en lazo abierto tiene
% un cero en z = -0.996 y dos polos en z = 0.99 y z = beta
% teta1 - fi1 - fi2 = -180º

teta1_rad=atan(imag(zpole)./(abs(zgpz)+real(zpole)));
teta1_grad=(teta1_rad*180)/pi;
fi1_rad= pi-atan(imag(zpole)/(pgpz(2)-real(zpole)));
```

```
fi1_grad=(fi1_rad*180)/pi;
fi2_grad=teta1_grad+180-fi1_grad;
fi2_rad=teta1_rad+pi-fi1_rad;
% teta1(grados) = 3.1º
% fi1(grados) = 151.3º
% fi2(grados) = 31.7º
beta=real(zpole)-imag(zpole)/tan(fi2_rad);
% beta = 0.6564
disp('  ');
disp('El valor de Beta es: ')
disp(beta)
%**************************************************************
% Hallar k por condicion de magnitud
% k*abs(Gc(z)*G(z)) = 1 en el polo dominante

gcz=zpk(alfa,beta,1,T);                    % Gc(z) con alfa y beta
gla=gcz*gpz; % Gla(z)=Gc(z)*Gp(z)

%               9.9601e-007 (z-0.998) (z+0.996)
% Gla(z) = -------------------------------------------
%                 (z-0.6564) (z-0.99) (z-0.998)

%  gla no aparece simplificada, no se cancela el polo con el cero
%  entonces se debe hacer
[z1,p1,k1]=zpkdata(gla,'v')
i=find(p1==alfa);
p1(i)=[ ]
i=find(z1==alfa);
z1(i)=[ ];
gla= zpk(z1,p1,k1,T)
gla_tf=tf(gla);                           % Convierto a Modelo tf
[num,den]=tfdata(gla_tf,'v');        % Extraigo los valores
gla1=poly2sym(num,'z')/poly2sym(den,'z');
z=zpole;
mag = abs(eval(gla1));
k=1/mag;
% k = 2.0593e+004
disp('  ');
disp('El valor de k es: ')
disp(k)

% (D) FUNCION DE TRANSFERENCIA DEL CONTROLADOR
disp(' El controlador es : Gc(z) = ')
gczz=zpk(alfa,beta,k,T)

%             20593.3092 (z-0.998)
%Gc(z) = ---------------------------
```

```
%                    (z-0.6564)

% (E) RESOUESTA AL PASO UNITARIO
glaz=k*gla;
glc=feedback(glaz,1);
grid
step(glc)
grid
% Comprobar el polo dominante
p=pole(glc);
disp('  ');
disp('Los polos en Lazo Cerrado del Sistema son: ')
disp(p)
% overshoot en porcentaje
Mp = exp(-cita*wn*pi/wd)*100
% Mp = 0.5%
```

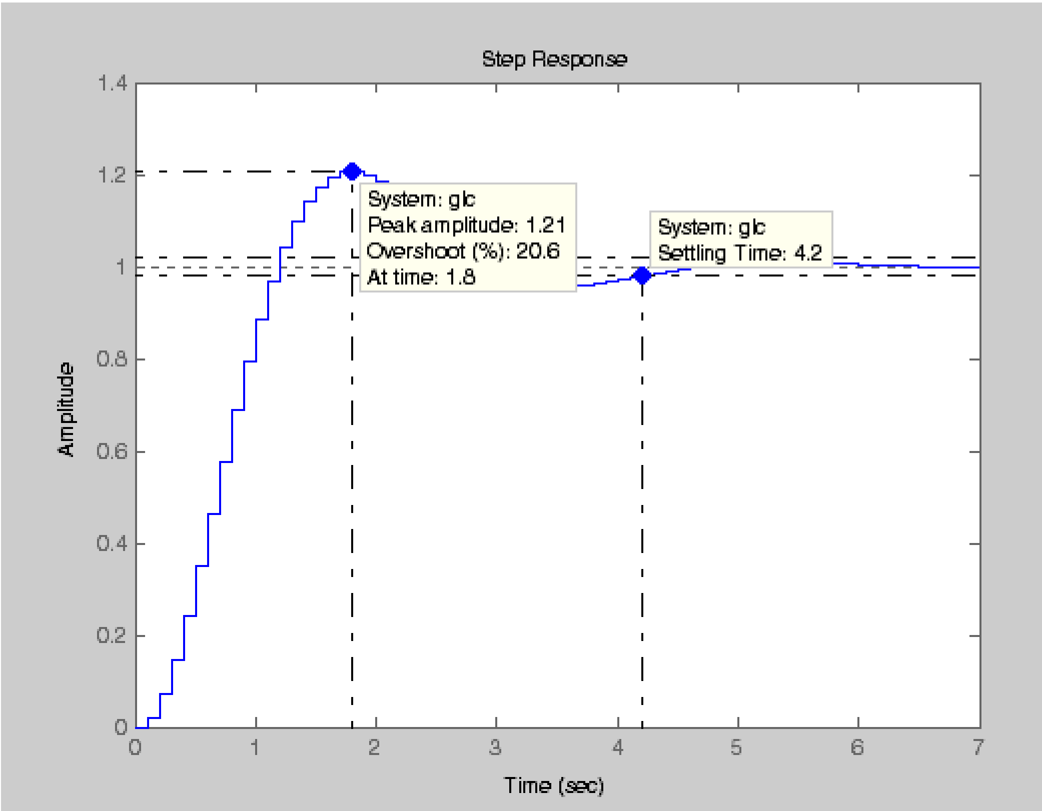


8.2 CONTROLADOR PI

```
clear all
home
disp(' ');
disp('*************************************************************')
disp('DISEÑO DE SISTEMAS DE CONTROL DISCRETO');
disp('DISEÑO EN EL DOMINIO DEL TIEMPO');
disp('DISEÑO DE UN CONTROLADOR PI');
disp('*************************************************************')
% FUNCION DE TRANSFERENCIA DE LA PLANTA Y DEL CONTROLADOR :
%                  10                                           z - alfa
% Gp(s)= ---------------------               Gpd(z)= k * ------------
%            (s+1)(s+2)                                          z - 1
%
%                    T                          ki*T-2*kp
% k = kp + ki * ---                alfa = - ----------------
%                    2                           ki*T+2*kp
% kp=Constante Proporcional
% ki=Constante Integral

%*****************************************************************************

T=input('Digite el tiempo de muestreo "T" (seg): ');
%  T=0.1;

% (A)DISCRETIZAR PLANTA
gps=zpk([],[-1 -2],10);
gpz=c2d(gps,T,'ZOH'); % Planta en el Dominio Z
%             0.04528 (z+0.9048)
% Gp(z) = ---------------------------
%            (z-0.9048) (z-0.8187)
%*****************************************************************************
% La planta discretizada no tiene polo en z =1 , o sea ess (error de estado
% estacionario no es cero, por tanto el controlador Gpi debe adicionar
% este polo y cancelar el polo mas cercano a z=1 de la planta adicionando
% el correspondiente cero elevando asi el margen de estabilidad del sistema
% y evitando el incremento del orden
%*****************************************************************************
pgpz=pole(gpz);           % Polos en Gp(z)
zgpz=zero(gpz);           % Zeros en Gp(z)
n=length(pgpz);
% n = 2

% (B) HALLAR ALFA por metodo de Cancelacion de Polos y Ceros
% El polo mas cercano a 1 se cancela con el cero del controlador (alfa)
polos= sort( pgpz)
% polos = 0.8187, 0.9048
% alfa es el polo mayor
```

```
alfa = polos(n)
% alfa = 0.9048

% (C) FUNCION DE TRANSFERENCIA DEL CONTROLADOR
% Gpi = kp + ki*(T/2)*((z+1)/(z-1))
% Gpi = k* (z-alfa)/ (z-1)
% k = kp + ki*(T/2)
% alfa = -(ki*T - 2*kp) / (ki*T + 2*kp)
syms ki kp
k1 = kp/ki;
x = solve('alfa=-(T-2*k1)/(T+2*k1)');
k1=eval(x);

% Simulacion para encontrar el valor de kp que encuentre un Mp = 20%
kp = 0.1;              % valor inicial de kp
mp = 10;
while mp<20
ki = kp/k1;
k = kp + ki*(T/2);
gpi = zpk(alfa,1,k,T)
% El polo mayor de la planta discreta se reemplaza por el cero del
controlador
% p1(1) es el polo mayor que se elimina
% eliminar el polo mayor de la planta
[z1,p1,k1] = zpkdata(gpz,'v');
p1 = sort(p1);
lon = length(p1);
p1(lon)=[ ];
gpz1 = zpk(z1,p1,k1,T);
% eliminar el cero del controlador
[z2,p2,k2] = zpkdata(gpi,'v');
gpi1 = zpk([ ],p2,k2,T);
% funcion de transferencia del lazo abierto y cerrado
gla = gpi1*gpz1
glc = feedback(gla,1)
[wn,cita]=damp(glc)
wd=wn*sqrt(1-cita(1)^2)
mp1=exp(-cita*wn(1)*pi/wd(1))
mp = mp1*100;
ts = 4/(cita(1)*wn(1));
kp = kp+0.01;
end
% Mp = 20.61%
disp(' ');
disp('Constante Proporcional: ')
disp(kp);
disp('Constante Integral: ')
```

```
disp(ki);
disp('controlador digital: ')
display(gpi)
%                  0.42085 (z-0.9048)
% Gpi(z) = -------------------------
%                        (z - 1)
% Respuesta al escalon
step(glc)
```

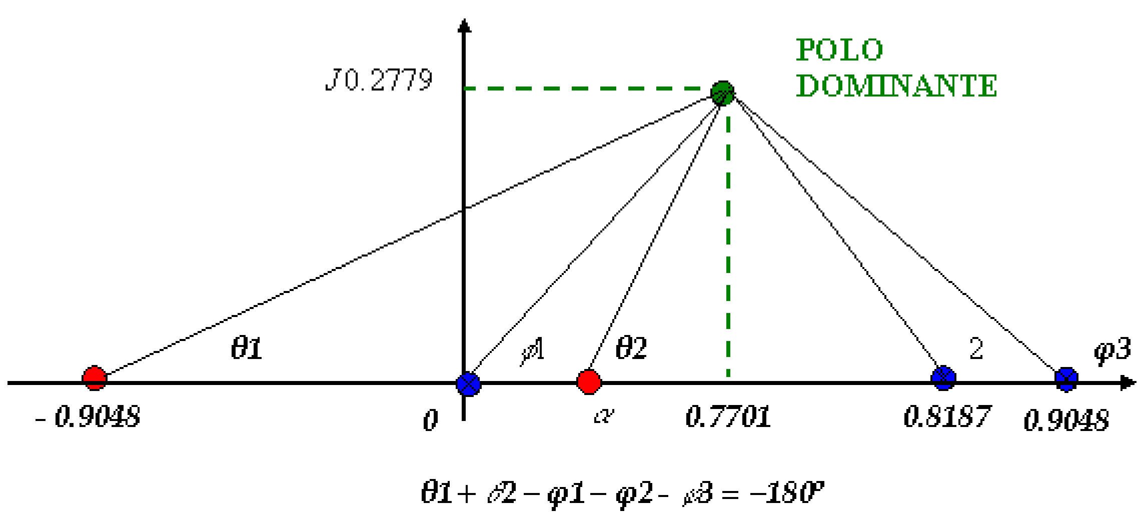


8.3 CONTROLADOR PD

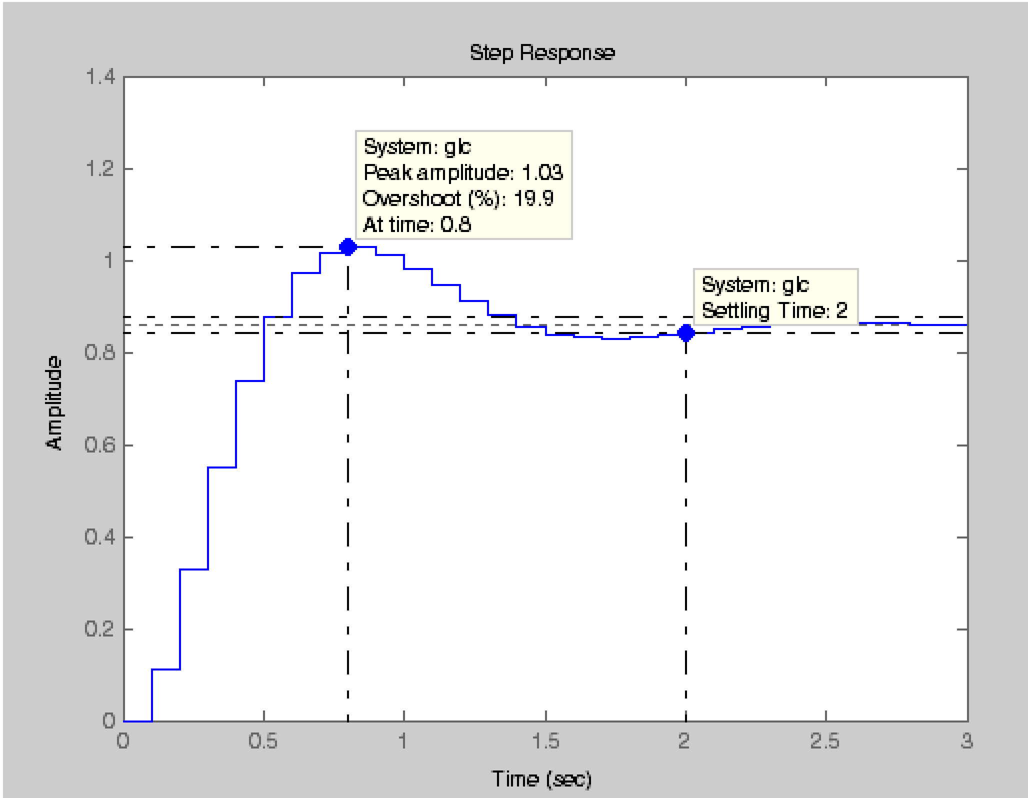
```
clear all
home
disp(' ');
disp('*****************************************************************')
disp('DISEÑO DE SISTEMAS DE CONTROL DISCRETO');
disp('DISEÑO EN EL DOMINIO DEL TIEMPO');
disp('DISEÑO DE UN CONTROLADOR PD');
disp('*****************************************************************')
% FUNCION DE TRANSFERENCIA DE LA PLANTA Y DEL CONTROLADOR :
```

```
%                 1                                      z - alfa
% Gp(s)= ---------------            Gpd(z)= k * ----------
%            (s+1)(s+2)                                  z
%
%       kp*T + kd                           kd
% k = ---------------              alfa = ---------------
%             T                          kp*T + kd
% kp=Constante Proporcional
% kd=Constante Derivativa

%*************************************************************
disp('  ');
ts=input('Digite el tiempo de Establecimiento "ts" (seg): ');
%  ts=2;
disp('  ');
mp=input('Digite el Porcentaje de SobreImpulso "Mp" (%): ');
%  mp=16.3;
mp=mp/100;
disp('  ');
T=input('Digite el tiempo de muestreo "T" (seg): ');
%  T=0.1;
disp('  ');
disp('  ');

% (A) DETERMINAR EL POLO DOMINANTE DEL SISTEMA EN LAZO
CERRADO
% ts = 4/sigma 2% tolerancia
sigma=4/ts;
% Mp = exp(-sigma*pi/wd)
wd = eval(solve('mp=exp(-sigma*pi/wd)'))
% polos del sistema continuo
s1=-sigma+j*wd;
s2=conj(s1);
ws=(2*pi)/T;        % Frecuencia de Muestreo
% numero de muestras por ciclo
muestras=ws/wd;
zpole=exp(T*s1);
disp('  ');
disp('El valor del Polo Dominante es: ')
disp(zpole)
% POLO DOMINANTE = 0.7701 + 0.2779i

% (B) FUNCION DE TRANSFERENCIA DE LA PLANTA DISCRETIZADA

gps=zpk([],[-1 -2],1);
gpz=c2d(gps,T,'ZOH');           % Planta en el Dominio Z
pgpz=pole(gpz);                  % Polos en Gp(z)
```

```
zgpz=zero(gpz);                     % Zeros en Gp(z)
% 0.004528 (z+0.9048)
% Gp(z) = ---------------------
% (z-0.9048) (z-0.8187)

% (C) APLICAR CONDICION DE ANGULO
% sumatoria angulos de ceros - Sumatoria angulos de polos = -180°
% ceros de la planta : -0.9048 (teta1)
% polos de la planta : [0.9048 (fi3) 0.8187 (fi2)]
% ceros del controlador : alfa (teta2)
% polos del controlador : 0 (fi1)
% polo dominante : 0.7701 + 0.2779i
% teta1 + teta2 -(fi1+fi2+fi3) = -180º
% teta(n) --> Angulo Zero respectivo

teta1_rad=atan(imag(zpole)./(abs(zgpz)+real(zpole)));
teta1_grad=(teta1_rad*180)/pi;
fi1_rad=angle(zpole);
fi1_grad=(fi1_rad*180)/pi;
fi2_rad=pi - atan(imag(zpole)./(abs(pgpz(1,1))-real(zpole)));
fi2_grad=(fi2_rad*180)/pi;
fi3_rad=pi - atan(imag(zpole)./(abs(pgpz(2,1))-real(zpole)));
fi3_grad=(fi3_rad*180)/pi;
teta2_grad=-180-teta1_grad+fi1_grad+fi2_grad+fi3_grad;
teta2_rad=(teta2_grad*pi)/180;
% teta1 = 9.4221º, fi1 = 19.8456º, fi2 = 115.8610º, fi3 = 99.9228º
```



```
alfa=real(zpole)-(imag(zpole)/tan(teta2_rad));
disp(' ');
disp('El valor de Alfa es: ')
disp(alfa)
% alfa = 0.5036

% (D) APLICAR CONDICION DE MAGNITUD abs(Gpz*Gcz) = 1

gcz=zpk(alfa,0,1,T);
gla=gcz*gpz;
gla_tf=tf(gla);
[num,den]=tfdata(gla_tf,'v');
gla1=poly2sym(num,'z')/poly2sym(den,'z');
z=zpole;
mag = abs(eval(gla1));
k=1/mag;
disp(' ');
disp('El valor de k es: ')
disp(k)
% k = 24.1058
gczz=zpk(alfa,0,k,T);
% El controlador es :
%              24.1058 (z-0.5036)
% Gc(z) = --------------------------
%                    z

% (E) RESPUESTA AL PASO UNITARIO
glaz=gczz*gpz;
glc=feedback(glaz,1);
grid
step(glc)
p=pole(glc);
disp(' ');
disp('Los polos en Lazo Cerrado del Sistema son: ')
disp(p)
% Los polos en Lazo Cerrado del Sistema son:
% [ 0.0742 0.7701 + 0.2779i 0.7701 - 0.2779i]
% Notese que el polos 0.0742 esta cerca al origen y por lo tanto no es
dominante
```

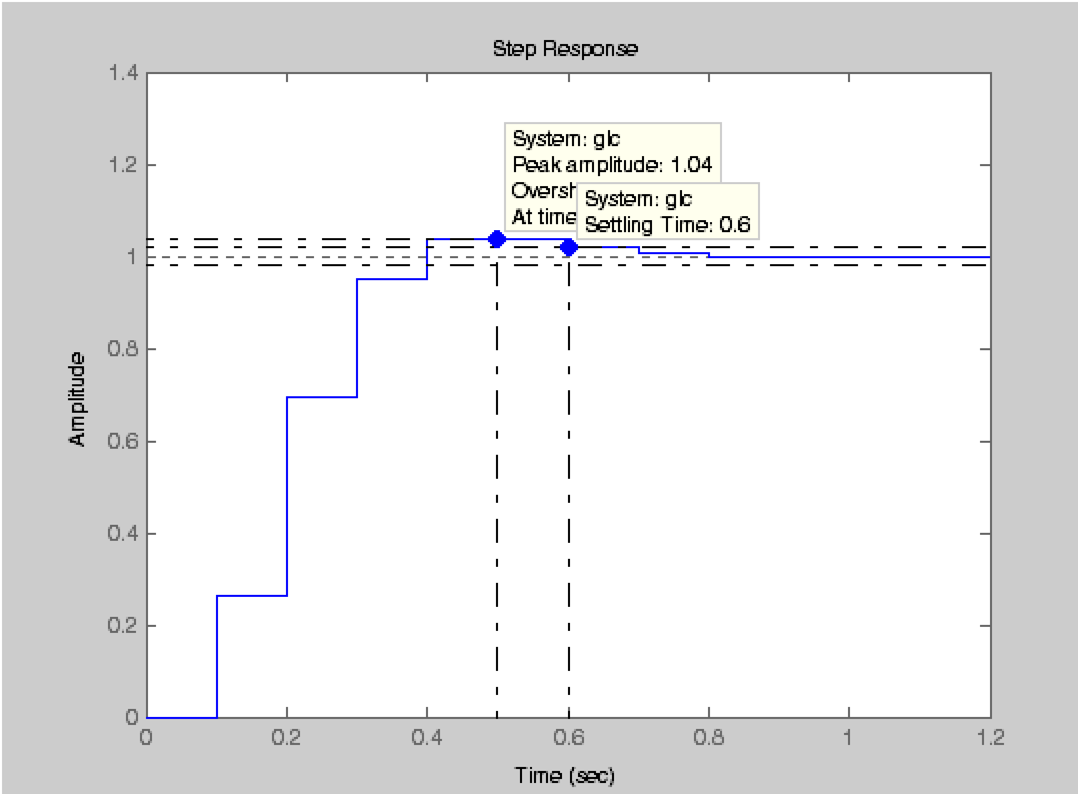


8.4 CONTROLADOR PID

```
clear all
home
disp(' ');
disp('**************************************************************')
disp('DISEÑO DE SISTEMAS DE CONTROL DISCRETO');
disp('DISEÑO EN EL DOMINIO DEL TIEMPO');
disp('DISEÑO DE UN CONTROLADOR PID');
disp('**************************************************************')
% FUNCION DE TRANSFERENCIA DE LA PLANTA Y DEL CONTROLADOR :
%
%                 10                              z^2 + a*z + b
% Gp(s)= ----------------      Gpid(z)= k * -------------------
%            (s+1)(s+2)                          z*(z - 1)
%
%         ki*T^2 + 2*kp*T + 2*kd
% k = -----------------------------------
%                  2*T
%         ki*T^2 - 2*kp*T - 4*kd
% a = -------------------------------------
%         ki*T^2 + 2*kp*T + 2*kd
```

```
%                     2*kd
% b = -------------------------------------
%         ki*T^2 + 2*kp*T + 2*kd
%
% kp=Constante Proporcional
% kd=Constante Derivativa
% ki=Constante Integral
%*****************************************************
disp('  ');
kv=input('Digite la Constante de error de Velocidad "kv": ');
% kv=5;
disp('  ');
T=input('Digite el tiempo de muestreo "T" (seg): ');
% T=0.1;
% Error en estado estacionario
ess=1/kv;
% ess = 0.2 no es nulo se debe corregir con un PID

% (A)DISCRETIZAR PLANTA
gps=zpk([],[-1 -2],10);
gpz=c2d(gps,T,'ZOH');           % Planta en el Dominio Z
pgpz=pole(gpz);                  % Polos en Gp(z)
zgpz=zero(gpz);                  % Zeros en Gp(z)

% (B) HALLAR ki ENCONTRANDO EL LIMITE PARA EL VALOR DADO DE kv
syms ki kd kp z
gpz_tf=tf(gpz);
[num1,den1]=tfdata(gpz_tf,'v');
num2=poly2sym(num1,'z');
den2=poly2sym(den1,'z');
gz=num2/den2;
gpid_1=kp + ki*(T/2)*((z+1)/(z-1)) + (kd/T)*((z-1)/z);
gla_1=(gz*gpid_1);
lim=limit(((z-1)/z)*(gla_1/T),z,1);
% kv = lim, entonces
x = lim-kv;
ki = solve(x)
ki=eval(ki)

% (C) HALLAR kp, kd POR CANCELACION DE POLOS
syms kp kd
[kd,kp]=solve('2*kd/(ki*T^2+2*kp*T+2*kd)=den1(3)','(ki*T^2-2*kp*T-
4*kd)/(ki*T^2+2*kp*T+2*kd) = den1(2)','kp','kd')
kp = eval(kp)
kd = eval(kd)
% kp = 1.4125, kd = 0.4295
disp('  ');
```

```
disp('Constante Proporcional: '),disp(kp);
disp('Constante Derivativa: '),disp(kd);
disp('Constante Integral: '),disp(ki);

% (D) FUNCION DE TRANSFERENCIA DEL CONTROLADOR
% Gpid(z) = k*(z^2 + a z +b) /(z (z - 1))
a = (ki*T^2-2*kp*T-4*kd)/(ki*T^2+2*kp*T+2*kd);
b = 2*kd/(ki*T^2+2*kp*T+2*kd);
k = (ki*T^2+2*kp*T+2*kd)/(2*T);
gpid = tf([k a*k b*k],[1 -1 0],T);
gpid = zpk(gpid)

%                5.7971 (z-0.9048) (z-0.8187)
% Gpid(z) = --------------------------------------
%                       z (z - 1)

% (E) RESPUESTA AL PASO UNITARIO
% los ceros del controlador se simplifican con los polos de la planta
discretizada
[z1,p1,k1] = zpkdata(gpid,'v');
% eliminar ceros del controlador
gpid1 = zpk([],p1,k1,T);
[z2,p2,k2] = zpkdata(gpz,'v');
% eliminar polos de la planta discreta
gpz1 = zpk(z2,[],k2,T);
gla = gpid1*gpz1;
glc = feedback(gla,1)
polos = pole(glc)
% Polo dominante : 0.3688 + 0.3186i
step(glc)
% ts = 0.6 seg, Mp = 3.8 %
```



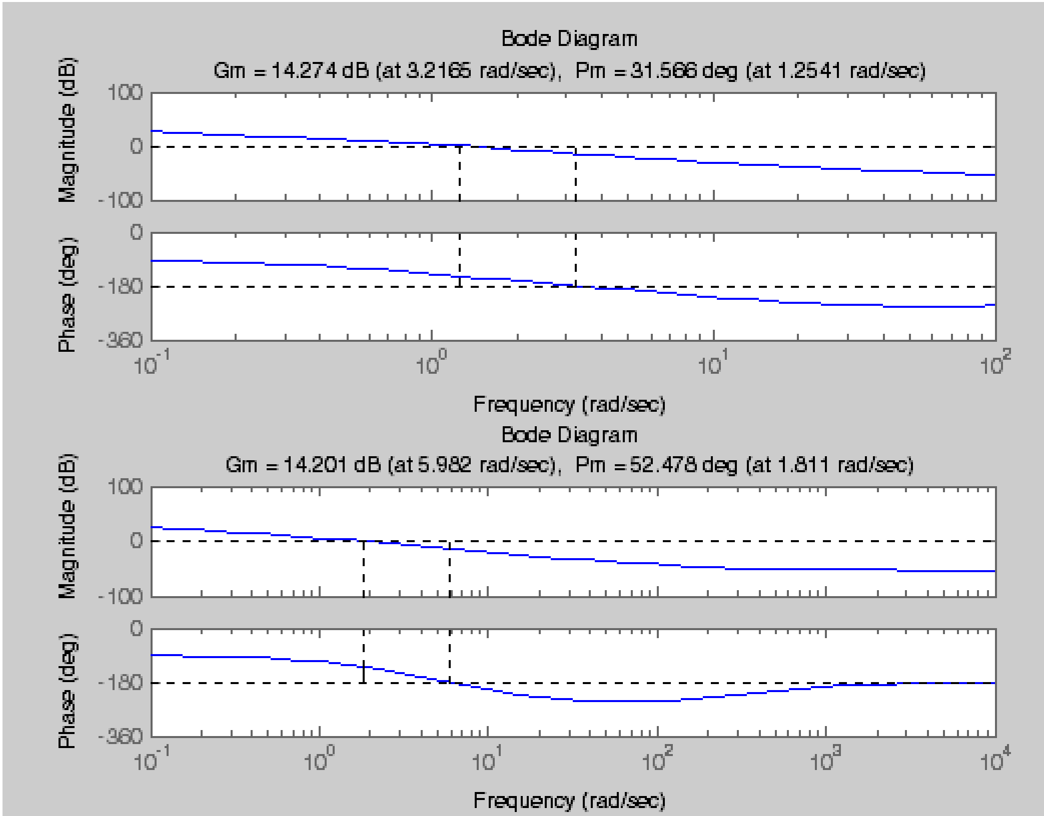
9. DISEÑO: DOMINIO -FRECUENCIA

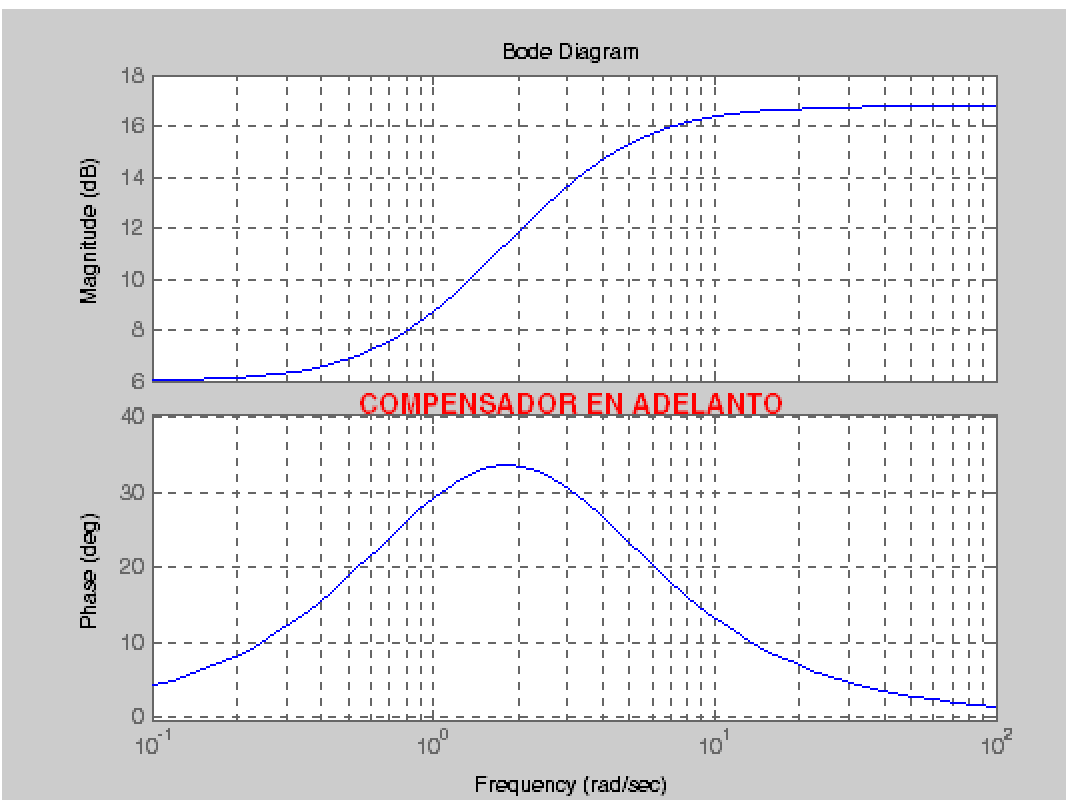
9.1 COMPENSADOR EN ADELANTO

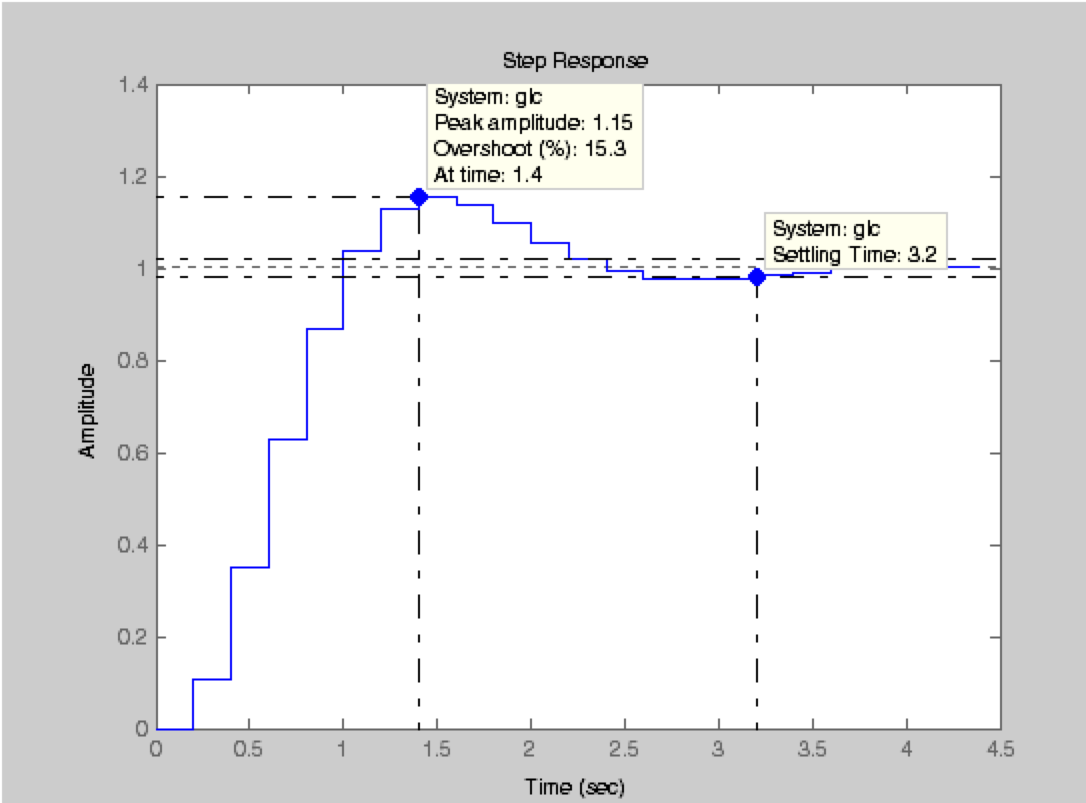
```
%*******************************************************************************
% Diseñar un compensador en adelanto para la planta Gp(s)=1/s(s+1)
% de tal forma que el m,argen de fase sea de 50º y la constante de
% de velocidad kv sea de 2/seg. El tiempo de muestreo es de 0.2 seg.
%*******************************************************************************
clear all
home
disp('DISEÑO DE UN COMPENSADOR DE FASE EN ADELANTO')
kv=2;
pms=50;
gp=zpk([ ],[0 -1],1);
T=0.2;
gz=c2d(gp,T,'zoh');
```

```
gz=tf(gz);
% a)Encontrar el valor de k para que kv=2
syms w tao alfa k
[numz,denz]=tfdata(gz,'v');
num=numz.*[1 -1 1];
den=denz.*[1 -1 1];
[numv,denv]=bilinear(num,den,0.5);
v=-T/2;
numw=numv.*[v^2 v 1];
denw=denv.*[v^2 v 1];
gw=tf(numw,denw);
gw1=zpk(gw);
[z p k]=zpkdata(gw1,'v');
% Encontrar el valor de k
p=sort(p)
n=length(p)
if p(n)<abs(1e-5)
p(n)=0
else
p(n)=p(n)
end
gw1=zpk(z,p,k);
gw1=tf(gw1);
[numw,denw]=tfdata(gw1,'v');
%gw1= -0.00033201*(w+300.2)*(w-10)/(w*(w+0.9997));
gw2=poly2sym(numw,'w')/poly2sym(denw,'w');
f=limit(w*(1+tao*w)/(1+alfa*tao*w)*gw2,w,0);
k=kv/f;
k=eval(k);
gw2=k*gw1
subplot(211)
margin(gw2)
[gm pm]=margin(gw2)
%Margen de ganancia=14.274 dB, margen de fase=31.566º

% B)Determinar el valor de phi del compensador
delta=5;
pm1=pm;
while pm1<pms
phi=pms-pm+delta;
phi=phi*pi/180;
alfa=(1-sin(phi))/(1+sin(phi));
ralfa=sqrt(alfa);
% gcmax=0.5365

% C) determinar el valor de wmax para gcmax=ralfa
ralfa1=ralfa-0.003;
```

```
ralfa2=ralfa+0.003;
w=0.1:0.01:100;
[mag,fase,w]=bode(gw2,w);
i=find(mag>ralfa1&mag<ralfa2)
w(i);
mag(i);
gcmax=mag(i);
wmax=w(i);
%wmax=1.82 rad/seg

% D) Encontrar la funcion del compensador gc(w)
wz=wmax*gcmax;
wp=wmax/gcmax;
kc=k/alfa;
gcw=zpk(-wz,-wp,kc);
gla=gcw*gw;
subplot(212)
margin(gla)
[gm1 pm1]=margin(gla);
delta=delta+5;
end              % end de while
% wz=0.9764 rad/seg, wp=3.3923 rad/seg
% gm1=5.1292, pm1=52.4783º

% E) Encontrar la funcion del compensador digital gc(z)
fs=1/T;
[numw denw]=tfdata(gcw,'v');
[numz denz]=bilinear(numw,denw,fs);
gcz=tf(numz,denz,T)
gcz=zpk(gcz)
gla=gz*gcz;
glc=feedback(gla,1);
figure
step(glc)
%              5.6619 (z-0.8221)
%   Gc(z)= -----------------------
%                  (z-0.4934)
```


Bode Diagram
Gm = 14.274 dB (at 3.2165 rad/sec), Pm = 31.566 deg (at 1.2541 rad/sec)
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)
Bode Diagram
Gm = 14.201 dB (at 5.982 rad/sec), Pm = 52.478 deg (at 1.811 rad/sec)
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)


Bode Diagram
Magnitude (dB)
COMPENSADOR EN ADELANTO
Phase (deg)
Frequency (rad/sec)

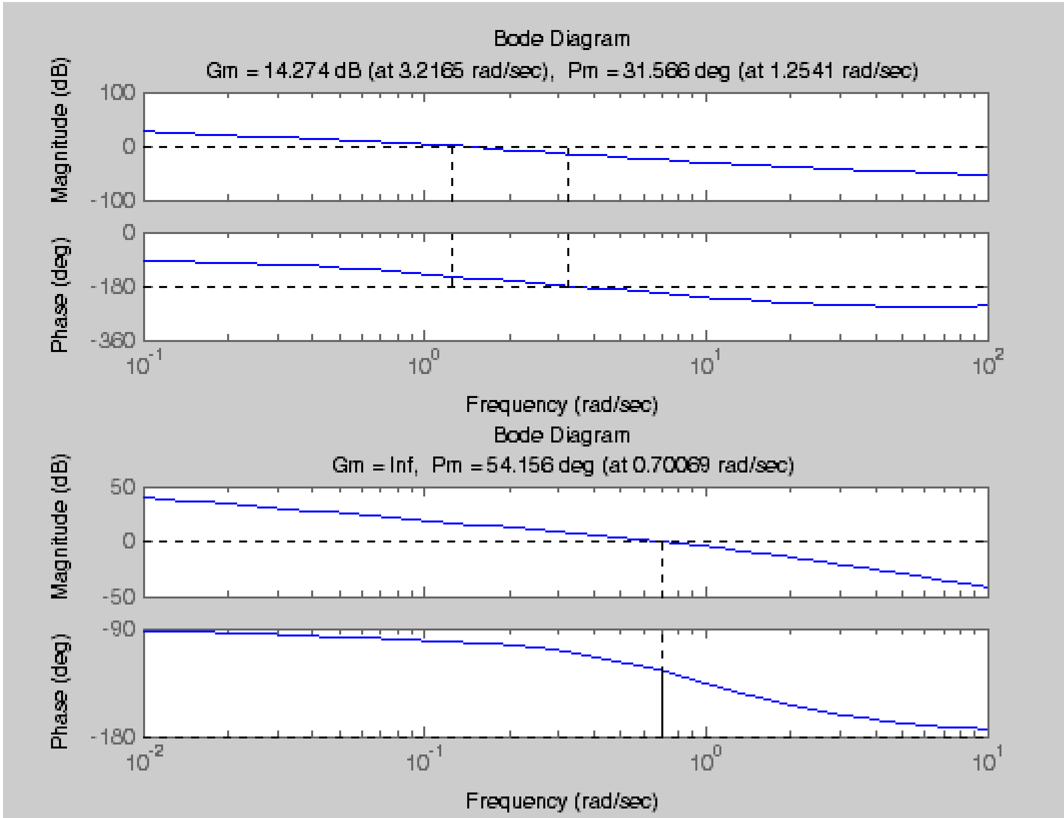


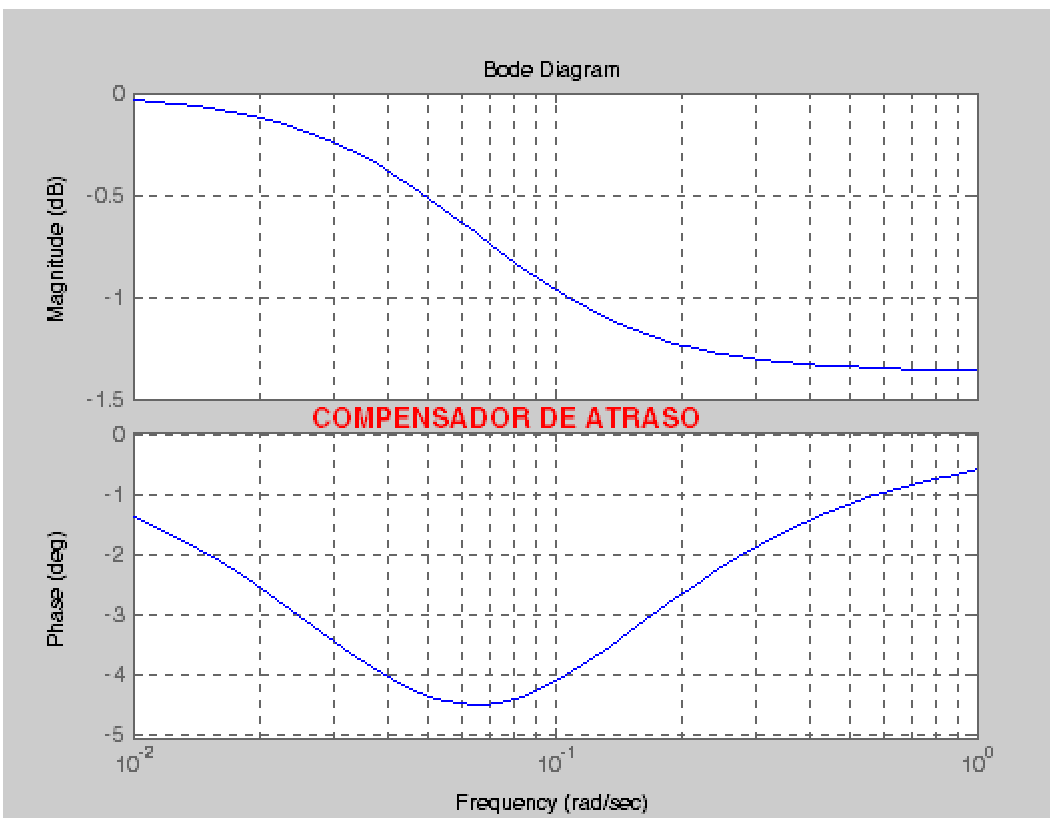
14.2 COMPENSADOR EN ATRASO

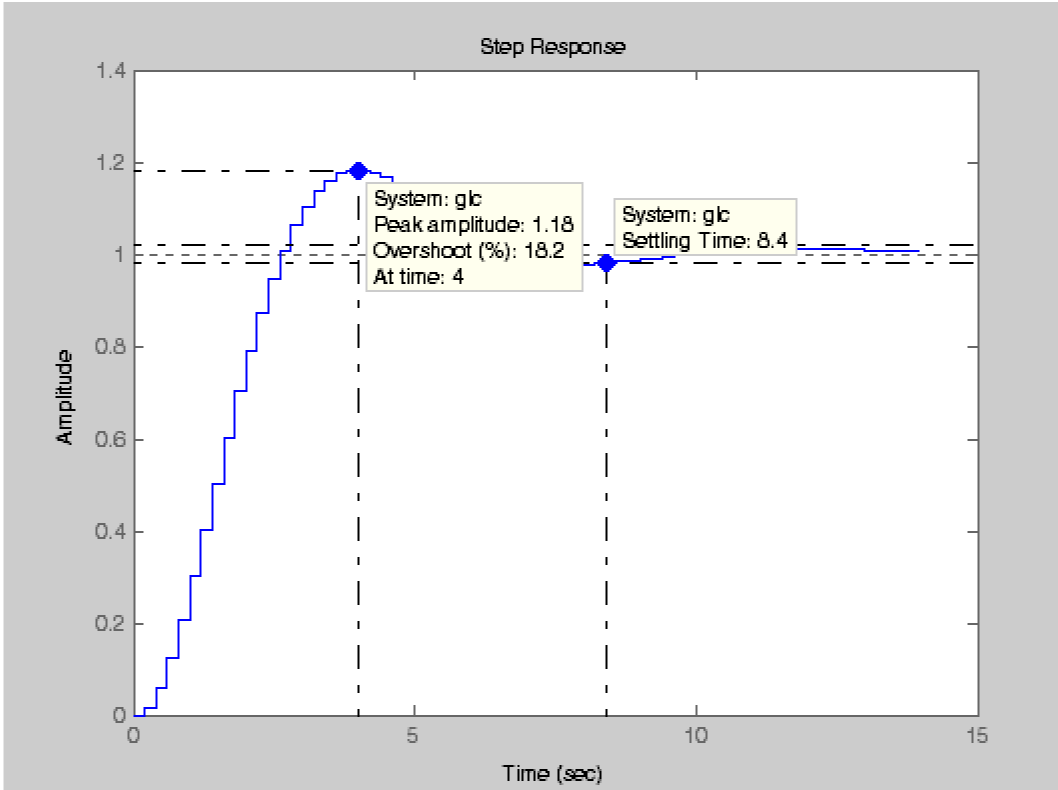
```
%***************************************************************************
% Diseñar un compensador en atraso para la planta Gp(s)=1/s(s+1)
% de tal forma que el margen de fase sea de 50º y la constante de
% de velocidad kv sea de 2/seg. El tiempo de muestreo es de 0.2 seg.
%***************************************************************************
clear all
home
disp('DISEÑO DE UN COMPENSADOR DE FASE EN ATRASO')
kv=2;
pms=50;
T = 0.2;
gp=zpk([ ],[0 -1],1);
gz=c2d(gp,T,'zoh');
gz=tf(gz);
%              0.01873 z + 0.01752
% G(z) = -------------------------------
%             z^2 - 1.819 z + 0.8187

% A) Valor de ganacia para cumplir con el kv
```

```
syms w tao alfa k
[numz,denz]=tfdata(gz,'v');
num=numz.*[ 1 -1 1];
den=denz.*[ 1 -1 1];
[numv,denv]=bilinear(num,den,0.5);
v=-T/2;
numw=numv.*[v^2 v 1];
denw=denv.*[v^2 v 1];
gw=tf(numw,denw);
gw1=zpk(gw);
[z p k]=zpkdata(gw1,'v');
%Encontrar el valor de k
p=sort(p)
n=length(p)
if p(n)<abs(1e-5)
p(n)=0
else
p(n)=p(n)
end
gw1=zpk(z,p,k);
gw1=tf(gw1);
[numw,denw]=tfdata(gw1,'v');
gw2=poly2sym(numw,'w')/poly2sym(denw,'w');
f=limit(w*(1+tao*w)/(1+alfa*tao*w)*gw2,w,0);
k=kv/f;
k=eval(k);
% k = 640
gw2=k*gw1
subplot(211)
margin(gw2)
[gm pm]=margin(gw2)
% gm = 5.1723, pm = 31.5664º

% B) A partir de Bode determinar el ángulo que
% requiere el compensador para cumplir con el
% margen de fase del sistema

delta=5;
pm1=pm;
while pm1<pms
phi=-(180-pms-delta);
phi1=phi-0.1;
phi2=phi+0.1;
w=0.1:0.01:100;
[mag,fase]=bode(gp,w);
% c) Ganancia máxima del compensador
i=find(fase>phi1 & fase<phi2);
```

```
% valores comprendidos entre phi1 y phi2
gcmax=mag(i);
% Parámetros intermedios del controlador
wmax=w(i)
% Valores de frec de corte wz y wp
wz=wmax(1,1)/10;
wp=wz(1,1)/gcmax(1,1);
kc=wp/wz;
% Función de transferencia del controlador Gc(s)
gcw=zpk(-wz,-wp,kc);
gla=gcw*gp;             % Lazo abierto = Planta * Controlador
subplot(212);            % Ubicacion de la Grafica
margin(gla)             % Bode del sistema compensado
% Valores de Margen de magnitud y fase
[gm1,pm1]=margin(gla);
delta=delta+5;          % Incremento de delta
end % Fin del Bucle
% pm1 = 54.1º, wz = 0.0700, wp = 0.0598
%              0.85446 (w+0.07)
%  Gc(w) = ------------------------
%                (w+0.05981)
% E) Encontrar la funcion del compensador digital de atraso gc(z)
fs=1/T;
[numw denw]=tfdata(gcw,'v');
[numz denz]=bilinear(numw,denw,fs);
gcz=tf(numz,denz,T);
gcz=zpk(gcz)
gla=gz*gcz;
glc=feedback(gla,1);
figure
step(glc)
%               0.85532 (z-0.9861)
%    Gc(z) = -------------------------
%                    (z-0.9881)
```


Bode Diagram
Gm = 14.274 dB (at 3.2165 rad/sec), Pm = 31.566 deg (at 1.2541 rad/sec)
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)
Bode Diagram
Gm = Inf, Pm = 54.156 deg (at 0.70069 rad/sec)
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)


Bode Diagram
Magnitude (dB)
COMPENSADOR DE ATRASO
Phase (deg)
Frequency (rad/sec)



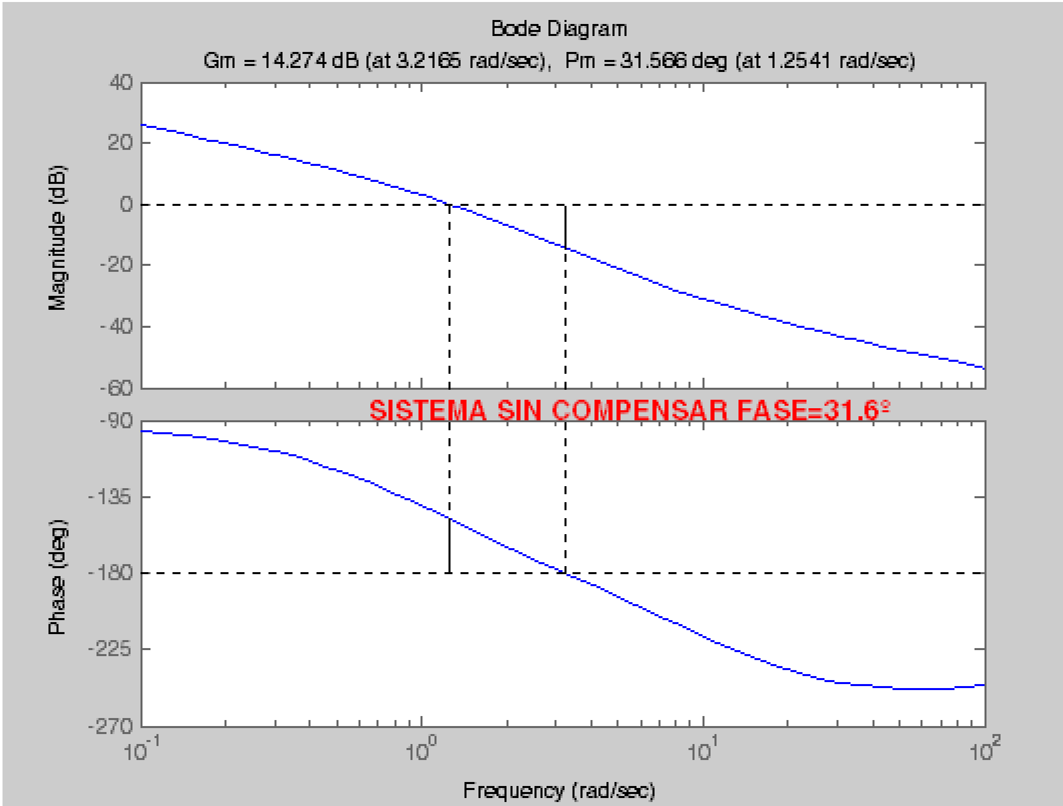
14.3 COMPENSADOR ATRASO-ADELANTO

```
%*********************************************************************************
% Diseñar un compensador en atraso-adelanto para la planta Gp(s)=1 /s
(s+1)
% de tal forma que el margen de fase sea de 80º y la constante de
% de velocidad kv sea de 2/seg. El tiempo de muestreo es de 0.2 seg.
%*********************************************************************************
clear all
home
% PRIMERA PARTE : ENCONTRAR EL COMPENSADOR DE ADELANTO
disp('DISEÑO DE UN COMPENSADOR DE FASE EN ATRASO')
kv=2;
% Compensador de atraso compensa a 35º
pms=50;
T = 0.2;
gp=zpk([ ],[0 -1],1);
gz=c2d(gp,T,'zoh');
gz=tf(gz);
%            0.01873 z + 0.01752
```

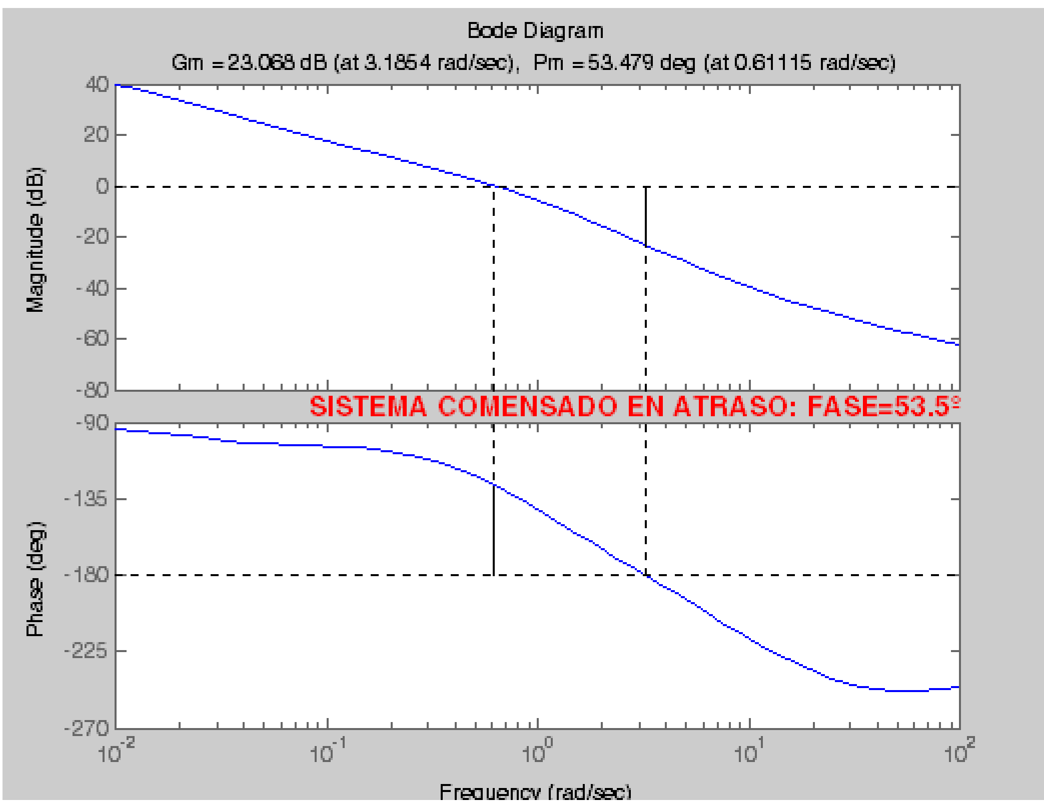
```
% G(z) = -----------------------------
%           z^2 - 1.819 z + 0.8187

% A) Valor de ganancia para cumplir con el kv
syms w tao alfa k
[numz,denz]=tfdata(gz,'v');
num=numz.*[ 1 -1 1];
den=denz.*[ 1 -1 1];
[numv,denv]=bilinear(num,den,0.5);
v=-T/2;
numw=numv.*[v^2 v 1];
denw=denv.*[v^2 v 1];
gw=tf(numw,denw);
gw1=zpk(gw);
[z p k]=zpkdata(gw1,'v');
%Encontrar el valor de k
p=sort(p);
n=length(p);
if p(n)<abs(1e-5)
p(n)=0;
else
p(n)=p(n);
end
gw1=zpk(z,p,k);
gw1=tf(gw1);
[numw,denw]=tfdata(gw1,'v');
gw2=poly2sym(numw,'w')/poly2sym(denw,'w');
f=limit(w*(1+tao*w)/(1+alfa*tao*w)*gw2,w,0);
k=kv/f;
k=eval(k);
% k = 2.00
gw2=k*gw1
figure
margin(gw2)
[gm pm]=margin(gw2)
% gm = 5.1723, pm = 31.5664º

% B) A partir de Bode determinar el ángulo que requiere
% el compensador de atraso para cumplir con el margen
% de fase del sistema de  50º

delta=5;
pm1=pm;
while pm1<pms
phi=-(180-pms-delta);
phi1=phi-0.2;
phi2=phi+0.2;
```

```
w=0.1:0.01:100;
[mag,fase]=bode(gw1,w);
% c) Ganancia máxima del compensador
i=find(fase>phi1 & fase<phi2);
% valores comprendidos entre phi1 y phi2
gwmax=mag(i);
% Parámetros intermedios del controlador
wmax=w(i);
% Valores de frec de corte wz y wp
wz=wmax(1,1)/10;
wp=wz(1,1)/gwmax(1,1);
kc=wp/wz;
% Función de transferencia del controlador Gc(s)
gcw=zpk(-wz,-wp,kc);
gla=gcw*gw1;              % Lazo abierto = Planta * Controlador
% Valores de Margen de magnitud y fase
[gm1,pm1]=margin(gla);
delta=delta+5;            % Incremento de delta
end                       % Fin de while
figure
margin(gla)               % Bode del sistema compensado

%pm1 = 53.4787º, wz = 0.0610, wp = 0.0435

%           0.71385 (w+0.061)
%Gc(w) = ------------------------ (atraso)
%             (w+0.04354)

% Se puede tener una nueva planta con funcion de
% transferencia igual a Gla y a esta se le aplica el compensador
% de adelanto para conseguir los 80º

% SEGUNDA PARTE : COMPENSADOR ATRASO-ADELANTO
disp('DISEÑO DE UN COMPENSADOR DE FASE EN ADELANTO')
pms=80;
[gm pm]=margin(gla)
%Margen de ganancia=14.2360 , margen de fase=53.4787º

% Determinar el valor de phi del compensador
delta=5;
pm1=pm;
while pm1<pms
phi=pms-pm+delta;
phi=phi*pi/180;
alfa=(1-sin(phi))/(1+sin(phi));
ralfa=sqrt(alfa);
% determinar el valor de wmax para gcmax=ralfa
```

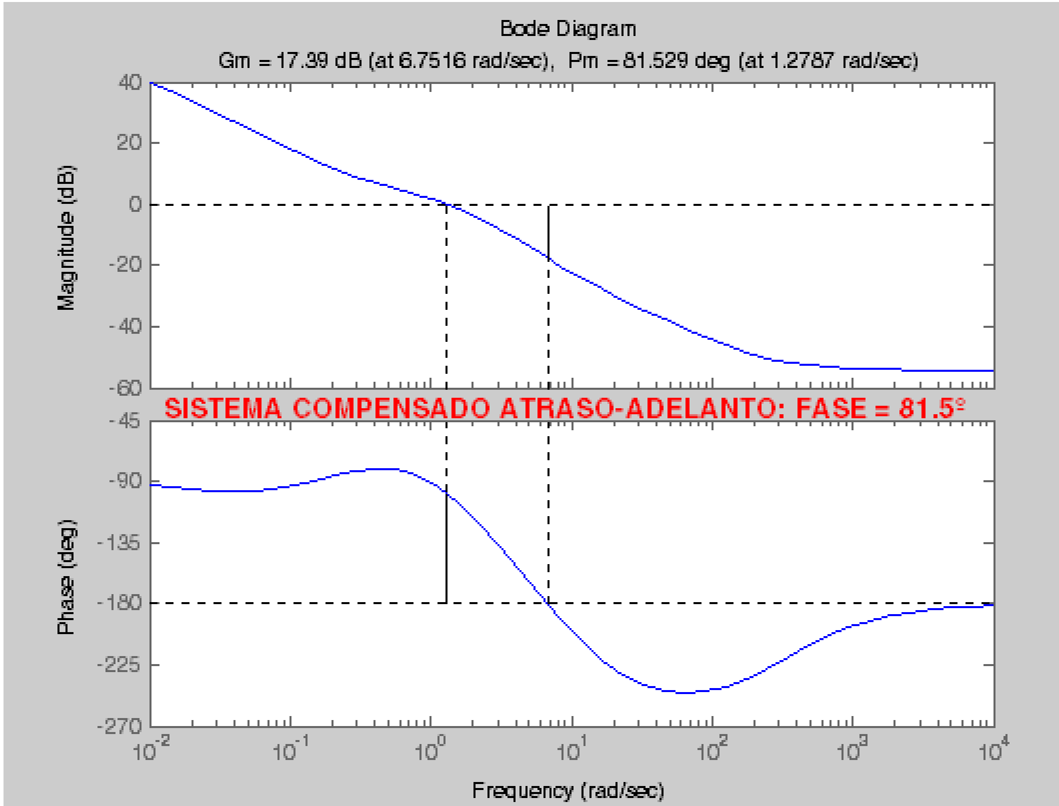
```
ralfa1=ralfa-0.005;
ralfa2=ralfa+0.005;
w=0.01:0.01:100;
[mag,fase,w]=bode(gla,w);
i=find(mag>ralfa1&mag<ralfa2);
w(i);
mag(i);
gwmax=mag(i);
wmax=w(i);
% Encontrar la funcion del compensador gc(w)
wz=wmax(1)*gwmax(1);
wp=wmax(1)/gwmax(1);
kc=1/alfa;
gcw=zpk(-wz,-wp,kc);
GlaSistema=gcw*gla;
[gm1 pm1]=margin(GlaSistema);
delta=delta+5;
end
figure
margin(GlaSistema)

% wz= 0.4445 rad/seg, wp= 3.6285 rad/seg
% gm1= 7.4044, pm1= 81.5288º

% E) Encontrar la funcion del compensador digital de atraso gc(z)
fs=1/T;
[numw denw]=tfdata(gcw,'v');
[numz denz]=bilinear(numw,denw,fs);
gcz=tf(numz,denz,T);
gcz=zpk(gcz)
glaz=gz*gcz;
glcz=feedback(glaz,1);
figure
step(glcz)

%            6.2917 (z-0.9149)
% Gc(z) = -----------------------
%             (z-0.4675)
```

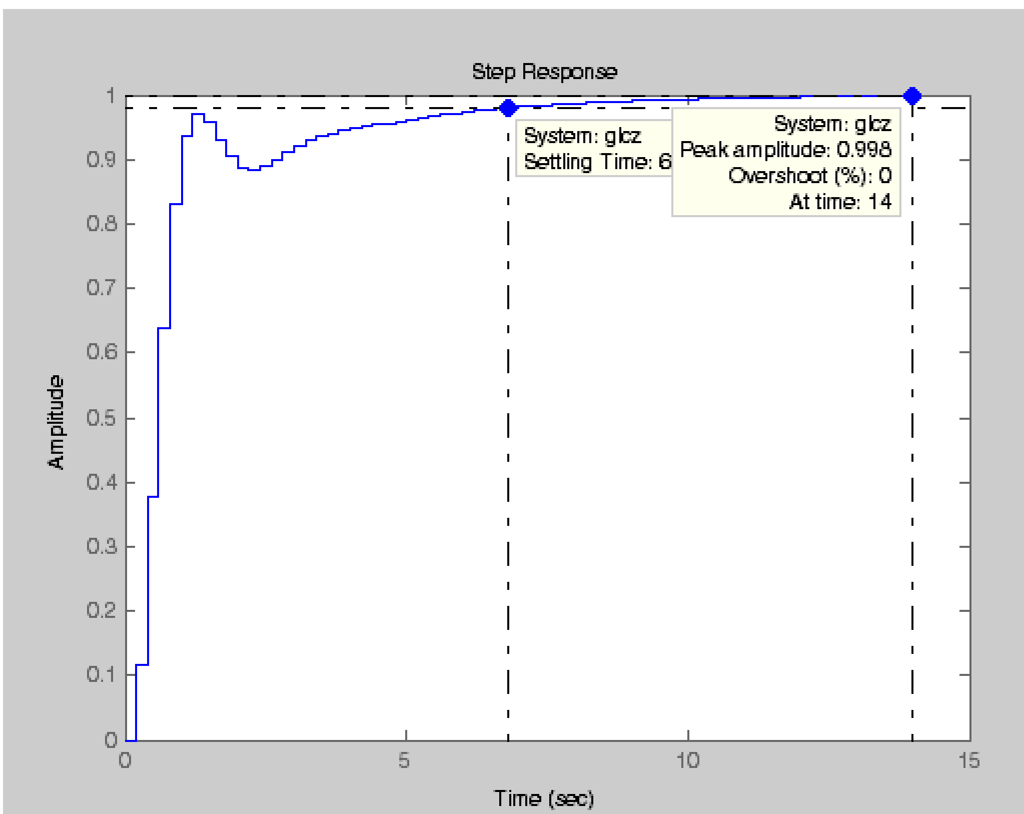


Bode Diagram
Gm = 14.274 dB (at 3.2165 rad/sec), Pm = 31.566 deg (at 1.2541 rad/sec)
Magnitude (dB)
40
20
0
-20
-40
-60
SISTEMA SIN COMPENSAR FASE=31.6º
Phase (deg)
-90
-135
-180
-225
-270
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/sec)



Bode Diagram
Gm = 23.068 dB (at 3.1854 rad/sec), Pm = 53.479 deg (at 0.61115 rad/sec)
Magnitude (dB)
40
20
0
-20
-40
-60
-80
SISTEMA COMENSADO EN ATRASO: FASE=53.5º
Phase (deg)
-90
-135
-180
-225
-270
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/sec)


Bode Diagram
Gm = 17.39 dB (at 6.7516 rad/sec), Pm = 81.529 deg (at 1.2787 rad/sec)
Magnitude (dB)
SISTEMA COMPENSADO ATRASO-ADELANTO: FASE = 81.5º
Phase (deg)
Frequency (rad/sec)

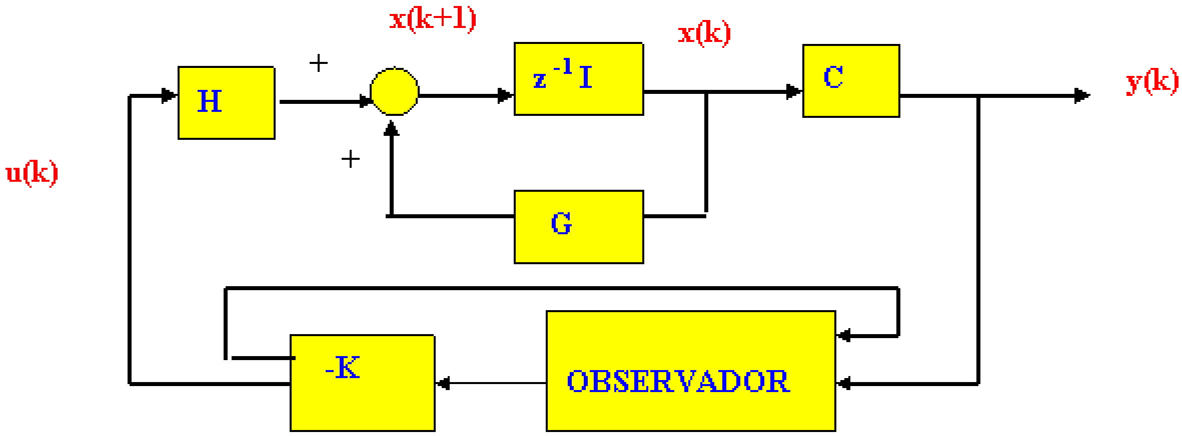

Step Response
System: glcz
Settling Time: 6
System: glcz
Peak amplitude: 0.998
Overshoot (%): 0
At time: 14
Amplitude
Time (sec)

10. DISEÑO: ECUACIONES DE ESTADO

10.1 REGULADOR DE ACKERMANN

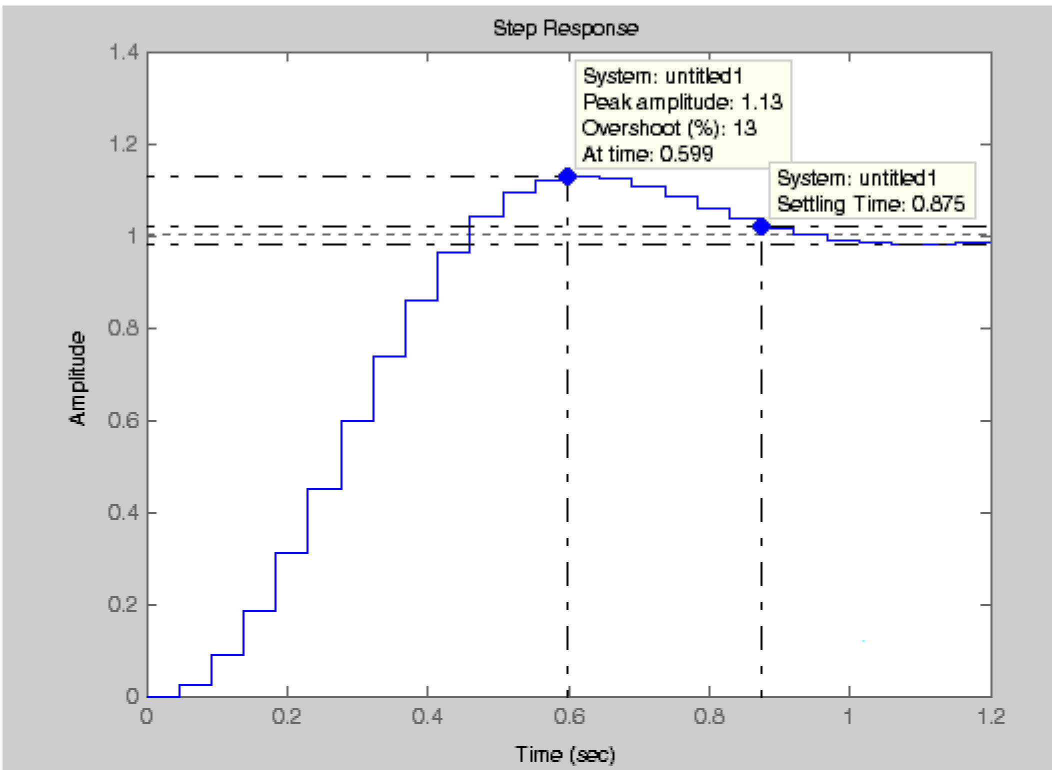
```
%*************************************************************************************
% PARA EL SISTEMA Gp(s)=36/(s(s+3.6)) DISEÑAR UN REGULADOR
DIGITAL
% APLICANDO FORMULA DE ACKERMANN DE TAL FORMA QUE :
% Mp=10% ; ts=0.8 Sg ; Ess=0
% Gc(z)=k(z-alfa)/(z-beta)
% SUPONER TIEMPO DE MUESTREO PARA OBTENER 20 MUESTRAS POR
CICLO.
% muestras/ciclo = ws/wd
%*************************************************************************************

clear all          % Borra todas las variables existentes
home

% Entre valores de Mp, ts y muestras/ciclo
Mp=0.1;
ts=0.8;
muestras=20;

% Calcular tiempo de muestreo
sigma=4/ts;        % tolerancia del 2%
% Mp = exp(-sigma*pi/wd)
wd=-sigma*pi/log(Mp);
ws=muestras*wd;
T=2*pi/ws;         % Tiempo de Muestreo
% T = 0.0461

% Calcular polo dominante del sistema en lazo cerrado
Z=exp(-sigma*T)*(cos(wd*T)+sin(wd*T)*i);
% Polo dominante : 0.7555 + 0.2455i
P1=real(Z)+imag(Z)*i;
P2=real(Z)-imag(Z)*i;
P=[P1 P2];
% P=[0.7555+0.2455i 0.7555-0.2455i]

% Funcion de la planta continua en ecuaciones de estado
Gps=zpk([ ],[0 -3.6],36);
Gpss=ss(Gps);
% .
% x = Ax + Bu
% y = Cx + Du
```

```
% A= [0 4; 0 -3.6], B= [0; 3]; C=[3  0], D=[0]

% Funcion de la planta discreta en ecuaciones de estado
Gpzss=c2d(Gpss,T,'zoh')
[G,H,C,D]=ssdata(Gpzss)        % Obtiene las matrices G,H,C,D
% x(k+1) = G x(k)+ H u(k)
% y(k) = C x(k)+ D u(k)
% G=[1  0.1697;  0 0.8472], H=[0.0120;  0.1273], C=[3  0], D=[0]

% Comprobar que la planta es controlable (observable)
% Es controlable (observable) si el rango de la matriz de
% controlabilidad(observabilidad) es igual al orden

orden=length(G);
co=ctrb(G,H);
rango=rank(co);
if rango==orden
disp('Es controlable')
else
disp('No es controlable')
end
ob=obsv(G,C);
rango=rank(ob);
if rango==orden
disp('Es observable')
else
disp('No es observable')
end
hold
% Obtencion de la matriz de realimentacion (K) y ganancia del observador (L)
:
% La formula de Ackermann calcula el vector ganancia K
% tal que la la realimentacion del estado u(k) = -K x(k)
% ubica los polos en lazo cerrado en las locaciones P
```




```
K=acker(G,H,P)              % Matriz de realimetación
L=acker(G',C',P)'           % Ganancia del observador
% K = [5.1193 2.1572]         % vector fila
% L = [0.1121; 0.1349]        % vector columna

% Obtencion del regulador (controlador) :
% El regulador es obtenido conectando la ley de
% realimentacion u(k)=-K x(k) y el observador de estado con
% la matriz de ganancia L
Gcss=reg(Gpzss,K,L);     % controlador en ecuaciones de estado
% A=[0.60199 0.14375; -1.0563 0.57259]
% B=[0.11211; 0.13486]
% C=[-5.1193 -2.1572]
% D= 0
Gcz=zpk(Gcss);
%              - 0.86483 (z-0.7631)
%    Gc(z) = ----------------------------
%              (z^2 - 1.175z + 0.4965)

% Funcion de transferencia en lazo cerrado
Glcss=feedback(Gpzss,-Gcss);
Glcz=tf(Glcss);

% Obtencion de la ganancia Ko :
% Ko se ajusta para que la respuesta al paso en estado
% estacionario sea 1, estoes y(inf)=1 , Ess=0
syms z  Ko
[num,den]=tfdata(Glcz,'v');
Glczsym=poly2sym(num,'z')/poly2sym(den,'z');
% Funcion de transferencia Fz=Yz/Rz
% Rz= 1/(1-z^-1) Transf_Z del escalon unitario
```

```
Fz=Ko*Glczsym;
% Error de estado estacionario
% yInf = limit((1-z^-1)Yz,z,1)=limit(Fz,z,1)
yInf=1;
y= limit(Fz,z,1)-yInf;
Ko=eval(solve(y));
% Ko = 0.6363

% Respuesta al paso
step(Ko*Glcss)
```

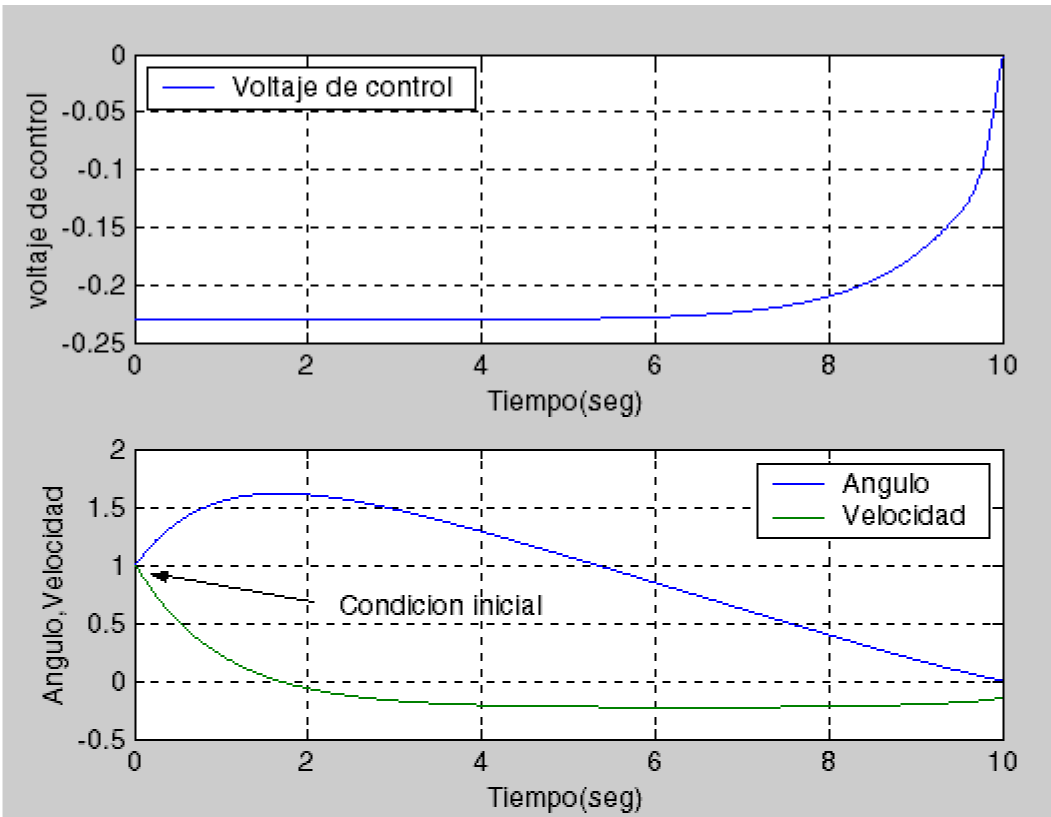


15.2 FUNCIÓN DE COSTO ÓPTIMO

```
% El desempeño de un sistema de control puede ser cuantificado  por una funcion de
% costo. El controlador que minimixa el costo se conoce como controlador optimo. Todos
% los estados deben ser medidos. Un SERVOMOTOR est´a descrito por las variables de
% estado % x1(t):posicion angular, x2(t): velocidad angular (medibles) entrada de control
```

```
% u(t):voltaje aplicado (voltaje maximo =5V) El modelo de la planta esta
definido por sus
% matrices:

clear all
home
A=[0 1;0 -1];
B=[0;1];
C=[1 0];
% La funcion de costo cuadratica esta definida por :
% J=(1/2)*x'(t)*H*x(t)+(1/2)*Integral(x'(t)*Q*x(t)+R*u^2(t))
% H: peso de la posicion final
% Q: peso de la posicion antes del tiempo final
% R: peso de la entrada de control
H=[10 0; 0 0];
Q=[0 0; 0 0];
R=0.1;
% tf es tiempo final=10. Si se hacen 1000 iteraciones, entonces dt=T=0.01
tf=10;
T=0.01;
t=0.01:0.01:tf;
% Inicializar solucion de Ricatti, ganancia optima y estado inicial
P=H;
n=1000;
K(n,:)=[0 0];
x(:,1)=[1;1];

% Realizar las iteraciones para ganancia de realimentacion
% y ley de control optimo
for i=n-1:-1:1
P=P-T*(-P*A-A'*P-Q+P*B*inv(R)*B'*P);
K(i,:)=inv(R)*B'*P;
end
for i=2:n,
u(i-1)=-K(i-1,:)*x(:,i-1);
x(:,i)=x(:,i-1)+T*(A*x(:,i-1)+B*u(i-1));
end
% Control final
u(n)=-K(n,:)*x(:,n);
% K(n,:)=[0 0], x(:,n)=[0.0038 -0.1456], u(n)=0

% Grafica de voltaje de control
subplot(211)
plot(t,u)
xlabel('Tiempo(seg)')
ylabel('voltaje de control')
legend('Voltaje de control',2);
```

```
grid
% Grafica de angulo y velocidad de rotacion
subplot(212)
plot(t,x(1,:),t,x(2,:))
xlabel('Tiempo(seg)')
ylabel('Angulo,Velocidad')
legend('Angulo','Velocidad')
grid

% Calcular costo optimo
% costo=P*x^2
Costo = x(:,1)'*P*x(:,1)
% costo = 0.0461
```



10.3 LQR: ESTADO TRANSITORIO

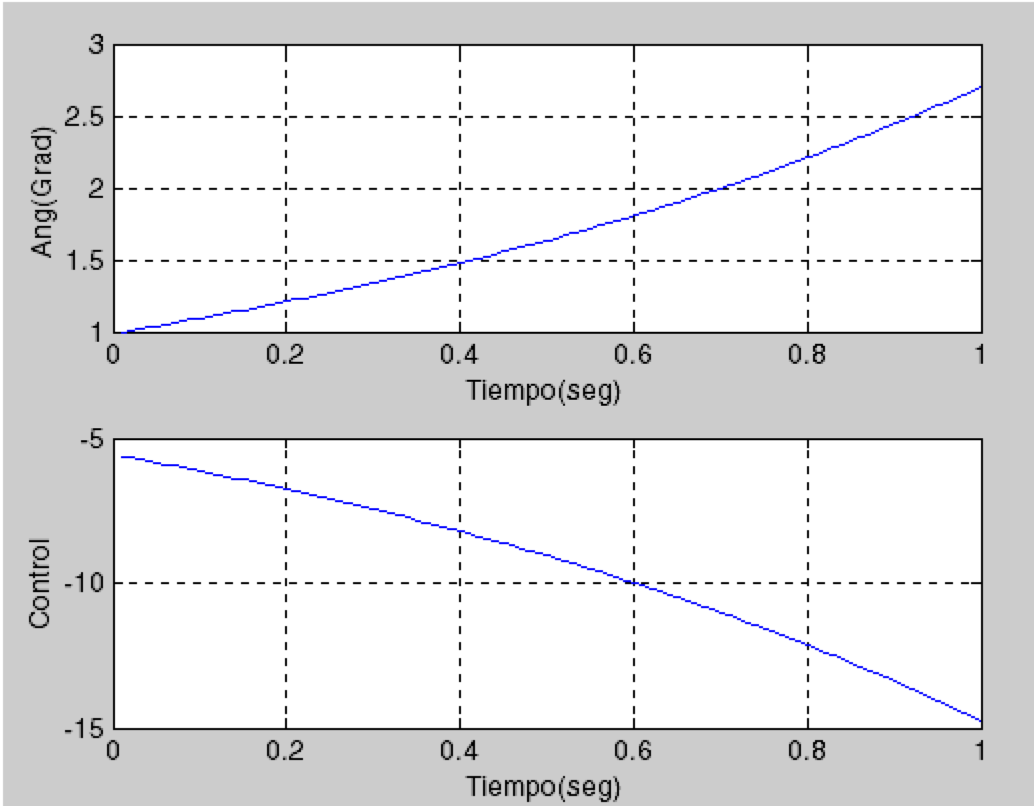
```
%**************************************************************************
% REGULADOR CUADRATICO LINEAL: ESTADO TRANSITORIO
% J : Indice de desempeño cuadratico,
```

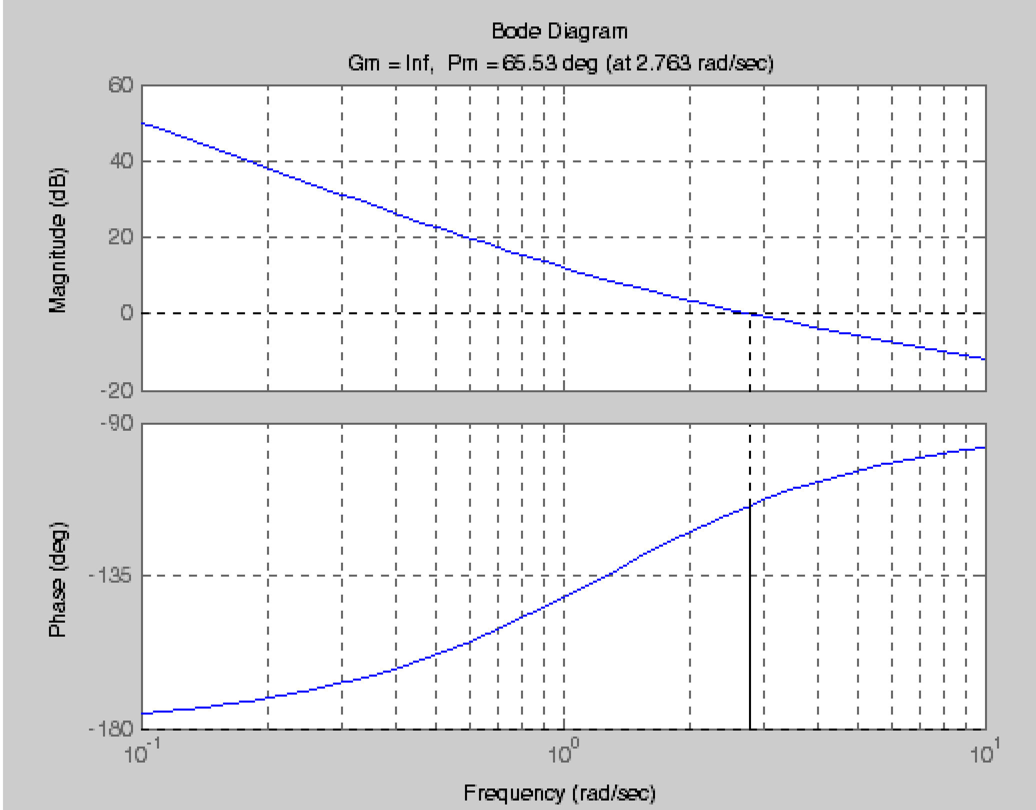
```
% x(n ): estado final, x(0) : condicion inicial
% J = (1/2)x'(n)*S*x(n)+(1/2)Sum(x'(k)*Q*x(k)+u'(k)*R*u(k)), para k:0 a n-1
% Para un sisitema de control lineal
% x(k+1)=G*x(k)+H*u(k), x(0)= c
%*************************************************************************
clear all
home
G=0.3679;
H=0.6321;
x0=1;
% Para obtener el vector de control optimo u(k) en lazo
% cerrado se debe solucionar la Ecuacion de Riccati
% P(k)=Q+G'*P(k+1)*G-G'*P(k+1)*H*inv(R+H'*P(k+1)*H*)*H'*P(k+1)*G
% Para el ejemplo, encontrar u(k) para minimizar J
% J= (1/2)*(x(10)^2)+(1/2)*Sum((x(k)^2)+(u(k)^2)), k= 0 a 9, entonces,
S=1;
Q=1;
R=1;
% Ecuacion de riccati. Para valor minimo de Jmin: P(n)=S
n=10;
Pk1=S;

% Valores de la ganancia de realimentacion K(k)
K(n,:)=0;
for i=n-1:-1:1
Pk=Q+G'*Pk1*G-G'*Pk1*H*inv(R+H'*Pk1*H)*H'*Pk1*G;
Pk1=Pk;
K(i,:)=inv(R)*H'*inv(G')*(Pk-Q);
end
% K(10)=0,K(9)=0.1662,K(8)=0.1773,K(7)=0.1781,......,K(0)=0.1781
% Valor en estado estacionario de K(7) = 0.1781

% Valores del control optimo
% u(k)=-K(k)*x(k); x(k+1)=(G-H*K(k))*x(k)
x(:,1)=1
for i=2:n,
u(i-1)=-K(i-1,:)*x(:,i-1);
x(:,i)=(G-H*K(i-1,:))*x(:,i-1)
end
% u(0)=-0.1781,u(1)=-0.0455,u(2)=-0.0116, u(3)=-0.0030
% u(4)=-0.0008,u(5)=-0.0002,u(6)=0,......,u(10)=0
% Valor en estado estacionario de u(0) = 0
```

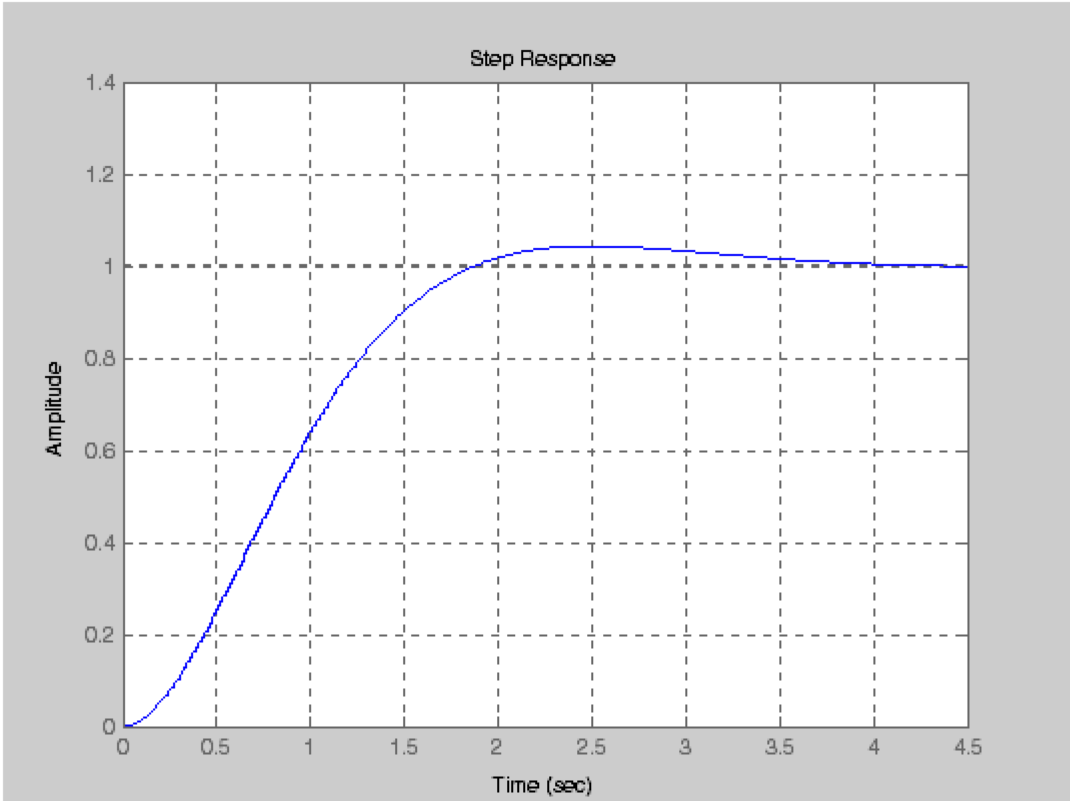
10.4 LQR: ESTADO ESTACIONARIO

```
%***********************************************************************
% REGULADOR CUADRATICO LINEAL: ESTADO TRANSITORIO
% En estado estacionario (n=inf) el indice de desmpeño J es :
% J=(1/2)*sumatoria(x'(k)*Q*x(k)+u'(k)*R*u(k)) k=0 a inf
%***********************************************************************
clear all
home

% Modelo de la planta
A=[0 1;0 0];
B=[0; 1];
C=[1 0];
D=0;
Gp=ss(A,B,C,D)

% Parametros de la funcion de costo
Q=[1 0; 0 0];
R=0.1;

% Planta discreta
% x(k+1)=G*x(k)+H*u(k)
T=0.01;
Gpz=c2d(Gp,T);
[G,H,C,D]=ssdata(Gpz);
% Resultados aplicando dlqr
% Para el estado estacionario
% K,P,E]=dlqr(G,H,Q,R)
% K=Matriz de ganancia en estado estacionario
% P= Solucion de la ecuacion de Riccati
% E= valores caracteristicos del sistema en lazo cerrado

[Kss,Pss,Ess]=dlqr(G,H,Q,R)

% Kss =[0.9874 + 0.0124i; 0.9874 - 0.0124i]
% Pss =[ 80.0286 31.6228; 31.6228 25.1492]
% Ess =[0.0908 0.3440]

% Tambien se puede calcular Ess=eig(G-H*Kss)

% Calcular la respuesta del sistema en lazo cerrado y el costo
T = 0.01;
n=100;
```

```
% Inicializar el estado y el costo
xss(:,1)=[1;1];
Costss=T*xss(:,1)'*Q*xss(:,1);
% Costss(inicial) = 0.0100

for i=2:n
uss(i-1)=-Kss*xss(:,i-1);
xss(:,i)=xss(:,i-1)+T*(G*xss(:,i-1)+H*uss(i-1));
Costss=Costss+T*(xss(:,i)'*Q*xss(:,i)+uss(i-1)'*R*uss(i-1));
end
uss(n)=-Kss*xss(:,n);
Costss=Costss*uss(n)'*R*uss(n);
% Costss = 275.7941

% Graficar resultados
figure(1)
t=T:T:n*T;
subplot(211)
plot(t,xss(1,:))
xlabel('Tiempo(seg)');
ylabel('Ang(Grad)');
grid
subplot(212)
plot(t,uss')
xlabel('Tiempo(seg)');
ylabel('Control');
grid

% Calcular margen de fase y ganancia sistema lazo abierto continuo
figure(2)
K=lqr(A,B,Q,R);
margin(A,B,K,0);
grid
[Gm,Pm]=margin(A,B,K,0)

% Respuesta al paso sistema en lazo cerrado discreto
figure(3)
Glc=ss(G-H*Kss,H*Kss,C,D,T);
step(Glc(:,1))
grid
```


Ang(Grad)
Tiempo(seg)
Control
Tiempo(seg)


Bode Diagram
Gm = Inf, Pm = 65.53 deg (at 2.763 rad/sec)
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)


Step Response
Amplitude
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.5
1.5
2
2.5
3
3.5
4
4.5

BIBLIOGRAFÍA

[1] B.C. KUO. Sistemas de Control Automático. Prentice Hall. 1996

[2] B.C. KUO. Sistemas de Control Digital. Cecsa. 1997

[3] K. OGATA. Ingeniería de Control Moderna. Prentice Hall. 1998

[4] K. OGATA. Sistemas de Control en el Tiempo Discreto. Prentice Hall. 1998

[5] K. OGATA. Problemas de Ingeniería de Control utilizando Matlab.Prentice Hall. 1999

[6] K. OGATA. Ingeniería de Control Moderna. Prentice Hall. 1998

[7] MathWorks. The Student Edition of Matlab. Prentice Hall. 1992

[8] MathWorks. Matlab User's Guide. Prentice Hall. 1992

[9] K. OGATA. Dinámica de Sistemas.. Prentice Hall. 1987

[10] C. PÉREZ. Matlab y sus Aplicaciones en las Ciencias y la Ingeniería. Prentice Hall. 2002

[11] CASTRO, SOLÉ, ALCALÁ, MORENO. Teoría de control. Diseño Electrónico. Alfaomega. 1999

[12] B. DORF. Sistemas de Control Moderno. Addison-Wisley.1999

[13] J.R. COGDELL. Fundamentos de Máquina Eléctricas. Prentice Hall. 2002

[14] A. SMITH. Curso Básico de Motores Eléctricos. Glem. 1980

[15] G. ENRIQUEZ. ABC de las instalaciones Eléctricas. Limusa. 1985

[16] J. F. McPARTLAND. Cómo diseñar sistemas eléctricos. Diana. 1980

[17] DELMAR. Manual de Electricidad . Tomo 1. Diana. 1971

[18] DELMAR. Manual de Electricidad . Tomo 2. Diana. 1971

[19] DELMAR. Manual de Electricidad . Tomo 3. Diana. 1971

[20] CHE-MUN ONG. Dynamic Simulation of Electric Machinary Using Matlab/Simulink. Prentice Hall. 1998

[21] Jeffrey B. Burl. Linear Optimal Control. Addison - Wesley. 1999

[22] N. S. NICE. Control System Engineering. addison - Wesley. 1995

[23] H. KWAKERNNAK & SIVAN. Linear Optimal Control System. 1994

[24] ANDERSON & MOORE. Optimal Control. Linear Quadratic Methods. Prentice Hall. 1990

[25] B. FRIEDLAND. Control System Design. An Introduction to State-Space Methods. Prentice Hall. 1986

[26] GRACE, LAUB, LITTLE and THOMSON. Control Systems Toolbox For Use with Matlab. MathWorks. 1990