

Práctica 7: Diseño de controladores discretos usando lugar geométrico de las raíces-Simulación

7.1. Objetivos

7.1.1. Objetivo general

Diseñar controladores discretos usando técnicas del lugar geométrico de las raíces en sisotool de MATLAB[®] para el análisis del desempeño del sistema en lazo cerrado con cada controlador diseñado.

7.1.2. Objetivos específicos

- Obtener los requerimientos de diseño de los controladores a partir de las gráficas dadas para el posterior diseño del controlador.
- Diseñar dos controladores en base a los requerimientos planteados en cada caso usando sisotool.
- Simular el sistema en lazo cerrado usando cada uno de los controladores diseñados en Simulink.
- Verificar el cumplimiento de los requerimientos de diseño a partir de las gráficas obtenidas.

7.2. Introducción

Como ya se estudió, el límite de estabilidad de los sistemas discretos queda definido por el círculo unitario en el plano z . La ecuación característica utilizada para bosquejar el lugar geométrico de las raíces (LGR) para sistemas discretos posee la misma forma que la utilizada para sistemas continuos. El lugar geométrico de las raíces es una gráfica que presenta las posibles ubicaciones de los polos de lazo cerrado del sistema dependiendo del valor que tome el controlador del mismo.

Sisotool es una herramienta de MATLAB[®] que permite obtener, entre otras cosas, el LGR de un sistema. Esta herramienta además permite modificar la función de transferencia del controlador y pre-filtro, para que el usuario pueda observar de manera dinámica cómo dichos cambios afectan al LGR, la ubicación de polos de lazo cerrado y la respectiva respuesta escalón del sistema. Además, sisotool permite a los usuarios especificar regiones y contornos en el plano z dependiendo de los índices de desempeño solicitados.

Para que un polo pertenezca al LGR debe cumplir los criterios de magnitud y fase, mismos que se muestran a continuación:

- $\angle F(z) = +180^\circ(2k + 1) \quad k = 0, 1, 2, \dots$
- $|F(z)| = 1$

Otras ecuaciones que debe tener en consideración para el desarrollo de esta práctica se muestran a continuación, donde ω_s representa la cantidad de muestras por segundo y ω_d representa la cantidad de oscilaciones por segundo:

$$\frac{\omega_s}{\omega_d} = \frac{2\pi}{\angle z} \tag{7.1}$$

$$\angle z = T\omega_n\sqrt{1 - \zeta^2} = T\omega_d \tag{7.2}$$

Finalmente, se debe recordar que el controlador PID discreto tiene la siguiente función de transferencia:

$$\frac{M(z)}{E(z)} = G_D(z) = K_{pD} + \frac{K_{iD}}{1 - z^{-1}} + K_{dD}(1 - z^{-1}) \tag{7.3}$$

$$G_D(z) = \frac{(K_{pD} + K_{iD} + K_{dD})z^2 - (K_{pD} + 2K_{dD})z + K_{dD}}{(z - 1)z} \tag{7.4}$$

7.3. Especificaciones del experimento

Se desea diseñar dos controladores discretos de la familia de controladores PID para una planta cuya función de transferencia se muestra a continuación:

$$G(s) = \frac{10}{(s + 10)}$$

Se conoce que la arquitectura de lazo de control a utilizar es la mostrada en la figura 7.1. El tiempo de muestreo a utilizar es de $T = 0.1[s]$.

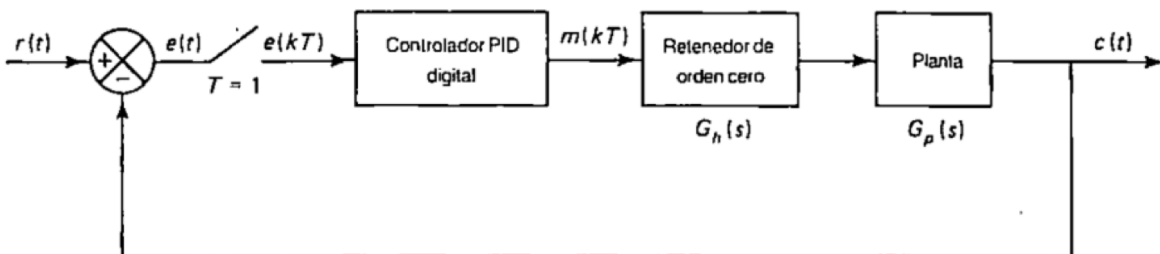


Figura 7.1: Arquitectura del lazo de control

Las respuestas de lazo cerrado deseadas en cada caso se muestran a continuación:

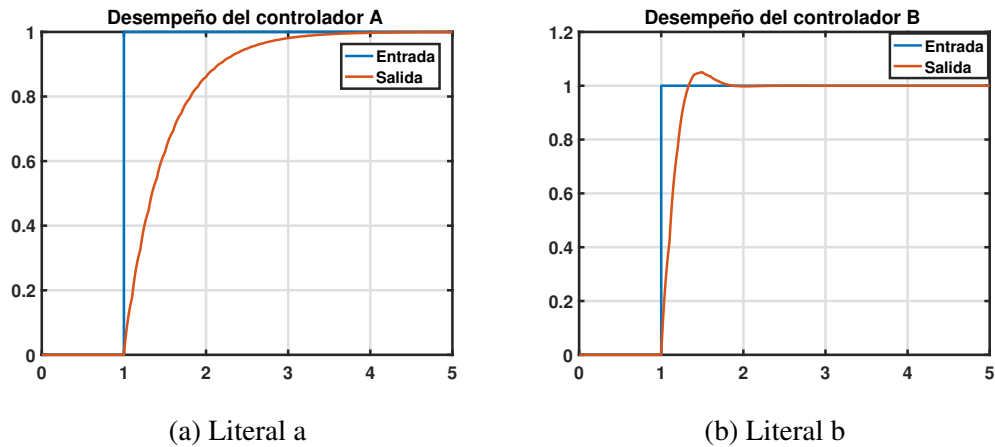


Figura 7.2: Respuestas escalón de lazo cerrado esperadas

7.4. Procedimiento

Cree un script en MATLAB[®] y coloque su nombre al inicio del mismo. Recuerde utilizar el símbolo % para comentar las líneas de código que considere necesario y utilizar %% para separar secciones de código. Resuelva en dicho script los ejercicios que se presentan a continuación y presente sus resultados en el *formato* proporcionado en el **blog** del curso.

1. Ingrese la función de transferencia de la planta y obtenga su equivalente discreto considerado el retenedor de orden cero y el tiempo de muestreo solicitado.

Para cada literal:

2. A partir de las gráfica dada obtenga los índices de desempeño necesarios para el diseño de del controlador.
3. Use el comando sisotool, cierre las ventanas que no sean necesarias y marque los requerimientos de diseño del controlador. Realice captura de pantalla del LGR y de la respuesta escalón inicial ($C(z) = 1$).

Nota: Recuerde usar integrador para aumentar el tipo del sistema y cumplir con el requerimiento de error de estado estacionario en caso de ser necesario.

4. Añada los ceros y polos que sean necesarios para lograr que exista LGR en la intersección de los contornos graficados, considerando la estructura de la familia de controladores PID.
5. Ubique los polos de lazo cerrado en las intersecciones de los contornos modificando la ganancia del controlador y obtenga la función de transferencia del controlador. Verifique el cumplimiento de los requerimientos de diseño en la respuesta escalón mostrada. Registre dicha función en su reporte y obtenga los parámetros K_p , K_i y K_d de su controlador discreto. Recuerde que todos los parámetros deben ser positivos.

6. Realice captura de pantalla del LGR final y de su respectiva respuesta escalón mostrando las características que evidencien el cumplimiento de los requerimientos de diseño. Registre dichas gráficas en su formato.
7. Arme un diagrama en Simulink similar al presentado en la figura 7.1. Recuerde configurar el bloque ZOH y el bloque del controlador (*Discrete Transfer Fcn*) con el tiempo de muestreo solicitado en el ejercicio.
8. Añada un Mux y configure el Scope o los bloques To Workspace que sean necesarios para que al simular el modelo se exporten los datos de la entrada, salida y señal de control del sistema al workspace de MATLAB[®]. Registre el diagrama en su reporte.
9. Realice la simulación considerando una entrada de prueba escalón unitario. Configure el tiempo de simulación de acuerdo al tiempo de estabilización solicitado en cada caso. Recuerde revisar que el tiempo de muestreo máximo de Simulink sea menor al tiempo de muestreo usado en el controlador.
10. Presente en su reporte las siguientes gráficas; recuerde incluir cuadrículas, títulos, nombres de los ejes y leyendas. Incluya marquillas en las gráficas que le faciliten el cálculo de los índices de desempeño obtenidos.
 - Gráfica de la entrada y salida del sistema superpuestas versus tiempo con el controlador del literal a.
 - Gráfica de la entrada y salida del sistema superpuestas versus tiempo con el controlador del literal b.
 - Subfigura mostrando la señal de control de cada controlador versus tiempo.
11. Complete la tabla presentada en el formato.

7.5. Conclusiones y Recomendaciones

Práctica 8: Diseño controlador discreto usando lugar geométrico de las raíces-Planta

8.1. Objetivos

8.1.1. Objetivo general

Diseñar controladores discretos usando técnicas del lugar geométrico de las raíces en sisotool de MATLAB[®] para la evaluación del desempeño de estos en plantas didácticas.

8.1.2. Objetivos específicos

- Diseñar dos controladores en base a los requerimientos planteados en cada caso usando sisotool.
- Probar cada uno de los controladores diseñados con el respectivo sistema en estudio.
- Verificar el cumplimiento de los requerimientos de diseño de las gráficas obtenidas.

8.2. Especificaciones

- Función de transferencia de la planta: $G(s) = \frac{0.018538}{(s + 0.2763)(s + 0.09827)(s + 0.004068)}$
- Puntos de operación: (3.175V, 30cm)
- Rango de operación válido: $\pm 2cm$
- Valor máximo a la entrada del proceso: 10V
- Tiempo de muestreo a utilizar: 2s

8.3. Procedimiento

1. Descargue el modelo de Simulink adjunto; este representa una planta didáctica de control de nivel disponible en el laboratorio de Sistemas de Control.
2. Cree un script y coloque su nombre y paralelo al inicio del mismo. Ingrese los datos de la práctica.

3. Obtenga la función de transferencia discreta equivalente del sistema con el tiempo de muestreo dado. Recuerde que esta función de transferencia es válida alrededor de los puntos y rango de operación dados.
4. Diseñe un primer controlador PID (o cualquiera de sus combinaciones) de manera que se cumplan las siguientes especificaciones ante una entrada escalón unitario:
 - Error de estado estacionario igual a cero.
 - Tiempo de estabilización alrededor de 120 segundos.
 - Sobrenivel porcentual alrededor del 15 %.

Realice capturas de pantalla del LGR inicial, LGR final mostrando regiones asociadas a los requerimientos de diseño, respuesta escalón inicial y final mostrando características y función de transferencia del controlador.

5. Exporte dicho controlador al espacio de trabajo como **C1** y obtenga los parámetros K_p , K_i , y K_d .
6. Diseñe un segundo controlador PID (o cualquiera de sus combinaciones) de manera que se cumplan las siguientes especificaciones ante una entrada escalón unitario:
 - Error de estado estacionario igual a cero.
 - Tiempo de estabilización alrededor de 60 segundos.
 - Sobrenivel porcentual igual a cero.

Realice capturas de pantalla del LGR final mostrando regiones asociadas a los requerimientos de diseño, respuesta escalón final mostrando características y función de transferencia del controlador.

7. Exporte dicho controlador al espacio de trabajo como **C2** y obtenga los parámetros K_p , K_i , y K_d .
8. Realice la simulación de cada controlador en Simulink para cambios de referencia positivos y negativos. En cada caso exporte los resultados al espacio de trabajo de MATLAB[®] y realice una figura con la entrada y salida del sistema superpuestas y una figura con la señal de control del sistema. Obtenga los índices de desempeño de las gráficas y llene la tabla al final del formato de la práctica.

Nota: Recuerde sumar a la salida del controlador el punto de operación y darle suficiente tiempo al sistema para llegar a su punto de operación antes de realizar el cambio de referencia.

8.4. Conclusiones y Recomendaciones

Práctica 9: Representación de un sistema en variables de estado

9.1. Objetivos

9.1.1. Objetivo general

Representar sistemas en variables de estado mediante comandos de MATLAB[®] y simulaciones en Simulink para el estudio de los diferentes comandos y formas canónicas.

9.1.2. Objetivos específicos

- Representar un sistema continuo en variables de estado a partir de las ecuaciones del mismo.
- Obtener la representación en variables de estado de un sistema a partir de su función de transferencia.
- Obtener la función de transferencia de un sistema a partir de su representación en variables de estado.

9.2. Ejercicio

Se desea representar las ecuaciones linealizadas del sistema mostrado en la figura 9.1 en variables de estado. Las ecuaciones lineales que gobiernan dicho sistema son 9.1 y 9.2. En la tabla 9.1 se presentan los datos conocidos del sistema y en la tabla 9.2, se presentan los valores de los parámetros de las ecuaciones lineales. Considere T_i y V_{AC} como entradas, T_1 y T_o como salidas.

$$K_1 t_i(t) + K_2 t_1(t) + K_3 v_{ac}(t) + K_4 \frac{dt_1(t)}{dt} = 0 \quad (9.1)$$

$$K_5 t_1(t) + K_6 t_o(t) + K_7 \frac{dt_o(t)}{dt} = 0 \quad (9.2)$$

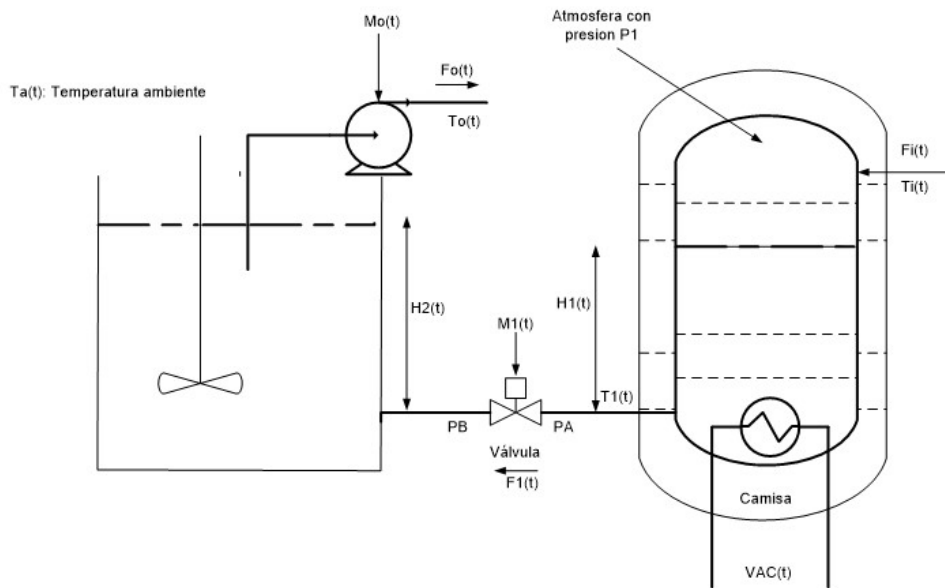


Figura 9.1: Diagrama del proceso

Parámetro	Valor
ρ	1000
c	4180
R_e	0.2
R_t	$1 * 10^{-3}$
r_1	1
r_2	2
F_{iop}	0.01
V_{acop}	200
H_{1cop}	2.058
H_{2op}	1.7359

Tabla 9.1: Datos conocidos del sistema

Parámetro	Valor
K_1	$F_{iop}\rho c$
K_2	$-K_1$
K_3	$2V_{acop}/R_e$
K_4	$-\pi H_{1op} cr_1^2 \rho$
K_5	K_1
K_6	$-1/R_t - F_{iop}c\rho$
K_7	$-\pi H_{2op} cr_2^2 \rho$

Tabla 9.2: Parámetros de las ecuaciones linealizadas

9.3. Procedimiento

Cree un script e incluya su nombre y paralelo al inicio del mismo. Comente cada línea de código usando el signo `%` y separe su script por secciones usando `%%` seguido de un espacio y encabezado de la sección.

En esta práctica se analizará la equivalencia entre representaciones en el dominio de la frecuencia (Laplace) y dominio del tiempo (espacio de estados) y entre las formas canónicas utilizadas en la representación de variables de estado.

1. A partir de las ecuaciones anteriores, determine e ingrese las matrices A, B, C y D de la representación en variables de estado del sistema en MATLAB[®]. Use el comando `ss` para este ítem y almacene el resultado como G_{ss} .
2. Encuentre la matriz de transferencia $G(s)$ del sistema a partir de la representación en variables de estado del literal anterior.
3. Compruebe que la representación en variables de estado G_{ss} sea equivalente a la matriz de transferencia $G(s)$. Use el comando `tf` o `zpk` sobre el resultado obtenido en el primer ítem.
4. Use el comando `ss` sobre la matriz de transferencia $G(s)$ ingresada en el segundo ítem. Almacene dicho resultado como G_{ss2} . Compare este resultado con respecto a G_{ss} .

En Simulink, realice lo siguiente:

5. Ingrese el bloque 'State-Space'. A partir de G_{ss} , realice los cambios necesarios en dicho bloque para **cada uno** de los siguientes escenarios de simulación.
 - Configure el bloque de manera que la salida sea únicamente la señal T_o .
 - Configure el bloque de manera que la salida sea únicamente la señal T_1 .
6. Configure las entradas del sistema de la siguiente manera:
 - La señal V_{ac} tiene una variación de 10 voltios a partir de $t = 5s$.
 - La señal T_i tiene una variación de 0.5 grados Celsius a partir de $t = 20000s$.
7. Realice las simulaciones respectivas y grafique las señales de entrada, de salida y variables de estado versus tiempo para cada caso.
8. En un nuevo modelo, configure el bloque 'State-Space', usando la representación en variables de estado obtenida a partir de la matriz de transferencia $G(s)$, es decir G_{ss2} , de manera que la salida del bloque corresponda a las variables de estado. Además se desea visualizar también las salidas del sistema T_1 y T_o .
9. Simule el sistema utilizando las mismas señales de entrada usadas previamente.
10. Grafique las señales de entrada, de salida y variables de estado versus tiempo.
11. Comente acerca de las similitudes y diferencias de G_{ss} y G_{ss2} . Use las gráficas obtenidas para fundamentar sus respuestas.

9.4. Conclusiones y Recomendaciones

Práctica 10: Controlabilidad, observabilidad y realimentación de estados

10.1. Objetivos

10.1.1. Objetivo general

Verificar la controlabilidad y observabilidad de un sistema representado en variables de estado y diseñar un controlador por realimentación de estados usando comandos de MATLAB® para mejorar el desempeño del sistema.

10.1.2. Objetivos específicos

- Obtener la matriz de controlabilidad a partir de un sistema descrito en variables de estado para la verificación de dicha propiedad.
- Obtener la matriz de observabilidad a partir de un sistema descrito en variables de estado para la verificación de dicha propiedad.
- Determinar la ubicación de los polos de acuerdo a los requerimientos planteados y diseñar el controlador usando MATLAB®.
- Simular el sistema en estudio con el controlador diseñado y verificar su desempeño.

10.2. Introducción

Al representar sistemas a través de variables de estados, es posible agregar un nuevo tipo de controlador a los ya revisados. En esta práctica se estudiará la realimentación de estados para reubicación de polos como estrategia de control.

Se conoce que un sistema puede ser representado a través de sus variables de estado x de la siguiente manera:

$$\dot{x} = Ax + Bu \quad (10.1)$$

$$y = Cx + Du \quad (10.2)$$

donde, \mathbf{u} y \mathbf{y} corresponden a la entrada y salida del proceso respectivamente. La señal \mathbf{u} , puede ser utilizada para regular la salida a través de las variables de estado x , por lo que esta suele ser

también llamada señal de control o **ley de control**. Además se conoce que la ecuación característica del sistema se la puede obtener como:

$$E.C. : \det(sI - A) = 0 \quad (10.3)$$

Como punto de inicio, se supondrá que la ley de control es $u = r - Kx$ donde r corresponde a la señal de referencia y K al vector o matriz de ganancias con las que se realimentarán las variables de estado x . Al reemplazar esta expresión en 10.1 y reagrupar los términos, se tendría que redefinir la matriz de estados como:

$$A' = A - BK \quad (10.4)$$

Por lo tanto la nueva ecuación característica del sistema quedaría expresada en función de los valores del vector de realimentación K . Sin embargo, esta ley de control no permite realizar seguimiento a la referencia o realizar control integral por lo que puede existir error de estado estacionario.

10.3. Procedimiento

Cree un script y documento de manera detallada el procedimiento seguido. Recuerde incluir su nombre y paralelo en el mismo. Para el **desarrollo** de esta práctica, se utilizará el sistema de la práctica anterior.

10.3.1. Controlabilidad y observabilidad

1. Obtenga el diagrama de flujo de señales del sistema G_{ss} y regístrelo en el reporte. Identifique claramente las variables de estado en su diagrama.
2. Obtenga la matriz de controlabilidad del sistema y determine si el mismo es controlable. Utilice el comando **ctrb** y el comando **rank** para este fin.
3. Obtenga la representación canónica controlable de la relación entre T_o y T_i usando comandos de MATLAB[®]. Regístrela en su reporte.
4. Obtenga la matriz de observabilidad del sistema y determine si el mismo es observable. Utilice el comando **obsv** y el comando **rank** para este fin.
5. Obtenga la representación canónica observable de la relación entre T_o y T_i usando comandos de MATLAB[®]. Regístrela en su reporte.

10.3.2. Realimentación de estados

Se desea que la respuesta en el tiempo del sistema sea diferente a la actual por lo que se requiere reubicar los polos de este de acuerdo a ciertos requerimientos.

6. Realice una prueba escalón unitario y registre los índices de desempeño de cada combinación entrada-salida.
7. Para las siguientes condiciones de diseño, encuentre la ubicación de los polos correspondientes.

Controlador a:

- Tiempo de estabilización alrededor de 500 segundos

- Sobrenivel porcentual alrededor del 15%

Controlador b:

- Tiempo de estabilización alrededor de 500 segundos
 - Sobrenivel porcentual igual a cero
8. Use el comando **place** en conjunto con los polos calculados en el ítem anterior para hallar las matrices de realimentación asociadas a cada controlador.
 9. Realice una simulación en cada caso, exporte los datos y grafique las señales de entrada, salida y control.
 10. Compruebe en cada caso si se cumplieron las condiciones de diseño solicitadas. Registre los índices de desempeño de cada combinación entrada-salida, para cada controlador.
 11. Calcule el error de estado estacionario en cada caso.
 12. Obtenga la nueva representación del sistema en variables de estado con cada controlador usando MATLAB[®]. Recuerde que la ley de control usada corresponde a $u = r - Kx$.
 13. Obtenga la respuesta escalón del sistema para cada caso usando el comando **step**. Verifique los resultados obtenidos en el ítem 10.

10.4. Conclusiones y Recomendaciones

Práctica 11: Control integral y observador de estados

11.1. Objetivos

11.1.1. Objetivo general

Diseñar un controlador por realimentación de estados con control integral usando comandos y simulaciones de MATLAB[®] para mejorar el desempeño de un sistema.

11.1.2. Objetivos específicos

- Realizar pruebas de controlabilidad y observabilidad del sistema usando MATLAB[®] previo al diseño del controlador.
- Diseñar un observador de estados que permita tener acceso a las variables de estado del sistema en estudio.
- Determinar la ubicación deseada de los polos de lazo cerrado de acuerdo a los requerimientos planteados.
- Ajustar la representación en variables de estado del sistema para realizar seguimiento a la referencia.
- Diseñar el controlador por realimentación de estados del sistema de manera que se ubiquen los polos en las posiciones deseadas.
- Simular el sistema en estudio con el controlador y observador diseñados.
- Verificar el desempeño del sistema en lazo cerrado con el controlador y observador diseñados.

11.2. Introducción

En la práctica anterior se revisó el procedimiento para diseñar un controlador para un sistema representado en variables de estados a través de la reubicación de sus polos. Sin embargo, la ley de control utilizada no permitía realizar seguimiento a la referencia o realizar control integral por lo que se revisará una nueva ley de control.

Suponga ahora que la ley de control es $u = K_e \int e \cdot dt - Kx$ donde e corresponde a la señal de error; es decir, $e = r - Cx$. Además, se debe añadir una variable de estado adicional $x_n = \int e \cdot dt$ por lo que se deben replantear las matrices usadas en el modelo de espacio de estados.

$$\begin{bmatrix} \dot{x} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_n \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \quad (11.1)$$

$$y = [C \ 0] \begin{bmatrix} x \\ x_n \end{bmatrix} \quad (11.2)$$

Al reescribir la ley de control como $u = K_e x_n - Kx$, reemplazarla en la expresión 11.1 y reagrupar los términos, la matriz de estados del sistema queda definida como:

$$A' = \begin{bmatrix} (A - BK) & BK_e \\ -C & 0 \end{bmatrix} \quad (11.3)$$

Se debe hallar los valores de la matriz K y K_e , de manera que los **polos de lazo cerrado** del sistema se uboquen en las posiciones deseadas. Se utilizará el comando **place** tal como se realizó en la práctica anterior; considere el nuevo orden del sistema al momento de crear el vector de polos deseados.

Nota: La matriz de entradas también se ve modificada al reemplazar la ley de control en la expresión 11.1. La ecuación característica del sistema se obtiene a partir de A' y para esta segunda ley de control estudiada, dicha matriz está definida por las matrices A , B y C originales del sistema, del vector de realimentación de estados K y la ganancia integral K_e .

En procesos reales, las variables de estado no siempre representan variables o señales físicas del sistema. Para realizar control por realimentación de estados bajo dicho escenario, se requiere diseñar un observador de estados que permita estimar los valores de estos a partir de las mediciones de entrada y salida del sistema. El observador está representado por las siguientes ecuaciones:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (11.4)$$

$$\hat{y} = C\hat{x} \quad (11.5)$$

Al reorganizar la ecuación se tendría que la matriz de estados del observador es $A' = A - LC$, donde L se diseña de manera que el observador tenga una dinámica más rápida que la del sistema en lazo cerrado. Las entradas del observador son las señales de entrada del proceso u y las señales de salida del proceso y . Para diseñar L use la siguiente sintaxis, donde po representa el vector que contiene los polos del observador.

$$L^T = \text{place}(A^T, C^T, po)$$

11.3. Procedimiento

Cree un script y documente el código necesario para desarrollar lo solicitado a continuación. Recuerde incluir su nombre al inicio del mismo y utilizar el símbolo % para comentar el código.

Se desea diseñar un controlador por realimentación de estados para el siguiente sistema:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -40 & -13 \end{bmatrix} \quad (11.6)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (11.7)$$

$$C = [6 \ 2 \ 0] \quad (11.8)$$

$$D = 0 \quad (11.9)$$

Como requerimientos de diseño considere, ante una entrada escalón:

- Sobrenivel porcentual alrededor del 10 %.
- Tiempo de estabilización alrededor de 2 segundos.
- Error de estado estacionario igual a cero.

1. Ingrese el sistema en MATLAB[®]. Compruebe que el sistema sea controlable y observable.
2. Determine el valor de los polos de lazo cerrado deseados de acuerdo a los requerimientos de diseño dados.
3. Para dar seguimiento a la referencia y lograr cumplir el requerimiento de error de estado estacionario igual a cero es necesario redefinir las variables de estado a utilizar. Haga los cambios necesarios para incluir la nueva variable de estado en la representación del sistema.
4. Use el comando **place** para ubicar los polos del sistema en las posiciones deseadas. En caso de que el orden de su sistema sea superior a la cantidad de polos deseados considere eliminar el o los ceros de la función de lazo cerrado o utilizar criterio de dominancia para la selección de la ubicación de los otros polos.
5. Pruebe el desempeño del controlador en Simulink. Exporte los datos al Workspace, grafique la señal de entrada y salida del sistema y verifique el cumplimiento de los requerimientos de diseño dados.
6. Determine el error de estado estacionario ante una entrada escalón unitario bajo estas condiciones.

Asuma que no se tiene acceso a las variables de estado del sistema. Diseñe el observador de estados y repita la simulación realimentando los estados estimados del observador.

11.4. Conclusiones y Recomendaciones