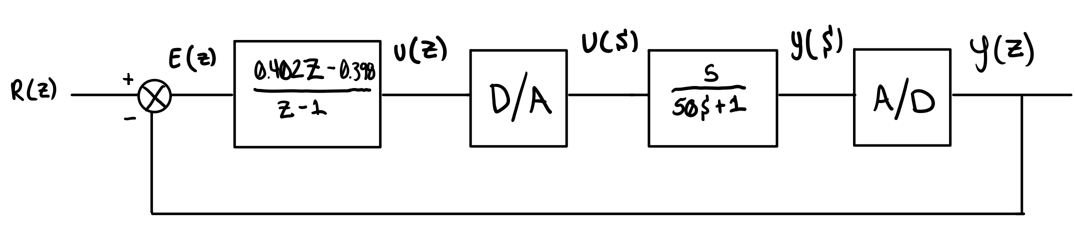
Examen P2

27/10/18 0:29



ExamenParcial2_Contr...

Gerardo Daniel Naranjo Gallegos, A01209499
Roberto Figueroa Saavedra, A01209689
27/10/2018

Segundo examen parcial simulado en MATLAB

Teniendo el Gp(s) dado: $\frac{10}{100s+2}$ decidimos dividirlo sobre dos, para simplificarlo y obtener: $\frac{5}{50s+1}$. También, graficamos esta función:

```
% Comenzamos definiendo la variable 's':
s = tf('s');
z = tf('z', 0.1);
% El problema nos presenta el siguiente Gp(s):
Gp_s = 10/(100*s+2);
% Aquí, procedemos a simplificarlo, dividiendo entre dos, por lo que:
Gp_s = (10/2)/((100/2)*s+(2/2))
% Y graficamos Gp(s):
figure();
step(Gp_s);
title('Gp(s)');
grid on
```

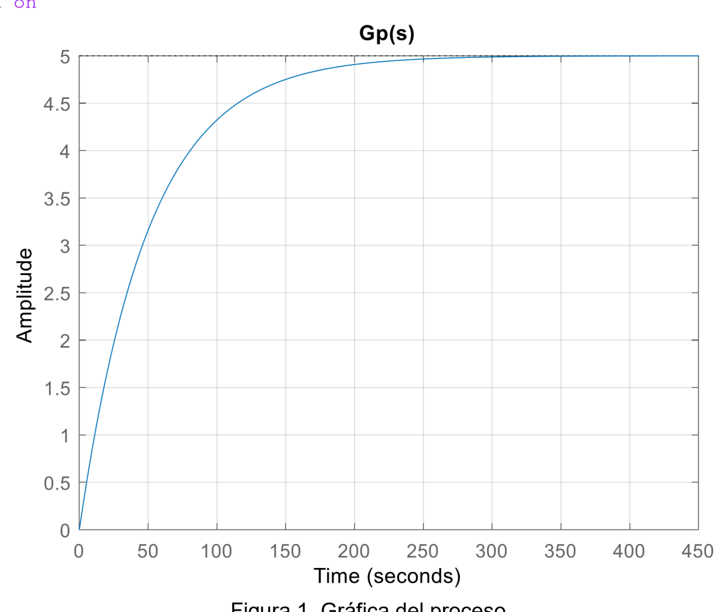


Figura 1. Gráfica del proceso.

Se pide considerar un $Ess = 0$ ante un escalón $R = 10$. Así mismo, se pide considerar un t dos veces más rápido.

A) Diseñe un controlador continúo usando el método de asignación de dinámica que garantice lo que se pide en el problema de control. Justifique los parámetros propuestos.

Primero, obtenemos, del proceso, los siguientes datos:

$R = 10.$

$K = 5.$

$t = 50.$

Por lo tanto, se escogen los siguientes valores para K^* y t^*, considerando el $Ess = 0$:

$K^* = 1$ y $t^* = \frac{t}{2} = 25$. Con lo que podemos sustituir en la fórmula: ${H_{YR(S)}}^* = K^*/(t^*s + 1)$ y, posteriormente, aplicar el método de asignación dinámica y obtener:

$$Gc(s) = \left(\frac{1}{Gp(s)}\right) * \left(\frac{{H_{YR(S)}}^*}{1 - {H_{YR(S)}}^*}\right)$$

El código en MATLAB para esta sección corresponde a:

```
%A) Diseñar un controlador continúo usando el método de asignación dinámica

    % Primero, podemos obtener del proceso los siguientes datos:
    R = 10;
    k = 5;
    t = 50;
    % Por lo tanto, se escogen los siguientes valores para k y t estrella,
    % para un Ess=0 y un T=2x.
    k_estrella = 1;
    t_estrella = t/2;
    Hyrs_estrella = k_estrella/(t_estrella*s+1)
    % Aplicando el método de asignación dinámica: (minreal simplifica)
    Gc_s = minreal((1/Gp_s) * (Hyrs_estrella/(1-Hyrs_estrella)))
    figure();
    step(feedback(Gc_s*Gp_s,1), Hyrs_estrella);
    title('Sistema de control en lazo cerrado');
    grid on;
```

Consecutivamente, se obtiene la siguiente gráfica:

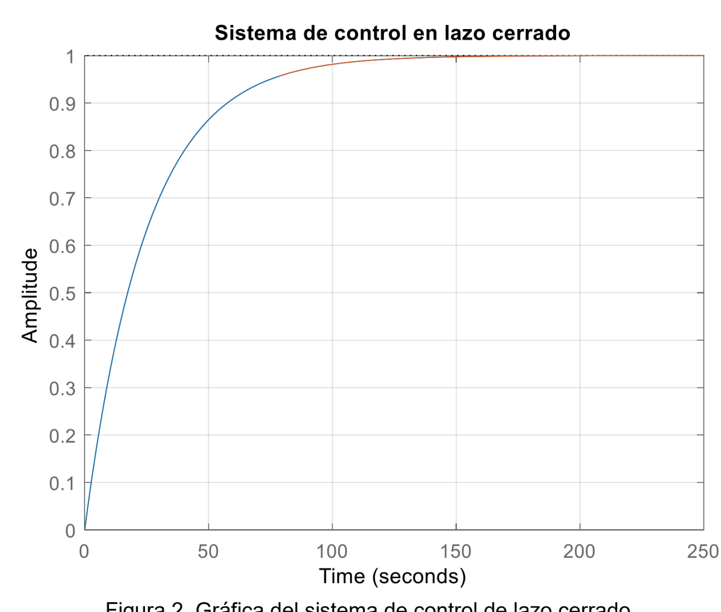


Figura 2. Gráfica del sistema de control de lazo cerrado.

B) Demuestre matemáticamente que, efectivamente, el error en estado estacionario del diseño en continuo es cero ante una señal de referencia del tipo escalón de magnitud 10.

Para verificar que el error en estado estacionario sea igual a cero, ante señales de tipo escalón (recordemos, con valor de 10), comenzamos por obtener el $E(s)$:

$$E(s) = R/(1 + Gc(s) * Gp(s))$$

Posteriormente, procedemos a graficar un step, como se muestra en el siguiente código, y obtenemos (mediante la gráfica) que el e_{ss} tiende a cero.

```
%B) Demostrar que el Ess sea cero:
    % Comenzamos por obtener el E(s):
    E_s = minreal((1/1+Gp_s*Gc_s)*(R))
    % Finalmente, obtenemos que el Ess=0 mediante la gráfica:
    figure();
    step(feedback(R,(1+Gp_s*Gc_s)));
    title('Error en estado estacionario Ess');
    grid on;
```

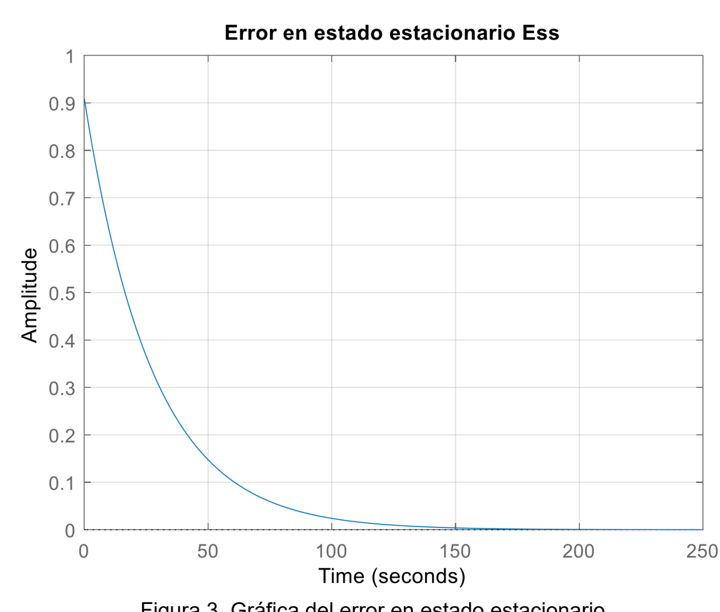



Figura 3. Gráfica del error en estado estacionario.

C) Utilizando discretización, obtener un controlador digital. Proponiendo un periodo de muestreo adecuado. Obtener la función de transferencia simplificada del controlador en su mínima expresión. Justificar la técnica de discretización.

Utilizando el método de Tustin, se obtiene la aproximación del equivalente discreto, para el controlador. El motivo de utilizar Tustin es porque resulta con mayor precisión en el rango de estabilidad, al momento de discretizar. En este caso, el periodo de muestre será de $t_{tustin} = 0.5$, ya que debe de ser entre 50 y 100 veces menor para poder garantizar un comportamiento similar al proceso en el dominio complejo. Todo esto, se logra mediante el siguiente código en MATLAB:

```
%C) Utilizando discretización, obtener un controlador digital.
%   Proponiendo un periodo de muestreo adecuado.
%   Obtener la función de transferencia simplificada del controlador en su
%   mínima expresión.
```



```
%   Justificar la técnica de discretización.
    % Se utiliza el método de Tustin. En donde el tiempo de muestreo es de
    % 50 a 100 veces menor al de estabilización del Proceso deseado.
    t_tustin = t_estrella/50;
    %Para el Gc(z), se evalúa Gc(s) en (2/t_tustin)*((z-1)/(z+1))
    Gc_z = c2d(Gc_s, t_tustin, 'Tustin')
```

D) Ahora, verificaremos que el diseño digital resuelve el problema de control planteado. Para esto, usando un retenedor de orden cero, obtenga el equivalente discreto del proceso considerando el período de muestreo previamente propuesto.

Con una implementación y lógica similar al inciso anterior, se utiliza el método de ZOH *(Zero Order Hold)* para obtener la aproximación del equivalente discreto, para el proceso. En este caso, el periodo de muestre será también de $t_{tustin} = 0.5$. Implementándolo en MATLAB, se puede obtener también la gráfica para el sistema de control en lazo cerrado:

```
%D) Usando un retenedor de orden 0, obtener el equivalente discreto:
    % Para aplicar el método ZOH:
    HGp_z = c2d(Gp_s, t_tustin, 'ZOH')
    figure();
    step(feedback((HGp_z*Gc_z), 1));
    title('Sistema de control de lazo cerrado en el dominio discreto');
    grid on;
```

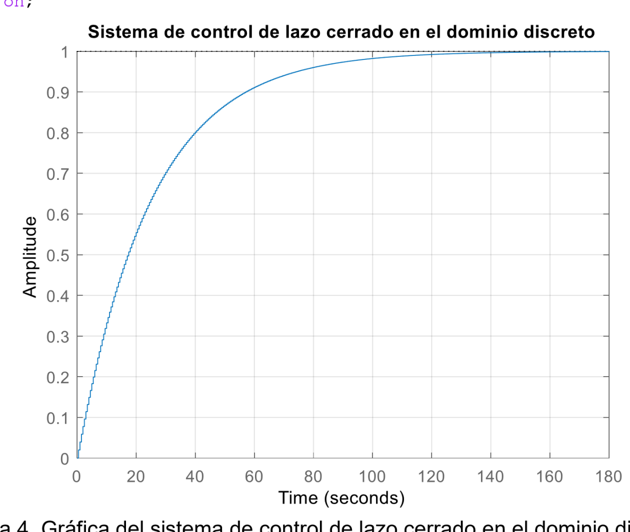


Figura 4. Gráfica del sistema de control de lazo cerrado en el dominio discreto.

E) Usando el controlador digital previamente obtenido y el equivalente discreto, verifique que el diseño digital cumple con garantizar el inciso b del problema de control. ¿Qué podemos decir sobre el inciso a, se verifica? Comente.

Con una lógica similar al inciso B, analizamos el error, en el dominio discreto:

```
%E) Checar que el diseño discreto cumple con el inciso b (Ess=0 con R=10)
    %Obtenemos el error discreto E(z):
    E_z = minreal((1/(1+HGp_z*Gc_z))*(R))
    % Finalmente, obtenemos que el Ess=0 mediante la gráfica:
    figure();
    step(feedback(R,(1+HGp_z*Gc_z)));
    title('Error en estado estacionario Ess en el dominio discreto');
    grid on;
```

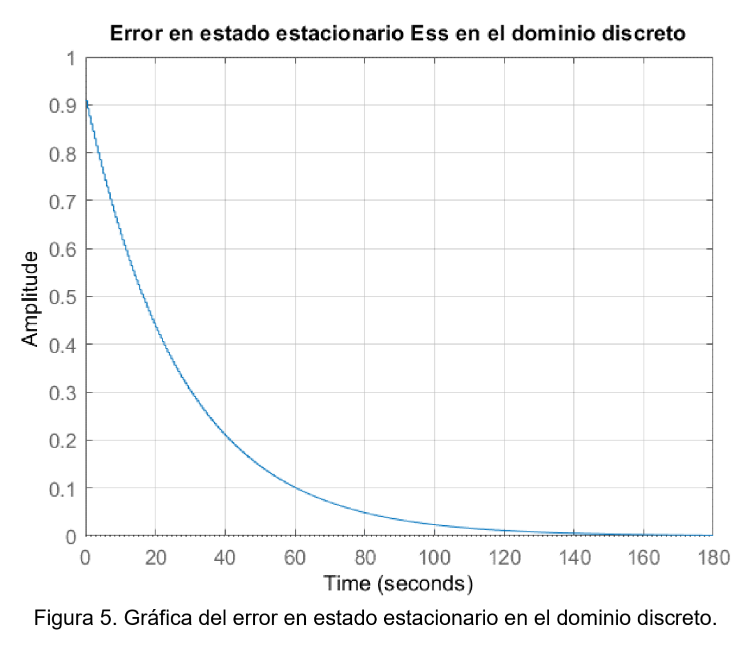


Figura 5. Gráfica del error en estado estacionario en el dominio discreto.

Posterior a ver la gráfica, podemos concluir que se cumple la condición del tiempo, ya que (a estimación visual) el tiempo de estabilización en la Figura 3 es de aproximadamente 100 segundos; en cambio, el tiempo presentado en la Figura 5 es de, aproximadamente, 40 segundos.

F) Ahora, seguiremos con la fase de implementación del controlador digital. Para esto, obtenga la ley de control digital, es decir, la ecuación de recurrencia del controlador que sería programado.

$$Gc(z) = \frac{0.402z - 0.398}{z - 1}$$

$$\frac{U(Z)}{E(Z)} = \frac{0.402 - 0.398z^{-1}}{1 - z^{-1}}$$

$$U(Z)(1 - z^{-1}) = (0.402 - 0.398z^{-1})E(Z)$$

$$u(k) = u(k - 1) + 0.402e(k) - 0.398e(k - 1)$$

G) Escriba el seudocódigo que sería utilizado para programar la ley de control digital.

```
R = 10;
Uant = 0;
Eant = 0;
While(1)
     Leer Y = leer A/D;
     Eact = R - Y;
     Uact = Uant + 0.402*Eact - 0.398*Eant;
     Eant = Eact;
     Uant = Uact;
     Mandar Uact en D/A;
End
```

H) Finalmente, se pide dibujar el diagrama a bloques final del sistema de control digital en lazo cerrado que sería implementado (incluya los convertidores, proceso, etc.)

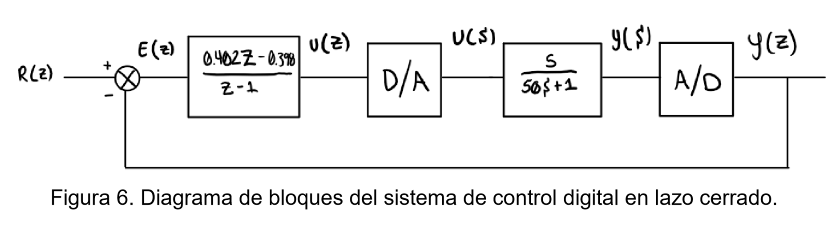


Figura 6. Diagrama de bloques del sistema de control digital en lazo cerrado.

Finalmente, podemos realizar una simulación en MATLAB del resultado.

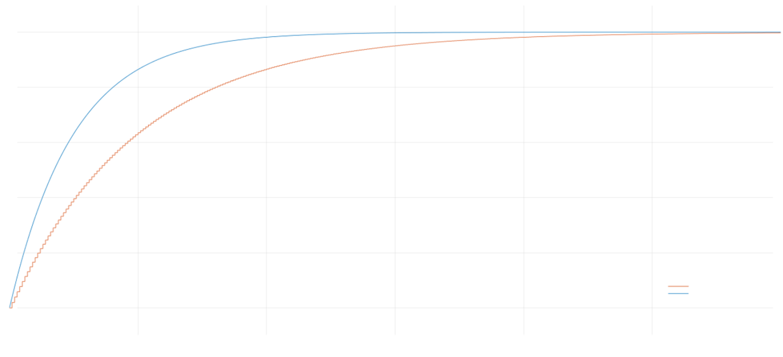


Figura 7. Señal de respuesta del sistema en dominio de z y s.

Como se puede apreciar en la imagen anterior, resultan señales que se estabilizan a los 5V (señal de referencia).

Nota: Como no hemos visto diagramas de bloques discretos en clase, no estamos seguros de esta comparativa.