Implementación de reguladores digitales

CONTROL AUTOMÁTICO

ESCUELA DE ELECTRÓNICA

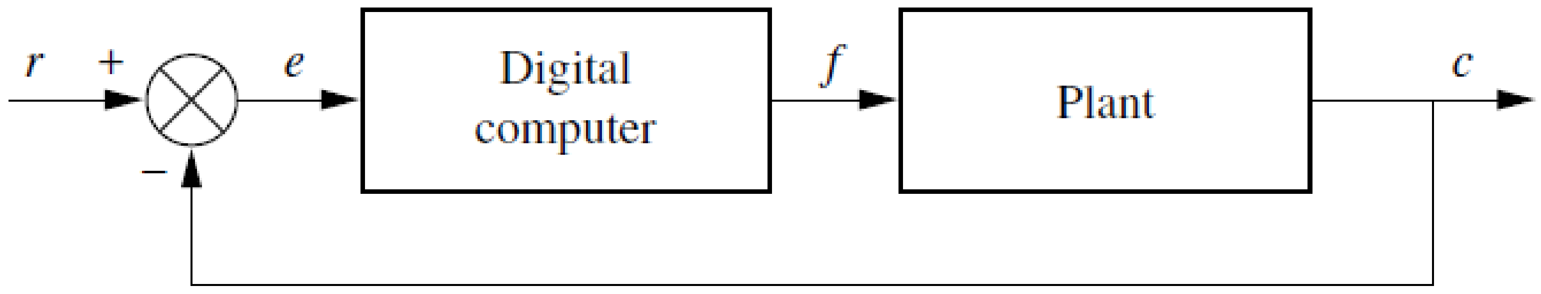
II SEMESTRE 2020

ING. LUIS MIGUEL ESQUIVEL SANCHO

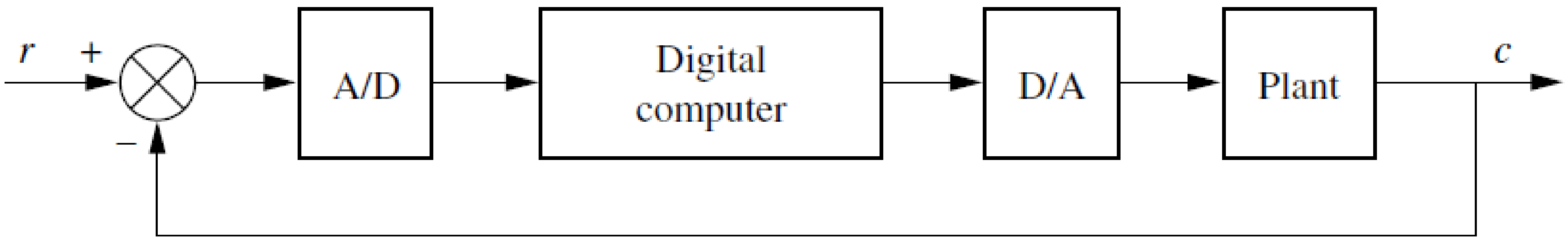
Contenido

- Esquema del control digital
- Discretización a partir de la FT en tiempo continuo
  - Por respuesta invariante al impulso
  - Por retenedor de orden cero (ZOH)
  - Por Tustin o transformación bilineal
  - Por mapeo de polos (solo para SISO)
- Implementación Digital

Esquema del control digital

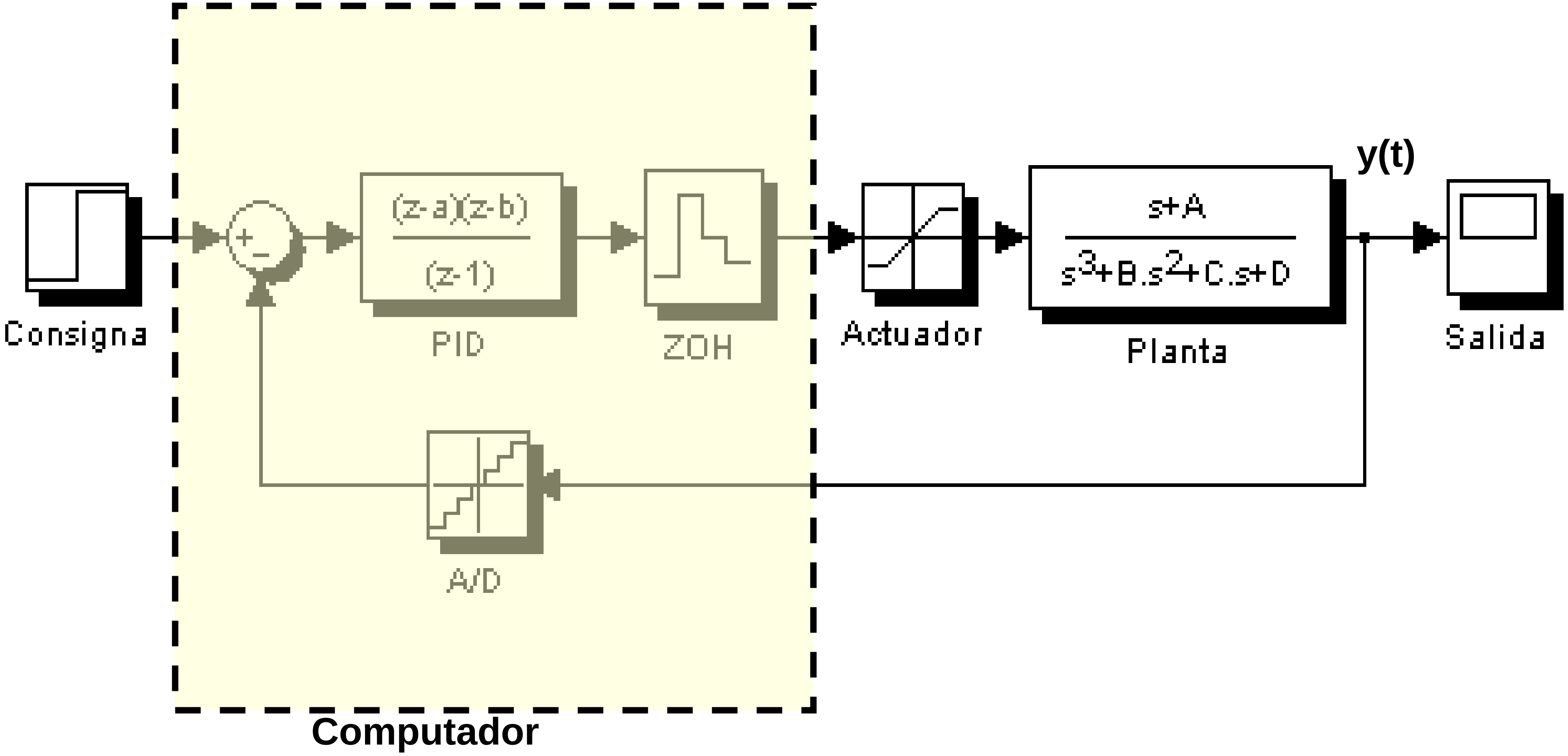


(a)



(b)

Esquema del control digital



Escogencia del periodo de muestreo

El periodo de muestreo T_s debe ser escogido para que sea menor que una décima parte de la constante de tiempo dominante, del sistema que se espera en **lazo cerrado**

$$T_s < \frac{1}{10} T_{dom.}$$

- Recomendaciones para escoger el periodo de muestreo T o frecuencia de muestreo f_T.
  - En lazo cerrado:

$$\frac{t_r}{20} < T < \frac{t_r}{10}; \text{ tiempo de subida}$$

$$\frac{t_s}{75} < T < \frac{t_s}{25}, \text{ tiempo de estabilización}$$

$$20BW < f_T < 40BW, \text{ ancho de banda } [Hz]$$

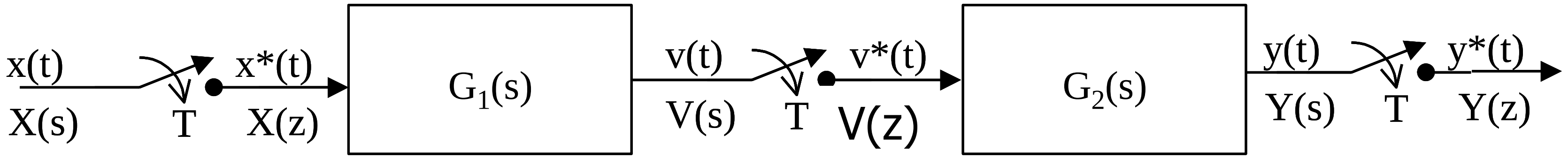
$$30\omega_d < \omega_T < 60\omega_d, \text{ frecuencia amortiguada } [rad/s]$$

  - En lazo abierto:

$$40\omega_{cg} < \omega_T < 80\omega_{cg}, \text{ frecuencia de cruce de g. } [rad/s]$$

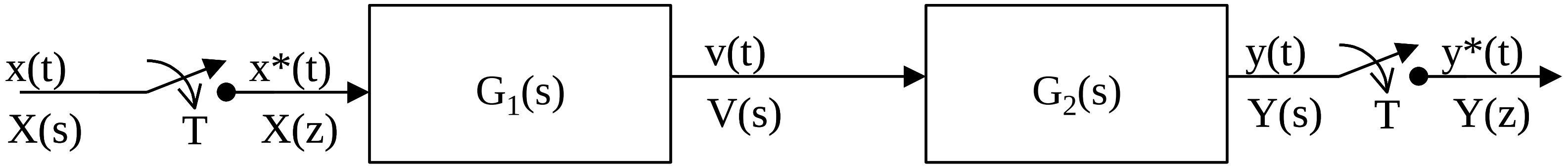
Condiciones de la conversión

Dos funciones continuas son discretizadas y se encuentran en cascada



$$G(z)=\frac{Y(z)}{X(z)}=G_1(z)\cdot G_2(z)=Z\{G_1(s)\}\cdot Z\{G_2(s)\}$$

Dos funciones continuas en cascada son discretizadas; $G(s) = G_1(s)G_2(s)$

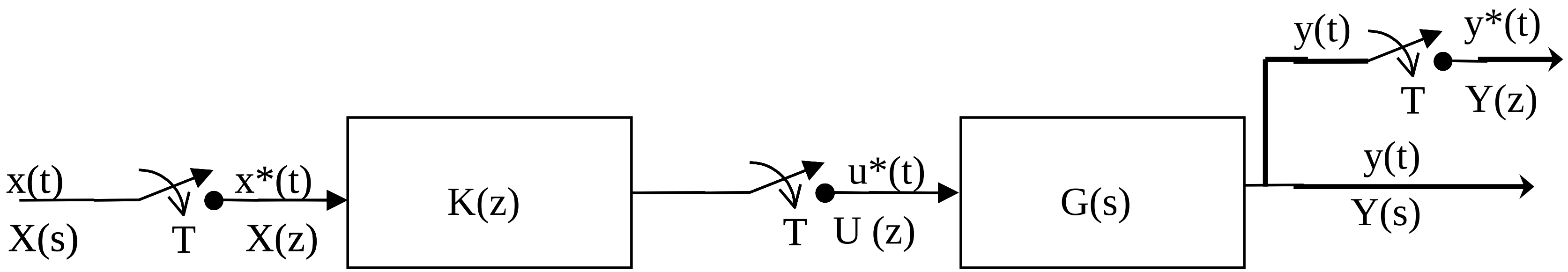


$$G(z)=\frac{Y(z)}{X(z)}=Z\{G(s)\}=Z\{G_1(s)G_2(s)\}$$

Condiciones de la conversión

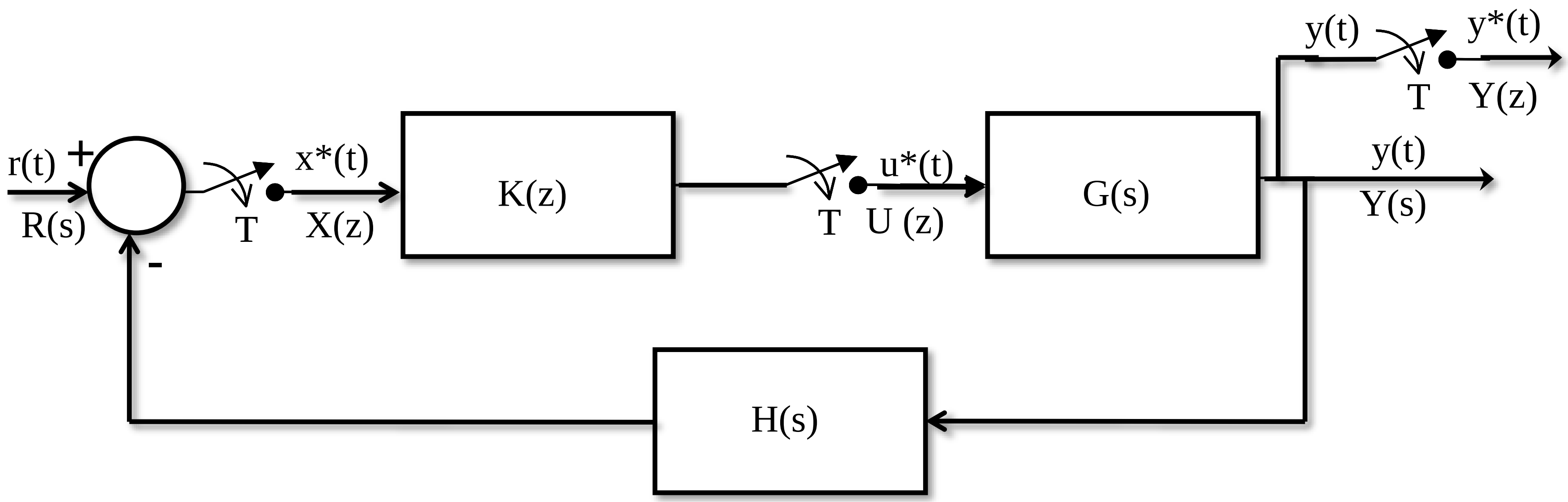
Procedimiento:

Combinar todos los elementos continuos que se encuentren directamente conectados en cascada en una única función G(s)



Considerar como si la función G(s), tiene además, paralelamente a su salida continua, una salida muestreada. G(z) = Y(z)/U(z)

El sistema híbrido de control



$$T(z)=\frac{Y(z)}{R(z)}=\frac{K(z)G(z)}{1+K(z)GH(z)}=\frac{K(z)Z\{G(s)\}}{1+K(z)Z\{G(s)H(s)\}}$$

Transformada Z

	$f(t)$	$F(s)$	$F(z)$	$f(kT)$
1.	$u(\mathrm{t})$	$\frac{1}{s}$	$\frac{z}{z-1}$	$u(kT)$
2.	t	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$	kT
3.	t^n	$\frac{n!}{s^{n+1}}$	$\lim_{a\to 0}(-1)^n \frac{d^n}{da^n}\left[\frac{z}{z-e^{-aT}}\right]$	$(kT)^n$
4.	e^{-at}	$\frac{1}{s+a}$	$\frac{z}{z-e^{-aT}}$	e^{-akT}
5.	$t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$	$(-1)^n \frac{d^n}{da^n}\left[\frac{z}{z-e^{-aT}}\right]$	$(kT)^n e^{-akT}$
6.	$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$	$\frac{z \sin \omega T}{z^2 - 2z\cos \omega T + 1}$	$\sin \omega kT$
7.	$\cos \omega t$	$\frac{s}{s^2+\omega^2}$	$\frac{z(z-\cos \omega T)}{z^2 - 2z\cos \omega T + 1}$	$\cos \omega kT$
8.	$e^{-at}\sin \omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$	$\frac{ze^{-aT}\sin \omega T}{z^2 - 2ze^{-aT}\cos \omega T + e^{-2aT}}$	$e^{-akT}\sin \omega kT$
9.	$e^{-at}\cos \omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$	$\frac{z^2 - ze^{-aT}\cos \omega T}{z^2 - 2ze^{-aT}\cos \omega T + e^{-2aT}}$	$e^{-akT}\cos \omega kT$

Transformada Z (Teoremas)

	Theorem	Name
1.	$z\{af(t)\} = aF(z)$	Linearity theorem
2.	$z\{f_1(t) + f_2(t)\} = F_1(z) + F_2(z)$	Linearity theorem
3.	$z\{e^{-aT}f(t)\} = F(e^{aT}z)$	Complex differentiation
4.	$z\{f(t - nT)\} = z^{-n}F(z)$	Real translation
5.	$z\{tf(t)\} = -Tz\dfrac{dF(z)}{dz}$	Complex differentiation
6.	$f(0) = \lim_{z\to\infty} F(z)$	Initial value theorem
7.	$f(\infty) = \lim_{z\to 1}(1 - z^{-1})F(z)$	Final value theorem

Transformación por respuesta invariante al impulso

- Combinar todos los elementos continuos que se encuentren directamente conectados en cascada en una única función G(s)
- Representar G(s) en fracciones parciales
- Convertir cada fracción parcial a su forma en Z usando la transformada Z de la función exponencial muestreada y con ayuda de tablas.

$$L\{e^{-at}\} = \frac{1}{(s+a)}$$

$$Z\{e^{-at}\} = Z\{e^{-akT}\} = \frac{1}{1-e^{-aT}z^{-1}}$$

Tiene problemas de *aliasing* dependiente de T

Transformación por respuesta invariante al impulso

Para raíces simples

$$G(z) = T\sum_{i=1}^{n}\frac{R_i}{(1-e^{p_iT}z^{-1})}$$

Donde: R_i = residuo del polo i-ésimo; p_i = polo i-ésimo;

T = periodo de muestreo

Para raíces repetidas

$$G(z) = T\sum_{i=1}^{j}\sum_{l=1}^{m_i}\frac{(-1)^{l-1}R_{il}}{(l-1)!}\frac{\partial^{l-1}}{\partial p_i^{\,l-1}}\left[\frac{1}{(1-e^{p_iT}z^{-1})}\right]$$

Donde: R_{il} = residuo de la repetición l del polo i-ésimo; p_i = polo i-ésimo; T = periodo de muestreo

Ejemplo 1: Discretización por respuesta invariante al impulso

Encuentre el modelo en tiempo discreto para la planta P(s) mostrada, con T = 0.1s

$$P(s) = \frac{4}{(s+1)(s+4)}$$

Evaluando los residuos para los polos

P = [-4 -1]

R = [-4/3 4/3]

Ejemplo 1: Discretización por respuesta invariante al impulso

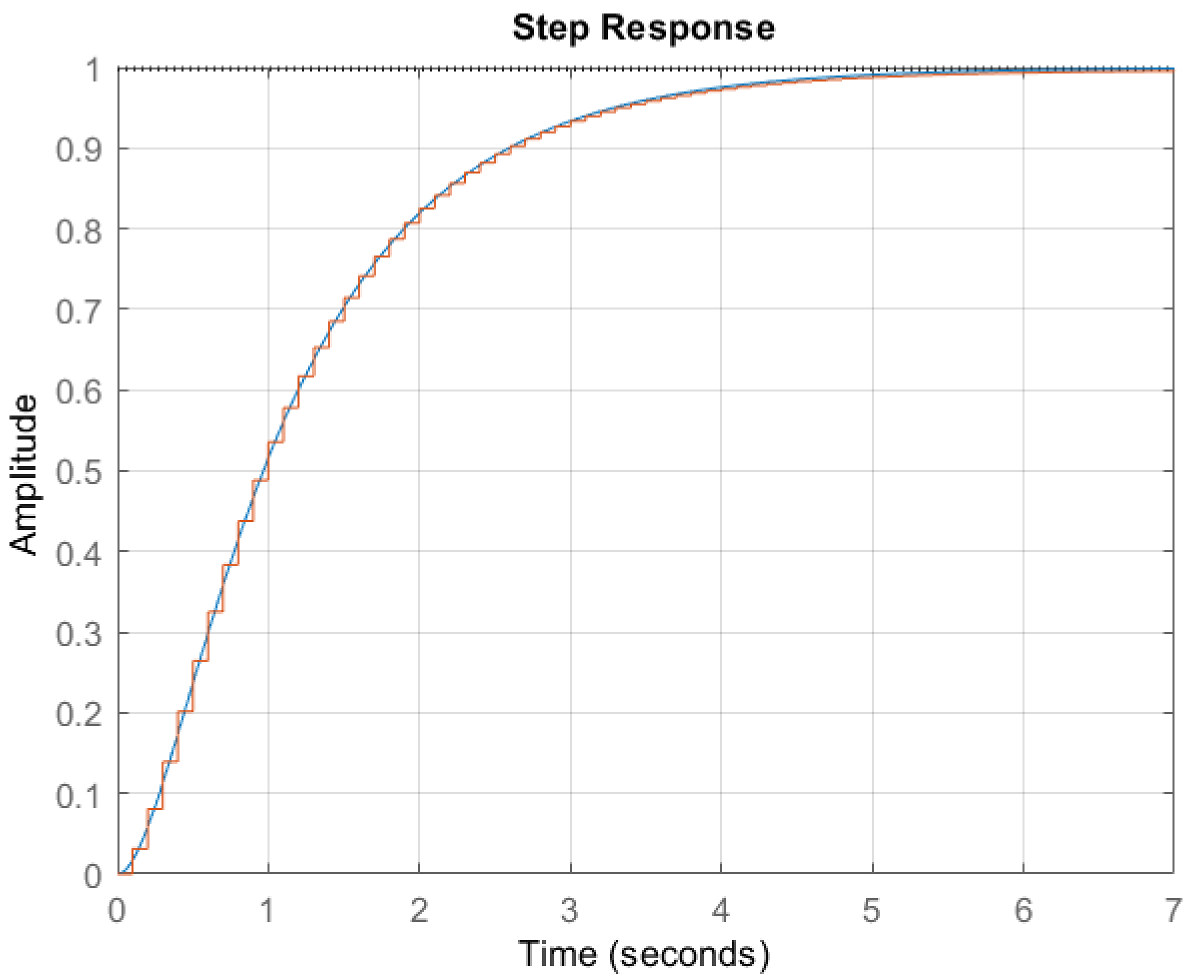
Evaluando T= 0.1s

$$P(z) = 0.1 * \sum_{i=1}^{2} \frac{R_i}{(1 - e^{p_i T} z^{-1})} = \frac{-4/3}{(1 - e^{-4*0.1} z^{-1})} + \frac{4/3}{(1 - e^{-1*0.1} z^{-1})}$$

$$P(z) = 0.1 * \left[\frac{-4/3 * z}{(z - 0.6703)} + \frac{4/3 * z}{(z - 0.9048)} \right]$$

$$P(z) = \frac{0.031269z}{(z - 0.6703)(z - 0.9048)}$$

Ejemplo 1: Discretización por respuesta invariante al impulso



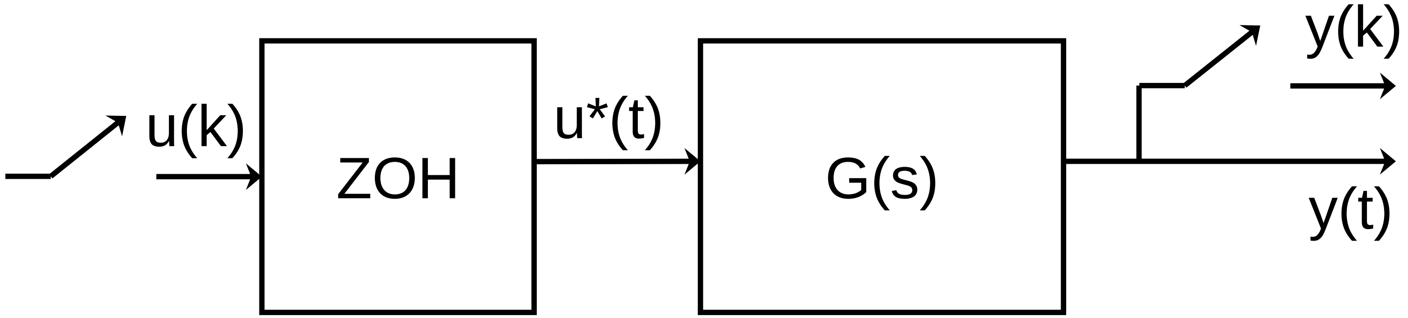

```
sysdi =

       0.031269 z
  ---------------------
  (z-0.9048) (z-0.6703)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

Código Matlab:

```
clc
clear all

z = []; %ceros de P(s)
p = [-1 -4]; %polos de P(s)
k = 4; %Constante de P(s)
T = 0.1; %Tiempo de muestreo
sys = zpk(z,p,k); %F.T. definida en ceros, polos y k
sysdi = c2d(sys,T,'impulse'); %F.T. discretizado por el método
                              %Respuesta invariante al impulso
step(sys,sysdi)  %Respuesta al escalón unitario.
grid
```

Retenedor de orden cero (ZOH)

Se combina la función de transferencia del retenedor de orden cero con la de la planta.



$$Z\left\{\frac{1-e^{-sT}}{s} * G(s)\right\} = (1-z^{-1}) * Z\left\{\frac{G(s)}{s}\right\}$$

Método preferible para sistemas con tiempo muerto

El resultado se descompone en fracciones parciales y cada fracción se transforma a Z.

Ejemplo 2: Discretización de una planta por ZOH

Encuentre el modelo en tiempo discreto para la planta P(s) mostrada, con T = 0.1s

$$P(s) = \frac{4}{(s+1)(s+4)}$$

Ejemplo 2: Discretización de una planta por ZOH

Descomponiendo en fracciones parciales P(s)/s los residuos y polos son:

R = [0.3333 -1.3333 1.0000]

P = [-4 -1 0]

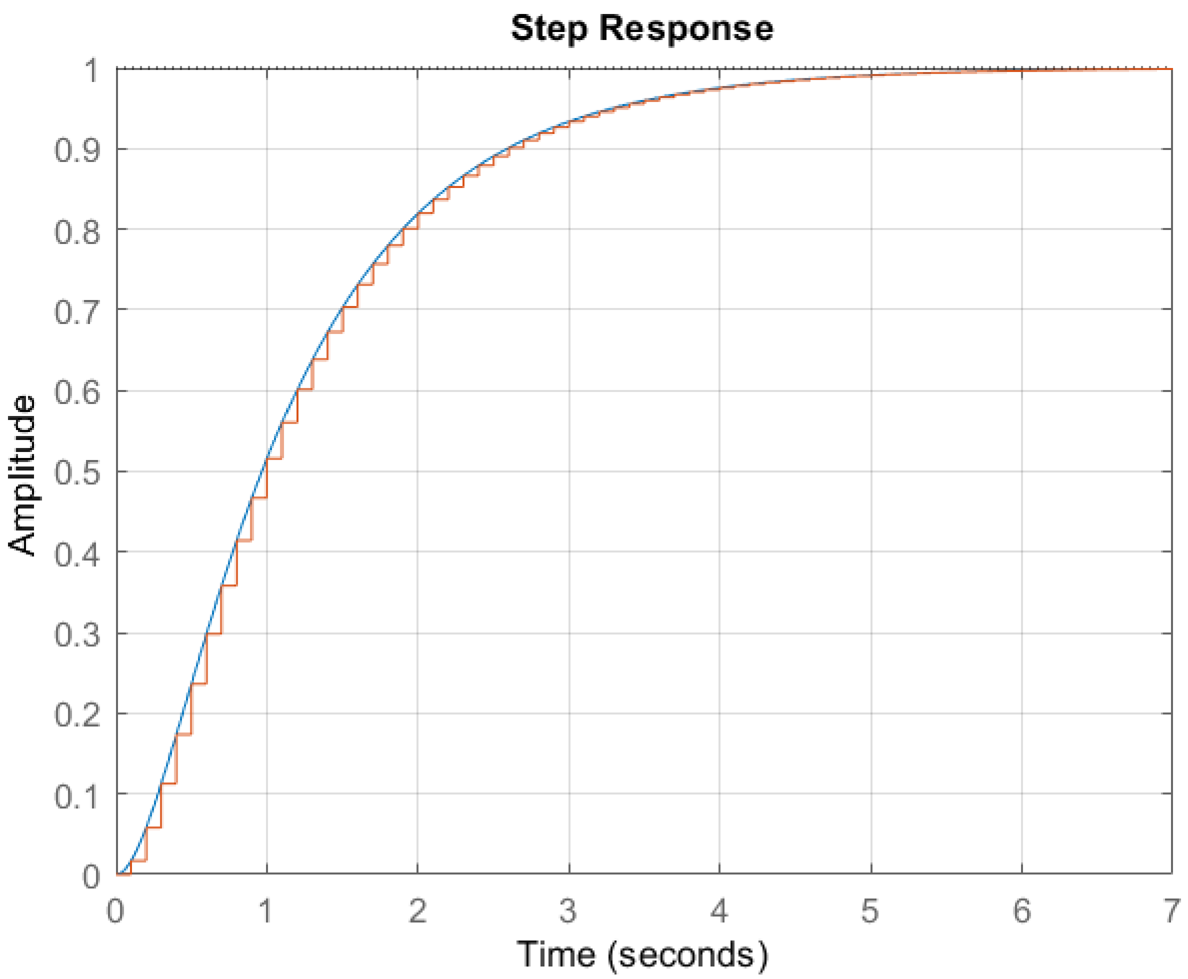
Después de transformar cada fracción parcial aplicando el periodo de muestreo T = 0.1s

$$P(z) = (1 - z^{-1})\left[\frac{0.33}{(1 - e^{-(4*0.1)}z^{-1})} - \frac{1.33}{(1 - e^{-(1*0.1)}z^{-1})} + \frac{1}{(1 - e^{0}z^{-1})}\right]$$

Sumamos, aplicamos el factor (1 - z^{1}), factorizamos y simplificamos:

$$P(z) = \frac{0.01699\ (z+0.8466)}{(z\text{-}0.9048)(z\text{-}0.6703)}, T = 0.1s$$

Ejemplo 2: Discretización de una planta por ZOH




```
syszoh =

    0.01699 (z+0.8466)
  ----------------------
  (z-0.9048) (z-0.6703)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

Código Matlab:

```
clc
clear all

z = []; %ceros de P(s)
p = [-1 -4]; %polos de P(s)
k = 4; %Constante de P(s)
T = 0.1; %Tiempo de muestreo
sys = zpk(z,p,k); %F.T. definida en ceros, polos y k
syszoh= c2d(sys,T,'ZOH') %F.T. discretizado por el método
                              %Retenedor de orden cero (ZOH)
step(sys,syszoh)  %Respuesta al escalón unitario.
grid
```

Método de Tustin o transformación bilineal

Partimos de la relación de definición de z $z = e^{sT}$

Despejamos *s* y desarrollamos la serie de potencias del logaritmo

No tiene problemas de *aliasing* dependiente de T

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

$$s = \frac{2}{T}\frac{(z-1)}{(z+1)} + \frac{2}{3T}\frac{(z-1)^3}{(z+1)^3} + \frac{2}{5T}\frac{(z-1)^5}{(z+1)^5} + \cdots$$

Finalmente aproximamos al primer término: $s \approx \frac{2}{T}\frac{(z-1)}{(z+1)}$

Ejemplo 3: Discretización de una planta por Tustin

Encuentre el modelo en tiempo discreto para la planta P(s) mostrada, con T = 0.1s

$$P(s) = \frac{4}{(s+1)(s+4)}$$

Ejemplo 3: Discretización de una planta por Tustin

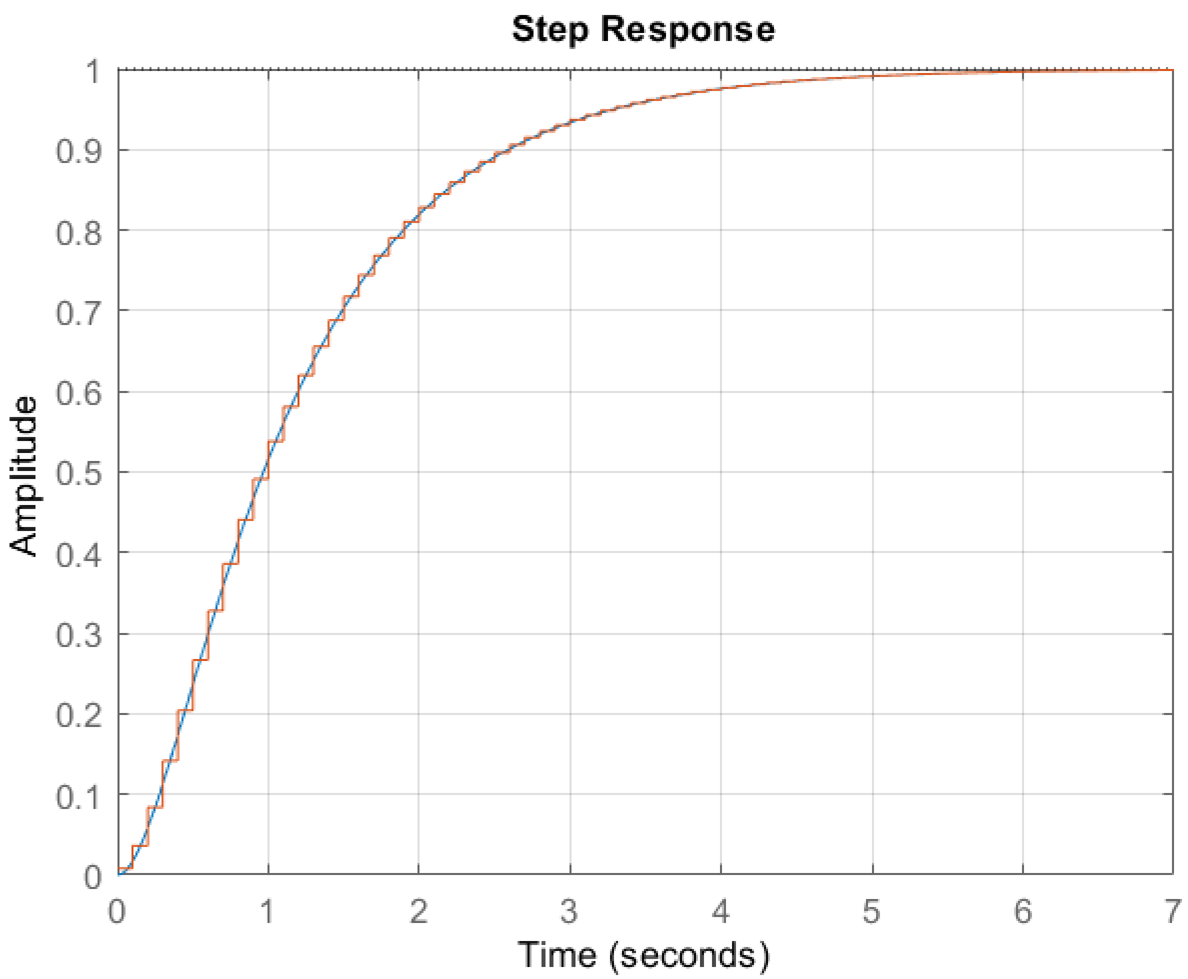
$$P(s) = \frac{4}{(s+1)(s+4)}$$

Sustituyendo

$$P(z) = \frac{4}{\left(\frac{2}{0.1}\frac{(z-1)}{(z+1)}+1\right)\left(\frac{2}{0.1}\frac{(z-1)}{(z+1)}+4\right)}$$

$$P(z) = \frac{0.0079365(z+1)^2}{(z-0.6667)(z-0.9048)}$$

Ejemplo 3: Discretización de una planta por Tustin




```
sysdT =

    0.0079365 (z+1)^2
  ----------------------
  (z-0.9048) (z-0.6667)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

Código Matlab:

```
clc
clear all

z = []; %ceros de P(s)
p = [-1 -4]; %polos de P(s)
k = 4; %Constante de P(s)
T = 0.1; %Tiempo de muestreo
sys = zpk(z,p,k); %F.T. definida en ceros, polos y k
sysdT= c2d(sys,T,'Tustin') %F.T. discretizado por el metodo
                              %Tustin (trans. bilineal)
step(sys,sysdT)  %Respuesta al escalon unitario.
grid
```

Mapeo de polos y ceros

Solo para sistemas SISO. Se sustituye cada polo de la forma $(s - pi)$ por $(z - e^{piT})$ y cada cero de la forma $(s - z_i)$ por $(z - e^{ziT})$. Finalmente se ajusta la ganancia K’ para una respuesta igual, típicamente a frecuencia cero.

$$G(s)\big|_{S=0} = G(z)\big|_{Z=1}$$

n > q

$$G(s) = K \frac{\prod_{i=1}^{q}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)} \qquad G(z) = K^1 (z+1)^{(n-q-1)} \frac{\prod_{i=1}^{q}(z - e^{z_i T})}{\prod_{i=1}^{n}(z - e^{p_i T})}$$

Mapeo de polos y ceros

La ganancia K' se calcula como

$$K^1 = K_B \cdot \left. \frac{\prod_{i=1}^{n}(z - e^{p_i T})}{(z+1)^{(n-q-1)}\prod_{i=1}^{q}(z - e^{z_i T})} \right|_{Z=1}$$

$$K_B = G(s)\Big|_{S=0} = K\,\frac{\prod_{i=1}^{q} - z_i}{\prod_{i=1}^{n} - p_i}$$

Se satisface exactamente la relación

$$z = e^{sT}$$

Ejemplo 4: Discretización de una planta por mapeo $z = e^{sT}$

Encuentre el modelo en tiempo discreto para la planta P(s) mostrada, con T = 0.1s

$$P(s) = \frac{4}{(s+1)(s+4)}$$

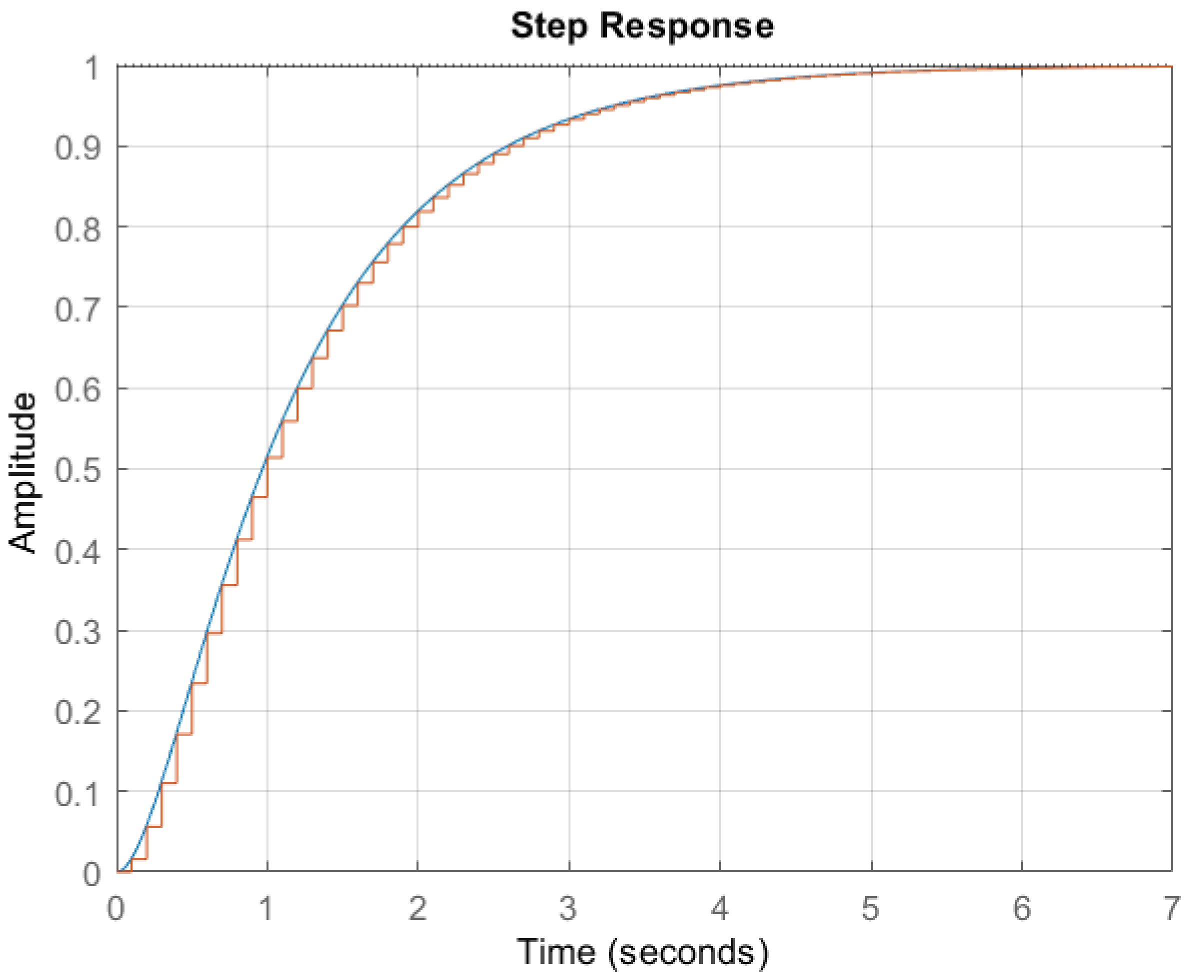
Los datos del sistema continuo son:

P = [-4 -1]

n = 2; q = 0

K_B = P(0) = 1

Ejemplo 4: Discretización de una planta por mapeo $z = e^{sT}$

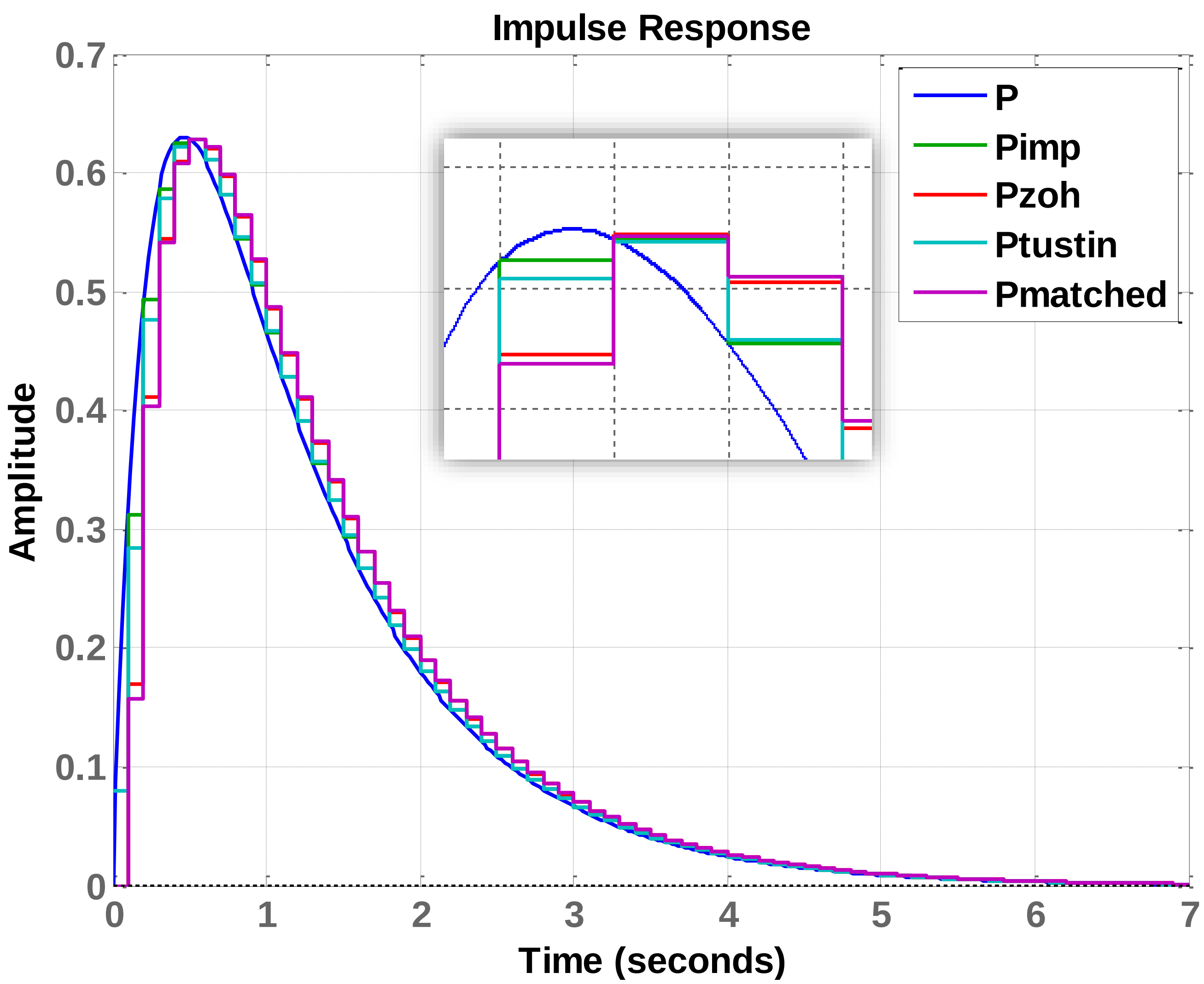



```
sysdm =

     0.015687 (z+1)
  ---------------------
  (z-0.9048) (z-0.6703)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

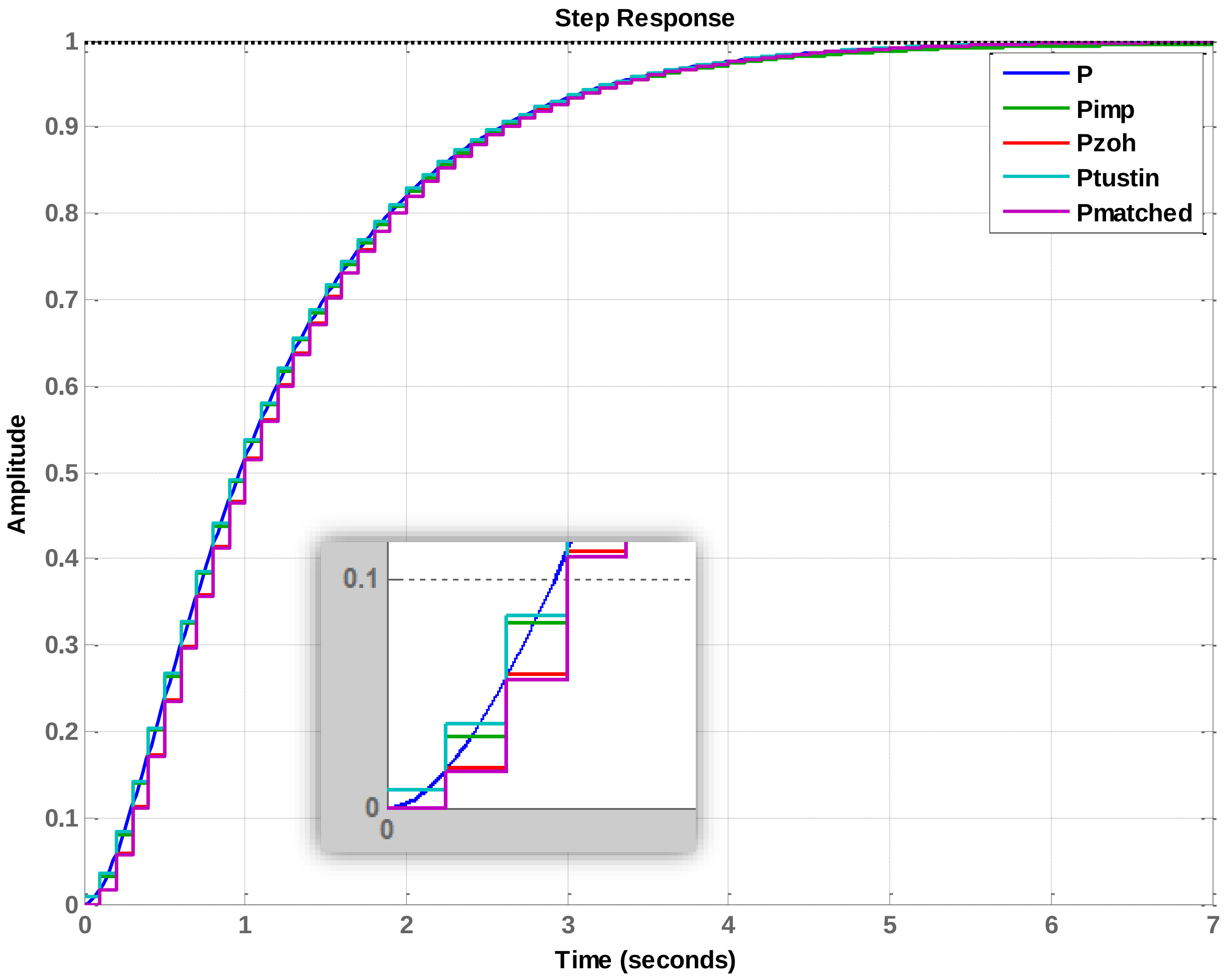
Código Matlab:

```
clc
clear all

z = []; %ceros de P(s)
p = [-1 -4]; %polos de P(s)
k = 4; %Constante de P(s)
T = 0.1; %Tiempo de muestreo
sys = zpk(z,p,k); %F.T. definida en ceros, polos y k
sysdm= c2d(sys,T,'Matched') %F.T. discretizado por el metodo
                            %Mapeos polos y ceros
step(sys,sysdm)  %Respuesta al escalon unitario.
grid
```

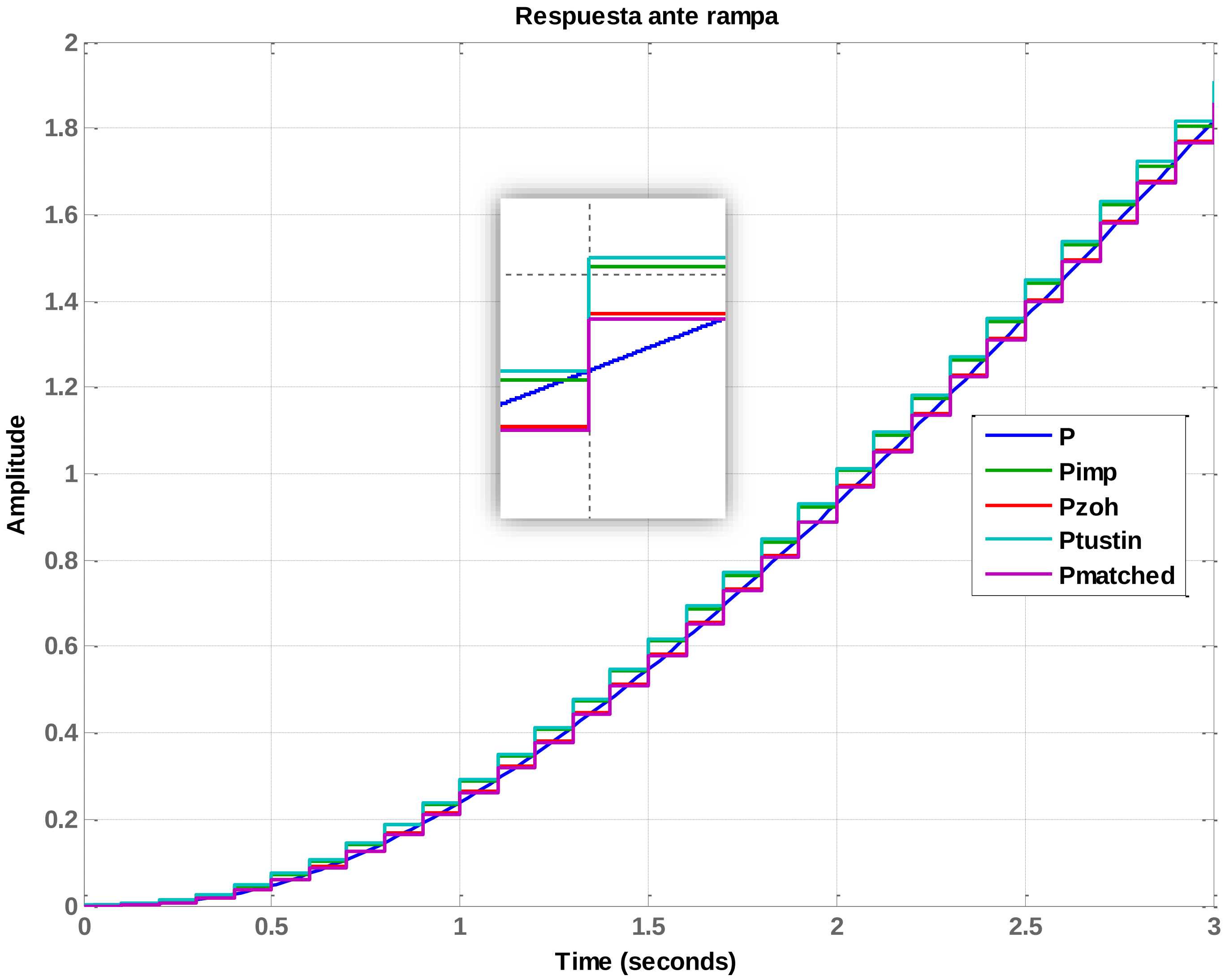
Respuesta al impulso



Respuesta al escalón

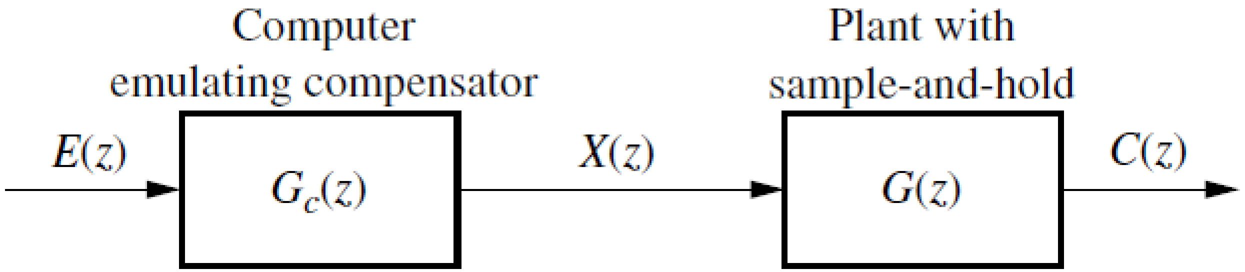


Respuesta ante rampa



Implementación Digital

Se puede implementar directamente mediante cálculos en el computador



Implementación Digital

Considere un compensador de segundo orden $G_c(z)$

$$G_c(z) = \frac{X(z)}{E(z)} = \frac{a_3 z^3 + a_2 z^2 + a_1 z + a_0}{b_2 z^2 + b_1 z + b_0}$$

Se multiplica en cruz de forma:

$$(b_2 z^2 + b_1 z + b_0) X(z) = (a_3 z^3 + a_2 z^2 + a_1 z + a_0) E(z)$$

Implementación Digital

Se despeja el termino de mayor orden de z en la salida X(z)

$$b_2 z^2 X(z) = (a_3 z^3 + a_2 z^2 + a_1 z + a_0)E(z) - (b_1 z + b_0)X(z)$$

Se dividen los términos del lado derecho entre el coeficiente de X(z)

$$X(z) = \left(\frac{a_3}{b_2} z + \frac{a_2}{b_2} + \frac{a_1}{b_2} z^{-1} + \frac{a_0}{b_2} z^{-2}\right)E(z) - \left(\frac{b_1}{b_2} z^{-1} + \frac{b_0}{b_2} z^{-2}\right)X(z)$$

Implementación Digital

Se se obtiene la transformada inversa de z para X(z)

$$
\begin{aligned}
x^*(t) \quad &= \frac{a_3}{b_2} e^*(t+T) + \frac{a_2}{b_2} e^*(t) + \frac{a_1}{b_2} e^*(t-T) + \frac{a_0}{b_2} e^*(t-2T) \\
&- \frac{b_1}{b_2} x^*(t-T) - \frac{b_0}{b_2} x^*(t-2T)
\end{aligned}
$$

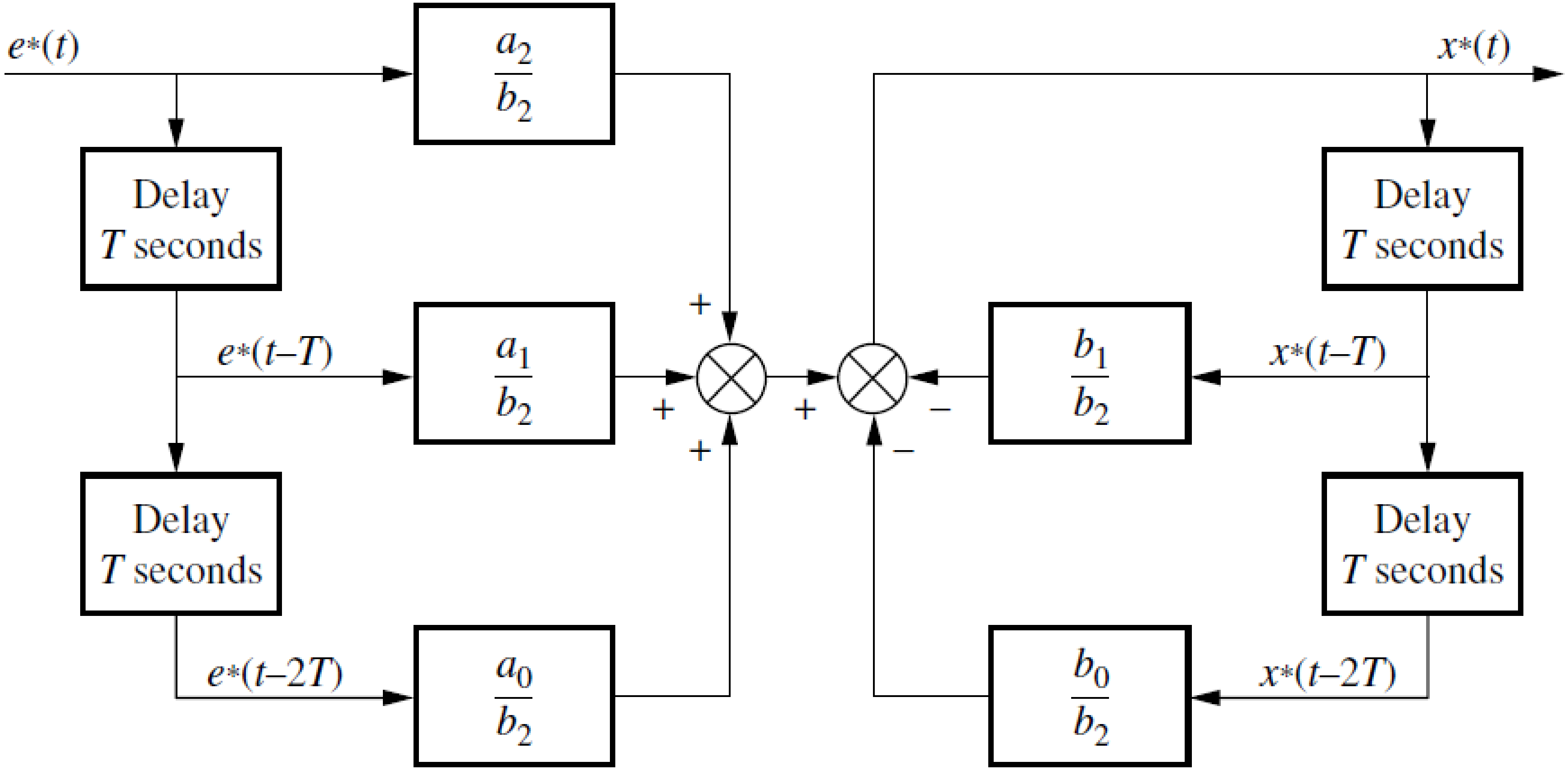
Se puede ver que en esta ecuación que la muestra actual de la salida del compensador, x(t), es una función de las muestras del futuro e(t + T) presente e(t) y pasado e(t - T) y e(t - 2T) de e (t), junto con los valores pasados de la salida, x(t - T) y x(t - 2T) . Si vamos a realizar físicamente este compensador, la salida no puede depender de valores futuros de la entrada. Por lo tanto, para ser físicamente realizable, a_3 debe ser igual a cero para que el valor futuro de e (t) sea cero.

Implementación Digital

Por lo tanto, la muestra de salida es una función de las muestras de entrada actuales y pasadas de la entrada, así como de las muestras pasadas de la salida.

$$x^*(t) = \frac{a_2}{b_2}e^*(t) + \frac{a_1}{b_2}e^*(t-T) + \frac{a_0}{b_2}e^*(t-2T) - \frac{b_1}{b_2}x^*(t-T) - \frac{b_0}{b_2}x^*(t-2T)$$

Implementación Digital



Ejemplo 5: Implementación Digital

Desarrolle el compensador digital implementable para el compensador descrito por la siguiente función de transferencia.

$$G_c(z) = \frac{X(z)}{E(z)} = \frac{z + 0.5}{z^2 - 0.5z + 0.7}$$

Ejemplo 5: Implementación Digital

Solución.

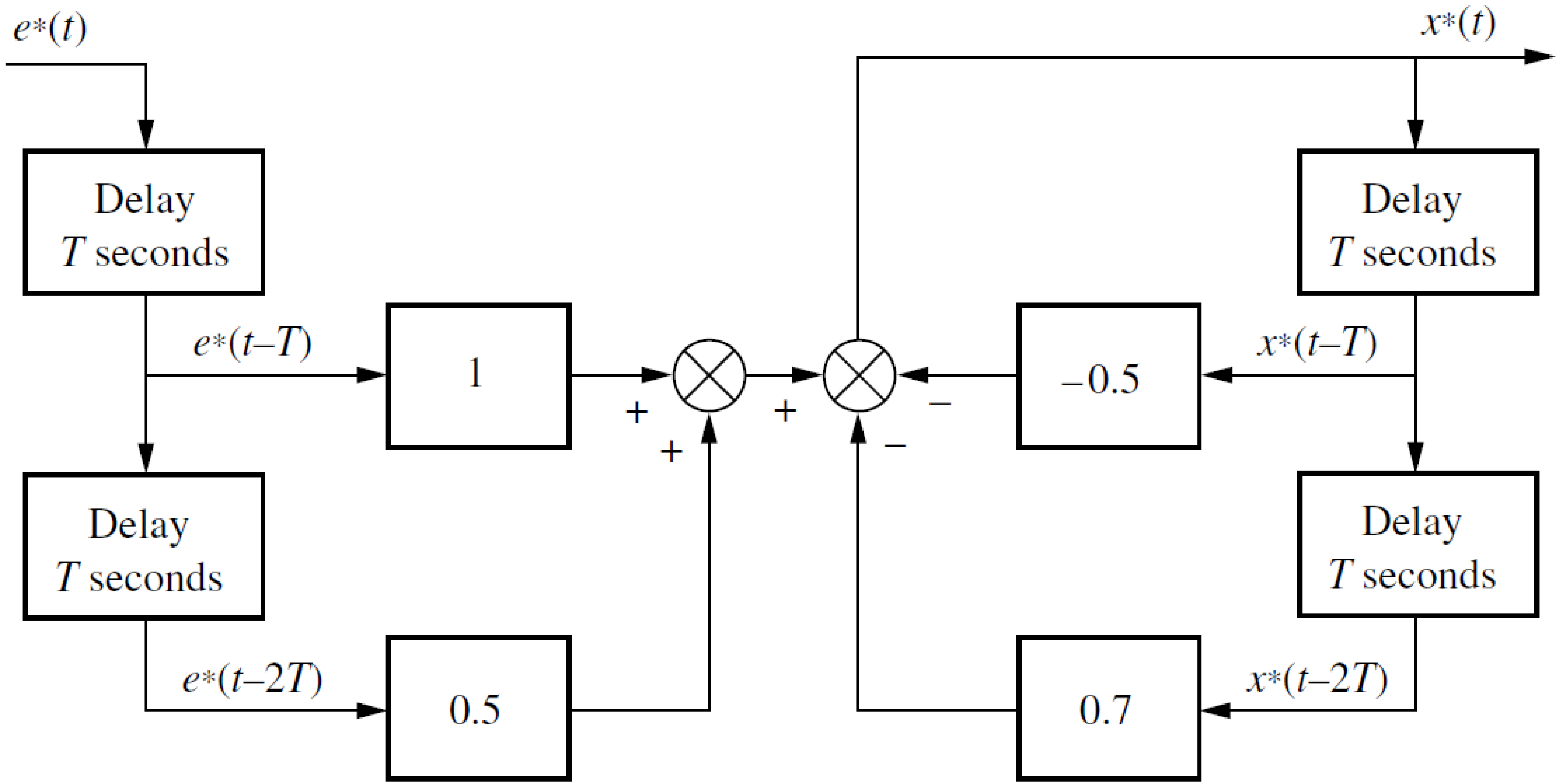
$$G_c(z) = \frac{X(z)}{E(z)} = \frac{z + 0.5}{z^2 - 0.5z + 0.7}$$

$$(z^2 - 0.5z + 0.7)X(z) = (z + 0.5)E(z)$$

$$z^2X(z) = (z + 0.5)E(z) - (-0.5z + 0.7)X(z)$$

$$X(z) = (z^{-1} + 0.5z^{-2})E(z) - (-0.5z^{-1} + 0.7z^{-2})X(z)$$

Ejemplo 5: Implementación Digital



Referencias

- Ogata, Katsuhiko. Ingeniería de Control Moderna, 5a. Ed. Prentice Hall, 2010, México.
- Nise, Norman. Control Systems Engineering, Sixth Edition.