Escuela Universitaria Politécnica - La Almunia
Centro adscrito
Universidad Zaragoza

22/23

REGULACIÓN Y CONTROL AUTOMÁTICO

EJERCICIOS RESULETOS

TRANSFORMADA-Z MATLAB C/C++

PROFESOR **Dra. Antolín Cañada, Diego**
AUTOR NAOUFAL EL RHAZZALI

UNIVERSIDAD DE ZARAGOZA

MATLAB

ARDUINO

Contenido

EJERCICIO 1

Enunciado

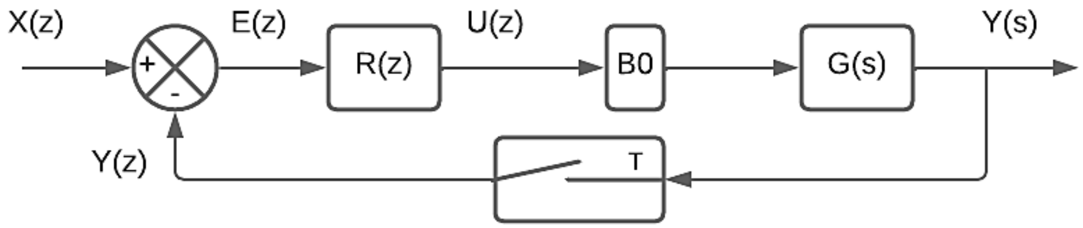


Diagrama 1 Sistema

Un sistema con realimentación unitaria, tiene la siguiente función de transferencia ***G(s)****:*

$$\boldsymbol{G(s) = \frac{5}{(s+2)(s+7)}}$$

Requerimientos

Descripción*: Se quiere que el sistema cumpla los siguientes requisitos:*

	Requerimientos
Error de posición (ep)	**0**
Sobreoscilación (SO)	**<50%**
Tiempo de respuesta (tr)	**<2s**

Tabla 1 Requerimientos de diseño

A Resolver

1. *Calcular el valor del periodo de muestreo, tomando $\mathcal{G}$ = 0.2 (1p)*
2. *Determinar la transformada Z del producto $B_0G(s)$ (2p)*
3. *Diseñar un regulador que verifique las especificaciones (4p)*
4. *Obtener la respuesta al escalón unitario de la FT en bucle cerrado (1p)*

De cara a implementación C/C++: Enlace
Teoría: Enlace

Antes de empezar

***Descripción**: ¿Qué es esto de muestrear y el tiempo discreto?*

El diseño de reguladores discretos para el control de un proceso se puede realizar de varias formas. No obstante, la idea de este documento es transformar la planta al mundo discreto y ya trabajar todo en el dominio "z". Pero ello requiere sumergirse en la teoría del Muestreo, así para poder entender diferentes expresiones matemáticas que aparecen en dicha transformación del mundo continuo en "s" al discreto en "z".

Muestreo

Sea Allah mediante, e(t) una señal cualquiera definida como se muestra en la siguiente figura.

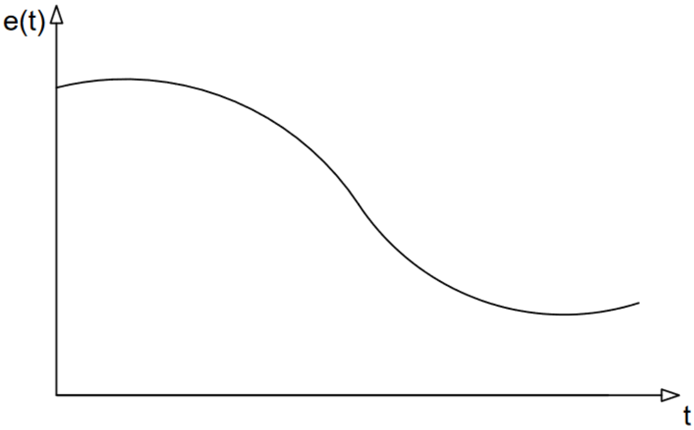


Figura 1 Señal de entrada cualquiera

Tras ser muestreada según un muestreo ideal, dónde $\Delta t \rightarrow 0$. Siendo Δt el tiempo que dura la toma de muestras. Si el muestreo se hace en un periodo T, se tiene el resultado de la figura siguiente.

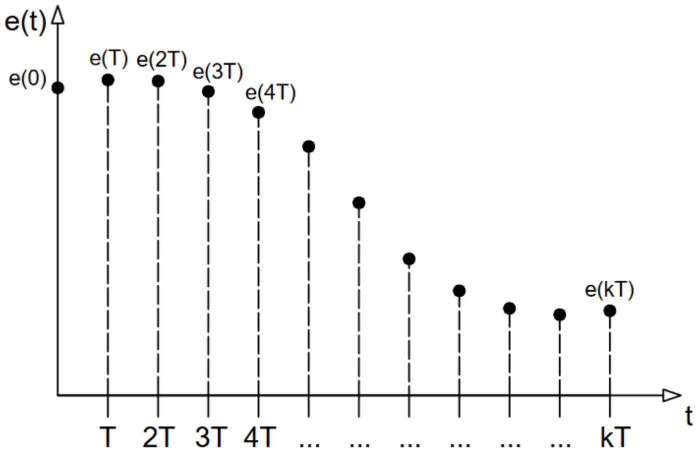


Figura 2 Muestreo de la señal de entrada con periodo T

El siguiente paso es reconstruir la señal, ello se hace mediante un bloqueador, o también llamado retenedor o extrapolador. En este caso se hará mediante un bloqueador de orden cero (B_0). Esta técnica consiste en bloquear la señal muestreada hasta la siguiente muestra. Así, la señal reconstruida se muestra en la siguiente figura.

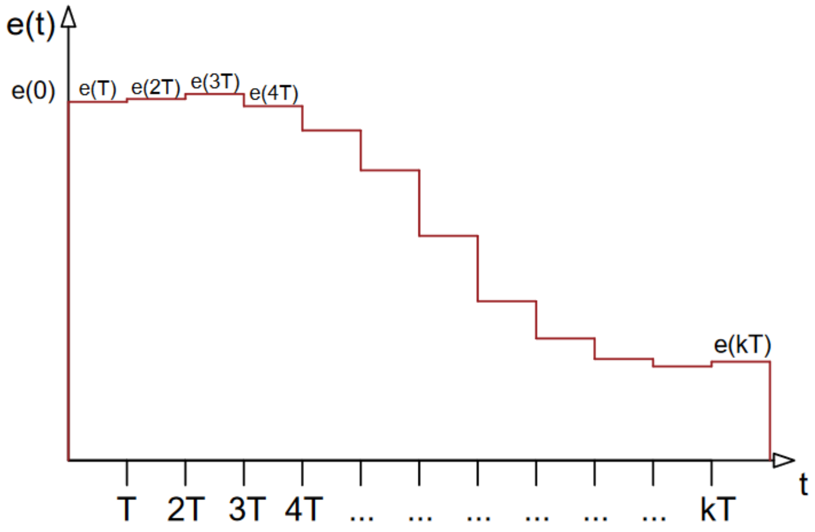


Figura 3 Reconstrucción de la señal muestreada mediante retención

Matemáticamente,

$$e(t) = cualesquier\ señal$$
$$e(0) = valor\ de\ la\ señal\ para\ t = 0$$
$$e(T) = valor\ de\ la\ señal\ para\ t = T$$
$$e(2T) = valor\ de\ la\ señal\ para\ t = 2T$$
$$e(3T) = valor\ de\ la\ señal\ para\ t = 3T$$
$$e(kT) = valor\ de\ la\ señal\ para\ t = kT$$

Evidentemente solo sobreviven los valores de la señal en los tiempos múltiples enteros del periodo de muestreo, que son los valores que han sido muestreados. Así, para un periodo de muestreo dado: $e(kT) = valor\ de\ la\ señal\ para\ t = kT$

Reconstrucción

En la reconstrucción para el primer periodo, siendo el bloqueador de orden cero, se pretende retener durante un periodo de muestreo completo la muestra anterior. Ello, se hace mediante una ventana, generada a base de la función escalón unitario (Heaviside).

$$Funcion\ Escalón\ unitario = u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & cualquier\ otro\ caso \end{cases}$$

$$u_T(t) = u(t-T) = \begin{cases} 1 & (t-T) \geq 0 \\ 0 & cualquier\ otro\ caso \end{cases}$$

$$ventana = u(t) - u(t-T) = \begin{cases} 1 & 0 \leq t < T \\ 0 & cualquier\ otro\ caso \end{cases}$$

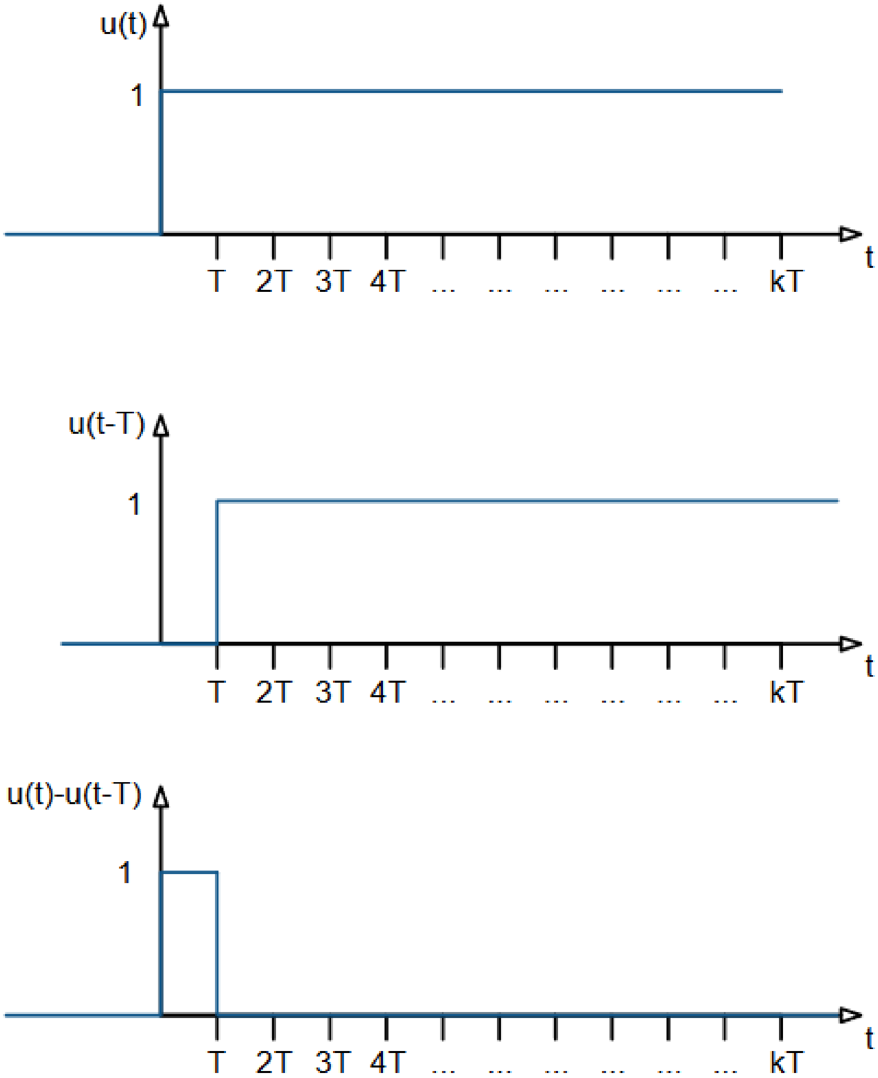


Figura 4 Función ventana a partir de escalones unitarios

Como se indica en la Figura 4, se consigue una ventana a partir de un escalón unitario restado de otro retrasado un periodo.

Así, la señal reconstruida se denotaría como $\bar{e}(t)$

$$\boldsymbol{\bar{e}(t) = e(0) \cdot [u(t) - u(t-T)] + e(1) \cdot [u(t-T) - u(t-2T)] +, \dots, + e(kT) \cdot [u(t-kT) - u(t-(k+1)T)]}$$

Hágase notar que multiplicamos el valor de la señal original muestreada para un periodo “kT”, por la ventana que la retiene hasta el siguiente instante de toma de muestras. Así, la expresión general para la señala reconstruida es:

$$\bar{e}(t) = e^*(t) = \sum_{k=0}^{\infty} e(kT) \cdot [u(t-kT) - u(t-(k+1)T)]$$

Lo siguiente es hacer la transformada de Laplace de $\bar{e}(t)$, quedando fuera $e(kT)$ porque es una constante, y separando la resta, dado que la transformada de Laplace cumple el principio de linealidad. Consultando tablas:

$$U(s) = \mathcal{L}[u(t)] = \frac{1}{s}$$

El retraso en el dominio del tiempo se traduce a e^{-as}, donde "a" cuantifica cuánto está retrasada la función dada. Así:

$$U(s) = \mathcal{L}[u(t-a)] = \frac{1}{s}e^{-as}$$

$$E(s) = \mathcal{L}[\bar{e}(t)] = \sum_{k=0}^{\infty} e(kT) \cdot \mathcal{L}[u(t-kT)] - \mathcal{L}[u(t-(k+1)T)]$$

$$\mathcal{L}[u(t-kT)] = \frac{1}{s}e^{-kTs}$$

$$\mathcal{L}[u(t-(k+1)T)] = \frac{1}{s}e^{-(k+1)Ts}$$

$$E(s) = \mathcal{L}[\bar{e}(t)] = \sum_{k=0}^{\infty} e(kT) \cdot \left(\frac{1}{s}e^{-kTs} - \frac{1}{s}e^{-(k+1)Ts}\right) = \sum_{k=0}^{\infty} e(kT) \cdot \left(\frac{e^{-kTs} - e^{-(k+1)Ts}}{s}\right)$$

$$E(s) = \sum_{k=0}^{\infty} e(kT) \cdot \left(\frac{e^{-kTs} - e^{-kTs}e^{-Ts}}{s}\right) = \sum_{k=0}^{\infty} e(kT) \cdot e^{-kTs}\left(\frac{1 - e^{-Ts}}{s}\right)$$

$$\boldsymbol{E(s) = \left(\frac{1 - e^{-Ts}}{s}\right)\sum_{k=0}^{\infty} e(kT) \cdot e^{-kTs}}$$

Esta expresión a la que acabamos de llegar es muy importante, y vamos a reflexionar acerca de sus componentes, que son principalmente dos términos:

1º Término: $\boldsymbol{\sum_{k=0}^{\infty} e(kT) \cdot e^{-kTs}}$

Es el valor de amplitud de la muestra tomada, $\boldsymbol{e(kT)}$, multiplicado por $\boldsymbol{e^{-kTs}}$, que traducido del dominio "**s**" al dominio del tiempo "**t**"; es un retraso de $\boldsymbol{kT}$, es decir, al final esa sumatoria acabaría generando los impulsos, antes mostrados en la Figura 2.

2º Término: $\boldsymbol{\left(\frac{1-e^{-Ts}}{s}\right)}$

Es el retenedor, este a partir de los impulsos genera una función continua (Figura 3) mediante la técnica de retención discutida arriba.

RESUMEN	
Función muestreada	$E^*(s) = \sum_{k=0}^{\infty} e(kT) \cdot e^{-kTs}$
Bloqueador de orden 0 ó Retenedor de orden 0 ó Extrapolador de orden 0	$B_0 = \dfrac{1 - e^{-Ts}}{s}$

Tabla 2 Resumen de las funciones muestreador y bloqueador

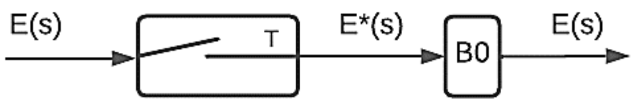


Diagrama 2 Función muestreada y reconstruida

Aclaración: E(s) no existe físicamente, ni tampoco $\frac{1-e^{-Ts}}{s}$. esta última no es función de transferencia de ningún dispositivo real.*

Muestreador ideal

Para definir el muestreador ideal se ha de recordar que,

$$\mathcal{L}^{-1}[e^{-kTs}] = \delta(t - kT)$$

Donde $\delta(t - kT)$, es la función delta de Dirac o función impulso unitario, retrasada kT, respecto de t.

$$\delta(t - kT) = \begin{cases} 1 & t = kT \\ 0 & cualquier\ otro\ caso \end{cases}$$

Por lo tanto, $e^*(t)$ sería la expresión que define al muestreador ideal en dominio del tiempo.

$$e^*(t) = \mathcal{L}^{-1}[E^*(s)] = \mathcal{L}^{-1}\left[\sum_{k=0}^{\infty} e(kT) \cdot e^{-kTs}\right] = \sum_{k=0}^{\infty} e(kT) \cdot \delta(t - kT)$$

De la expresión anterior, se puede interpretar el muestreador ideal como un modulador impulsivo, esto es, un tren de pulsos unitario que son multiplicados por las muestras en cada kT, dando lugar a la señal original muestreada mediante pulsos de diferentes amplitudes.

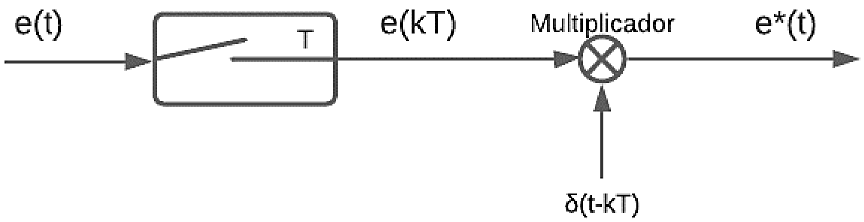


Diagrama 3 Proceso de muestreo ideal

Llegados a este punto entendemos que muestrear una señal continua es muestrear dicha señal cada periodo T, y a esa secuencia multiplicarla por un tren de impulsos unitarios. Entonces para una señal de entrada e(t), su equivalente señal muestreada $e^*(t)$, se define como sigue.

$$e(t) \cdot \sum_{k=0}^{\infty} \delta(t - kT)$$

Se quisiéramos encontrar la transformada de Laplace de lo anterior, procederíamos como sigue,

$$E^*(s) = \mathcal{L}[e^*(t)] = \mathcal{L}\left[e(t) \cdot \sum_{k=0}^{\infty} \delta(t - kT)\right] = E(s) * \Delta_T(s)$$

En la expresión de arriba, se ha empalado la propiedad que nos informa de la que multiplicación de dos funciones en el dominio del tiempo se convierte en la convolución entre dichas funciones en el dominio de Laplace. Siendo,

$$\Delta_T(s) = \mathcal{L}\left[\sum_{k=0}^{\infty} \delta(t - kT)\right]$$

Por otro lado, y dado que “s” o la nueva variable “p” son variables complejas, se recurre a la fórmula Laurent, que nos informa de la solución de una integral evaluada para una variable compleja a lo largo de un camino cerrado entorno a una singularidad es,

$$E^*(s) = E(s) * \Delta_T(s) = \int_{s-jw}^{s+jw} E(p)\, \Delta_T(s-p) dp = \frac{1}{2\pi j}\int_{c-jw}^{c+jw} E(p)\, \Delta_T(s-p) dp$$

Donde: $\boldsymbol{c} \in \boldsymbol{R} \,/\, (\boldsymbol{c}+\boldsymbol{j\omega}) \in \boldsymbol{ROC\ de\ E(p)}\ \forall -\infty < \boldsymbol{\omega} < \infty$

Si además resolvemos la sumatorio de tren de pulsos consultando tablas, encontramos,

$$\boldsymbol{\Delta_T(s) = \sum_{k=0}^{\infty} e^{-kTs} = \frac{1}{1-e^{-Ts}}}$$

Así,

$$E^*(s) = \frac{1}{2\pi j}\int_{c-jw}^{c+jw} E(p)\boldsymbol{\frac{1}{1-e^{-T(s-p)}}} dp = \sum_{Polos\ E(p)} \left[residuos\ (E(p)\boldsymbol{\frac{1}{1-e^{-T(s-p)}}})\right]$$

Finalmente,

$$E^*(s) = \sum_{Polos\ E(p)} \left[residuos\ (E(p)\boldsymbol{\frac{1}{1-e^{-Ts}e^{Tp}}})\right]$$

Pasando a Z[]

Qué pasa si queremos pasar dicha señal muestreada del dominio “s” al dominio “z”.

Primero volvamos a la definición formal de la transformada Z de una secuencia dada, que se muestra a continuación.

$$\boldsymbol{E(z) = Z[\{e(k)\}] = \sum_{k=0}^{\infty} e(k)z^{-k} = e(0)z^{-0} + e(1)z^{-1} + \cdots = e(0) + e(1)z^{-1} + \cdots}$$

Podemos ver que esto nos suena a algo, ¿no?

$$E^*(s) = \sum_{k=0}^{\infty} e(kT)\cdot e^{-kTs} = \boldsymbol{e(0T)z^{-0T} + e(T)z^{-T} + \cdots = e(0) + e(T)z^{-T} + \cdots}$$

Por simple inspección, encontramos que lo que sigue,

$$E(z) = E^*(s)|_{e^{sT}=z} = \sum_{Polos\ E(p)} \left[residuos\ (E(p)\frac{1}{1-z^{-1}e^{Tp}})\right]$$

Finalmente, hemos llegado a la expresión que tanto anhelábamos, dada una función en “s” sabremos discretizarla. Ahora nos encontramos preparados para afrontar el enunciado y resolverlo.

Resolución

Apartado-1

***Descripción**: Calcular el valor del periodo de muestreo, tomando ξ = 0.2*

1º Método

$$T_1 = \frac{tr}{10} = \frac{2}{10} = 0.2s$$

2º Método

$$T_2 = \frac{\pi}{5\omega_d} = \frac{\pi}{5\omega_n\sqrt{1-\xi^2}} = \frac{\pi}{5\frac{\sigma}{\xi}\sqrt{1-\xi^2}} = \frac{\pi}{5\frac{4}{\frac{tr}{\xi}}\sqrt{1-\xi^2}} = \frac{\pi}{5\frac{4}{t_r\xi}\sqrt{1-\xi^2}} = \frac{\pi t_r\xi}{20\sqrt{1-\xi^2}}$$

$$T_2 = \frac{\pi\cdot 2\cdot 0.2}{20\sqrt{1-(0.2)^2}} = 0.064127$$

Obtenidos dos periodos de muestreo diferentes, se opta por el mínimo valor,

$$T = min(T_1, T_2) = 0.064127$$

$$\boldsymbol{T = 0.064127s}$$
$$\boldsymbol{T \approx 64\ ms}$$
$$\boldsymbol{T \approx 64127\ \mu s}$$

Apartado-2

***Descripción**: Determinar la transformada Z del producto $B_0G(s)$*

Antes de empezar a resolver, hay que hacer tener claro, o al menos suponer que, la planta G(s) es la señal pulsada de la planta en el dominio "s". Esto es porque dicha señal viene de la salida del bloqueador, el cual reconstruye la señal a base de retenciones de muestras. Esto último se encuentra detalladamente explicado en el apartado Reconstrucción (pincha en ello).

Siendo asumido lo anterior, una señal muestreada cumple lo que sigue,

$$G^*(s) = \sum_{Polos\ G(p)}\left[residuos\ (G(p)\frac{\mathbf{1}}{\mathbf{1 - e^{-Ts}e^{Tp}}})\right]$$

Siendo lo que se busca es la discretización, y como se ha demostrado en apartados anteriores, G(z) es,

$$G(z) = G(s)|_{e^{sT}=z} = \sum_{Polos\ G(p)}\left[residuos\ (G(p)\frac{1}{1 - z^{-1}e^{Tp}})\right]$$

No obstante, lo que se pretende encontrar es la transformada zeta no solo de G(s), sino $B_0G(s)$. Por ello nuestra nueva función pulsada es $G_s(s)$.

$$G_s(s) = B_0G(s) = \frac{1-e^{-sT}}{s}\frac{5}{(s+2)(s+7)}$$

$$G_s(z) = G_s(s)|_{e^{sT}=z} = \sum_{Polos\ G_s(p)} \left[\frac{1-z^{-1}}{p}\frac{5}{(p+2)(p+7)}\frac{1}{1-z^{-1}e^{Tp}})\right]$$

Finalmente, todo lo que no depende de "p", puede salir fuera o quedar dentro, como más cómo nos sea. En todo caso, la variable de evaluación es "p", y todo lo que no sea "p" es constante.

Generalmente, se suele escribir la expresión última de arriba como sigue,

$$G_s(z) = G_s(s)|_{e^{sT}=z} = (1-z^{-1}) \sum_{Polos\ G_s(p)} \left[\frac{5}{p(p+2)(p+7)}\frac{1}{1-z^{-1}e^{Tp}})\right]$$

$$\boldsymbol{Polinomio\ característico \rightarrow p(p+2)(p+7) = 0}$$
$$\boldsymbol{Polos \rightarrow |p_1 = 0\ |\ p_2 = -2\ \ |\ p_3 = -7|}$$

Llegado a este punto se procede a aplicar el método de los residuos:

$$res|_{p=p_j} = \frac{1}{(r-1)!}\lim_{p\to p_j}\frac{d^{r-1}}{dz^{r-1}}\left[(p-p_j)^r G_s(p)\frac{z}{z-e^{pT}}\right]$$

$$res|_{p=0} = \lim_{p\to 0}\left[(p-0)\frac{5}{p(p+2)(p+7)}\right] = \lim_{p\to 0}\left[\frac{5}{(p+2)(p+7)}\frac{z}{z-e^{sT}}\right] = \frac{5}{14}\frac{z}{z-1}$$

$$\boldsymbol{res|_{p=0} = \frac{5}{14}\left[\frac{z}{z-1}\right]}$$

$$res|_{p=-2} = \lim_{p\to 0}\left[(p+2)\frac{5}{p(p+2)(p+7)}\right] = \lim_{p\to -2}\left[\frac{5}{p(p+7)}\frac{z}{z-e^{sT}}\right] = -\frac{5}{10}\frac{z}{z-e^{-2T}}$$

$$\boldsymbol{res|_{p=-2} = -\frac{1}{2}\left[\frac{z}{z-e^{-2T}}\right]}$$

$$res|_{p=-7} = \lim_{p\to -7}\left[(p+7)\frac{5}{p(p+2)(p+7)}\right] = \lim_{p\to -7}\left[\frac{5}{p(p+2)}\frac{z}{z-e^{sT}}\right] = \frac{5}{35}\frac{z}{z-e^{-7T}}$$

$$\boldsymbol{res|_{p=-7} = \frac{1}{7}\left[\frac{z}{z-e^{-7T}}\right]}$$

$$G_s(z) = (1-z^{-1})\left[\left(res|_{p=0}\right) + \left(res|_{p=-2}\right) + \left(res|_{p=-7}\right)\right]$$

$$G_s(z) = \frac{0.008518z + 0.007028}{z^2 - 1.5179z + 0.5615}$$

$$\boldsymbol{G_s(z) = 0.008518\left[\frac{z-0.82550}{(z-0.88028)(z-0.6385)}\right]}$$

Verificación -Matlab

Para verificar todo este proceso por medio de computador, se recurre a Matlab en su versión online. Lo más fácil sería recurrir a la función $\boldsymbol{c2d(Gs, T)}$, y emplear el siguiente script.

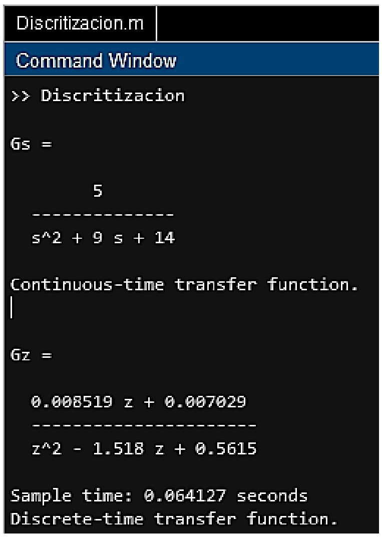
```
% Periodo de Muestreo
Ts=0.064127;

% Planta G(s)
num = [5]
den = [1 9 14]
Gs = tf(num, den)

% Continous to Discret
Gz=c2d(Gs, Ts)
```

Script. 1 – Discretización
(Realizado por Naoufal El Rhazzali)

Resultado de la ejecución



```
Discritizacion.m
Command Window
>> Discritizacion

Gs =

        5
  -------------
  s^2 + 9 s + 14

Continuous-time transfer function.

Gz =

  0.008519 z + 0.007029
  ---------------------
  z^2 - 1.518 z + 0.5615

Sample time: 0.064127 seconds
Discrete-time transfer function.
```

Figura 5 – Resultado de la ejecución del Script. 1

No obstante, la metodología que plantea este documento es la de ir resolviendo la discretización paso a paso, y encontrando scripts que verifican dichos pasos. Esto último, brindará al alumno la capacidad de verificar los ejercicios de diseño de reguladores de cara al examen de la asignatura.

Entonces, repitamos todos los apartados anteriores (Apartado-1, Apartado-2), pero usando Matlab.

¡Vamos allá!

Apartado-1 (Matlab)

***Descripción**: Calcular el valor del periodo de muestreo, tomando $\mathcal{G}$ = 0.2*

Para resolver este apartado, se implementa "***get_samplingTime**(tr, dampingFactor)*", la cual es una función implementada en Matlab, que recibe dos argumentos, *tr* (tiempo de asentamiento) y *dampingFactor* (Factor de amortiguamiento o $\mathcal{G}$). Lo que hace la función es calcular el periodo de muestreo, a partir del tr y el $\mathcal{G}$ dados, ello con los dos métodos presentados ya arriba. Finalmente, devuelve el valor más pequeño hallado.

```
function [T] = get_samplingTime(tr, dampingFactor)
%% Sampling time calculationing
    % First method
    % Sampling time:
    T1=tr/10;

    % Second method
    % ζ = dampingFactor
    % tr = establishment time

    % sigma = σ
    sigma = 4/tr;
    % wn = parte real del polo (Hz), y fctAmortg = ζ
    wn = sigma/dampingFactor;
    % wd = parte imaginaria del polo (Hz)
    wd=wn*sqrt(1-((dampingFactor)^2));
    % Sampling time:
    T2=pi/(5*wd);

    % The optimal sampling time
    T=min(T1, T2);
end
```

Fun. 1 función get_samplingTime(tr, dampingFactor)
(Implementado por Naoufal El Rhazzali)

Apartado-2 (Matlab)

```
syms z s T
% Requerimientos
fctAmortg=0.2;
tr=2;
SO=0.5;
ep=0;
% Tiempo de Muestreo
T = get_samplingTime(tr, fctAmortg);
%% OBTENER TRANSFORMADA-Z DE UNA FUNCIÓN PULSADA DADA EN "S"
% Defina su bloqueador:
B0 = (1-z^-1)/s;
% Defina su función pulsada de la Planta
Gs = 5/((s+2)*(s+7));
% Definimos el tren de pulsos o modulador impulsivo (VER TEORÍA)
TrenPulsosUnitario = (z/(z-exp(vpa(s*T,5))));
% Función resultante
Gs_s=vpa(B0*Gs*TrenPulsosUnitario, 5);
% Obtención de las raíces y no olvidar el s=0, que se genera de lo que
% queda el bloqueador en "s"
[numGss, denGss] = numden(Gs);
raices = roots(fliplr(coeffs(denGss)));
raices(end+1) = 0;
% La transformada Z de B0*G(s), es la sumatoria de los residuos
% de B0*G(s)*A, siendo A el término adicional. los residuos son
% obtenidos tras evaluar los polos Gs_s
res1 = res(Gs_s, raices(1), 1);
res2 = res(Gs_s, raices(2), 1);
res3 = res(Gs_s, raices(3), 1);

% Obtención final de la planta discretizada
Gs_z = collect(res1+res2+res3)
pretty(Gs_z)
```

Script. 2 – Discret_PasoAPaso
(Realizado por Naoufal El Rhazzali)

El Script. 2 parte de la filosofía de que el alumno tenga que modificar mínimamente el script. Por ello, se empieza por dar carácter simbólico a “z”, “s” y “T”.

El siguiente paso es declarar los requerimientos del sistema, y obtener el periodo de muestreo con la función implementada anteriormente.

Lo siguiente es declarar el bloqueador, que tendrá diferentes expresiones según su orden (0, 1, 2, etc. 3 y órdenes mayores son menos habituales debido a los retardos que introducen en el sistema). Se sigue con la definición de la función de transferencia (FT) de la planta como la da el enunciado y se concluye con la declaración de aquel tren de pulsos que modula la señal (ver Muestreador ideal)

Lo siguiente es obtener ese productorio del cual se acabaría sacando la transformada Z. Hágase notar que se han omitido simplificaciones.

El siguiente paso es obtener los polos para los cuales obtendremos los residuos correspondientes, para ello, primero se le saca el denominador a la planta del enunciado, mediante la función ***numden()***. Más tarde se sacan los coeficientes de dicho denominador obtenido mediante ***coeffs()***, los cuales se recogen en un vector que se vierte su orden, mediante ***fliplr()***, para luego poder operar con ellos en Matlab. Finalmente, se obtienen las raíces con la función ***roots()*** de Matlab y se añade un 0 al final de dicho array de raíces final (por $\frac{1}{s}$ que introduce el bloqueador).

El siguiente paso, sería ya obtener el residuo para una raíz dada (se puede hacer un bucle for, no obstante, en el script se hace de forma manual).

Finalmente, se suman los residuos y se les aplica ***collect()*** para tener un función en condiciones, y se imprime en pantalla por medio de la ***pretty()***, que arroja el resultado en una forma elegante.

El único problema de este script es que no consigue obtener una función con el denominador normalizado y expresado en forma numérica. Ello, queda para los colaboradores que puedan mejorar este documento para el futuro.

Antes de mostrar el resultado, se hablará de cómo se implementó la función res(Gs, p), que es implementada por Naoufal El Rhazzali, igualmente.

```
function [ri] = res(Gs, p, r)
%% Residue calculation
    % Where “r” is the multiplicity of the pole
    syms s;
    syms T;
    syms z;
    ri=(1/factorial(r-1))*limit(diff(Gs*(s-p),s,r-1), s, p);
end
```

Fun. 2 función res(Gs, p)
(Implementado por Naoufal El Rhazzali)

Dicha función se limita a aplicar la fórmula de los residuos que aparece en el formulario de la asignatura (Ver Anexos).

Apartado-3

***Descripción**: Diseñar un regulador que verifique las especificaciones:*
Ep=0, SO<50%, tr<2”

Para diseñar el regulador R(z), que cumpla las especificaciones existen diversas metodologías a seguir. No obstante, en este documento se procede como se ha visto en clase con el profesor Diego Antolín. Así, se emplea el método de Truxal o también llamado de síntesis directa.

Método de la síntesis directa

1º Paso: Eliminación de polos y ceros estables

Para ello, habrá que saber cuándo es estable o no un polo o un cero. La respuesta es que cuando un polo está dentro de la circunferencia unidad en el plano imaginario es estable, esto es, su módulo es estrictamente menor que la unidad, en cualquier otro caso, estaríamos ante un polo o cero inestable o marginalmente estable.

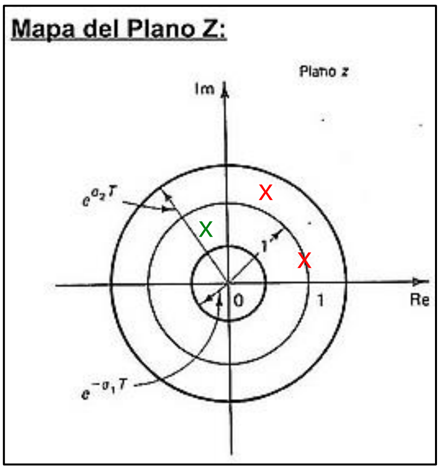


Figura 6 Plano Z y regiones de convergencia

Los polos o ceros señalados en rojo en la Figura 6, son inestables o marginalmente estables, y los polos o ceros señalados en verde son estables.

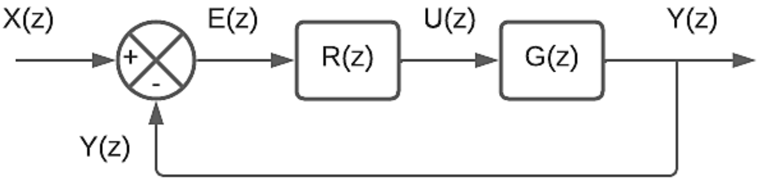
FT Planta Discretizada	$G(z) = 0.008518\left[\frac{z-0.82550}{(z-0.88028)(z-0.6385)}\right]$
Ceros	$z = 0.82550$
Polos	$p_1 = 0.88028$
	$p_2 = 0.6385$

Entonces eliminamos, todos los polos y ceros:

$$R(z) = \frac{(z-0.88028)(z-0.6385)}{(z-0.82550)}$$

Por lo pronto, ya tenemos parte del regulador.

2º Paso: Análisis estacionario (Steady State)



$$G(z) = \frac{k}{(z-1)^N} \frac{\prod_{i=1}^{m}(z-z_i)}{\prod_{j=1}^{n}(z-z_j)} \ siendo, \begin{cases} z_i \neq 1 \\ z_j \neq 1 \\ z_i \neq z_j \end{cases} \forall (i,j)$$

Siendo, $(z-1)^N$ los integradores del sistema, esto es, polos del sistema con módulo 1. Así N, determina el tipo del sistema y determina el error de posición del mismo en el régimen estacionario.

$$N = 0 \rightarrow Sistema\ de\ tipo\ 0$$
$$N = 1 \rightarrow Sistema\ de\ tipo\ 1$$
$$N = 2 \rightarrow Sistema\ de\ tipo\ 2$$
$$\dots$$
$$N = n \rightarrow Sistema\ de\ tipo\ n$$

Tipo del sistema (N)	$Error_{posición}$	$Error_{velocidad}$	$Error_{aceleración}$
0	$\frac{1}{1+k_p}$	∞	∞
1	0	$\frac{T}{k_v}$	∞
2	0	0	$\frac{T^2}{k_a}$
$N \geq 3$	0	0	0

Al final el objetivo es que el error de posición sea nulo. En dicho caso, para cumplir con ello, la FT en bucle abierto debe tener un integrador (sistema de tipo 0 como mínimo). Al no tener la función de la planta discretizada ningún integrador, el regulador ha de tenerlo para cumplir con un error de posición nulo en estado estacionario. Por lo pronto el regulador adquiere el siguiente aspecto:

$$R(z) = \frac{(z-0.88028)(z-0.6385)}{(z-0.82550)(z-1)}$$

3º Paso: Análisis transitorio (Transient State)

La estrategia a seguir en este paso es ir añadiendo términos al regulador R(z), tal que se acabe cumpliendo con los requerimientos del sistema.

Requerimientos	S.O. < 50% tr < 2s
Posibles términos a añadir	$k, \frac{1}{z+a}, (z+b), \frac{z+a}{z+b}, etc.$

$$R(z) = k\frac{(z-0.88028)(z-0.6385)}{(z-0.82550)(z-1)}$$

En la expresión última de arriba, se ve que se ha optado por añadir un término k al regulador. Así, ya tenemos el regulador diseñado. No obstante, una vez se tiene el regulador, hay que cerciorase de que el regulador sea causal: Grado del denominador mayor o igual que el grado del numerador. En este caso, R(z) hallado es casual.

El siguiente paso es hallar la función de transferencia en bucle cerrado, cuyos polos son (ecuación característica del sistema) los que determinan la respuesta transitoria del sistema.

$$\frac{Y(z)}{X(z)} = FT_{BucleCerrado} = \frac{R(z)G(z)}{1 + R(z)G(z)}$$

$$FT_{BucleCerrado} = \frac{k\frac{(z-0.88028)(z-0.6385)}{(z-0.82550)(z-1)} \cdot \frac{0.008518 \cdot (z-0.82550)}{(z-0.88028)(z-0.6385)}}{1 + k\frac{(z-0.88028)(z-0.6385)}{(z-0.82550)(z-1)} \cdot \frac{0.008518 \cdot (z-0.82550)}{(z-0.88028)(z-0.6385)}}$$

$$FT_{BucleCerrado} = \frac{k\frac{0.008518}{z-1}}{1 + k\frac{0.008518}{z-1}} = \frac{\frac{0.008518 \cdot k}{z-1}}{\frac{(z-1) + 0.008518 \cdot k}{z-1}} = \frac{0.008518 \cdot k}{z + (0.008518 \cdot k - 1)}$$

Ecuación característica del sistema: $\boldsymbol{z + (0.008518 \cdot k - 1) = 0}$

La ecuación característica nos indica que el sistema es de 1º Orden, por ello la SobreOscilación es nula, cumpliendo con ese **SO<50%**. Lo que quedaría es asegurar un tiempo de asentamiento menor que el 2s.

En el caso de los sistemas continuos, los polos siempre presentan la siguiente forma,

$$s_j = \sigma_j \pm j\omega_d$$

En el caso del sistema que nos ocupa, la parte imaginaria es nula dado que el sistema es de 1º Orden. Así,

$$s_j = \sigma_j$$

Por otra parte, sabemos que,

$$tr = \frac{4}{\sigma} \to \sigma = \frac{4}{tr} = \frac{4}{2} = 2$$

$$\boldsymbol{\sigma = 2 \ \to s = 2}$$

Por otra parte, sabemos, $\boldsymbol{z_i = e^{sT} = e^{2T} = e^{2 \cdot 0.064127} = 1.1368}$

$$\boldsymbol{1.1368 + (0.008518 \cdot k - 1) = 0}$$

$$\boldsymbol{k = -16.0601}$$

$$FT_{BucleCerrado} = -\frac{0.1368}{z - 1.1368}$$