

Diseño en Espacio de Estados

Control Digital Avanzado

Camilo Andres Vera Ruiz

caverar@unal.edu.co

1. Características de la Planta

Se trabaja con la misma planta utilizada en la prácticas anteriores, sin embargo dado que se pretende hacer control de posición, se añade un integrador, y un factor de conversión para convertir la velocidad que está en rpm, en posición en grados, teniendo como resultado la siguiente función de transferencia:

$$G_p(s) = \frac{55.99}{s + 33.95} \cdot \frac{360}{s}$$

Para hacer control en variables de estado, se seleccionan los estados de: velocidad en rpm(x_1) y posición en grados(x_2), luego se separa el sistema en dos bloques que permitan distinguir las señales de entrada y los dos estados, en la figura 1 se muestra el diagrama de bloques correspondiente.

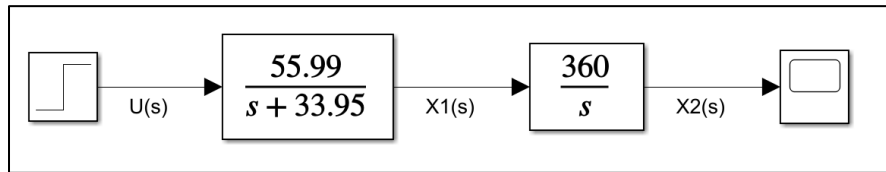


Figura 1: Diagrama de bloques del sistema.

A partir de la expresión de función de transferencia de cada bloque, se puede plantear una ecuación que relacione la derivada de cada estado, con respecto al mismo estado y la entrada de cada bloque, a continuación se muestran dichas expresiones:

$$sX_1(s) = -33.95X_1(s) + 55.99U(s)$$

$$sX_2(s) = 360X_1(s)$$

A partir de las expresiones anteriores y teniendo en cuenta que la salida del sistema es igual al estado X_2 , se construye la siguiente representación en variables de estado:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -33.95 & 0 \\ 360 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 55.99 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u(t)$$

Adicionalmente se tiene en cuenta que el periodo de muestreo sigue siendo el mismo, es decir 5ms, y dado que las técnicas de diseño en variables de estado, se pueden aplicar directamente en tiempo discreto, se realiza la discretización del sistema mediante la función de MATLAB c2d, usando la discretización ZOH, dando como resultado el siguiente sistema:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} 0.8439 & 0 \\ 1.656 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.2575 \\ 0.2383 \end{bmatrix} u(k)$$

$$y(k) = [0 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + [0]u(k)$$

2. Modelo Interno y Full State Feedback

Los requerimientos del diseño, incluyen un sobre pico menor al 25% y cero error de estado estacionario para la rampa, esto implica que es necesario aprovechar el principio del modelo interno para incluir dos estados extra que permitan incluir un doble integrador, y realimentar el error, para ello se plantea el diagrama de bloques de la figura 2

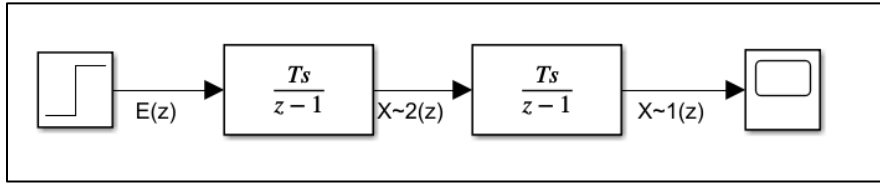


Figura 2: Bloques del modelo interno.

Este bloque se puede representar en variables de estado, de la siguiente forma:

$$\begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} = \begin{bmatrix} 1 & Ts \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ Ts \end{bmatrix} u(k)$$

Ambos sistemas se pueden juntar en una topología de realimentación con dos vectores de ganancias, uno para el modelo interno y otro para los estados, en la figura 3 se muestra el esquema de realimentación.

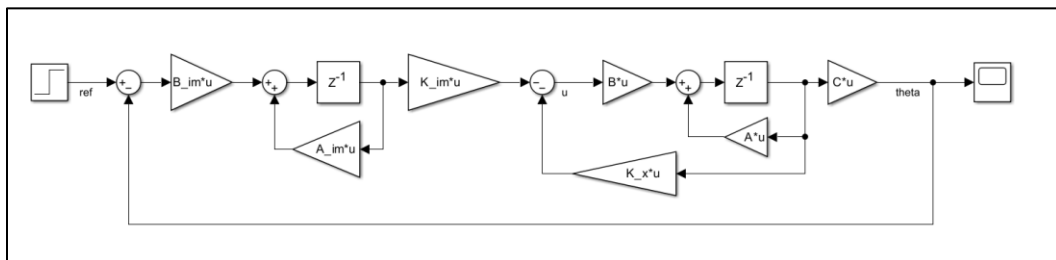


Figura 3: Realimentación de estados con modelo interno.

Finalmente se tendría una representación equivalente para el lazo abierto de este sistema, con la siguiente expresión:

$$\begin{bmatrix} \dot{x}_1(k) \\ \dot{x}_2(k) \\ \dot{\tilde{x}}_1(k) \\ \dot{\tilde{x}}_2(k) \end{bmatrix} = \underbrace{\begin{bmatrix} 0.8439 & 0 & 0 & 0 \\ 1.6555 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.005 \\ 0 & -0.005 & 0 & 1 \end{bmatrix}}_{A_{au}} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix}}_{x_{au}} + \underbrace{\begin{bmatrix} 0.2575 \\ 0.2383 \\ 0 \\ 0 \end{bmatrix}}_{b_{au}} u(k) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.005 \end{bmatrix}}_{b_r} r(k)$$

$$u(k) = -\underbrace{[k_x \quad k_{im}]}_{c_{au}} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix}}_{x_{au}}, \quad y(k) = \underbrace{[0 \quad 1 \quad 0 \quad 0]}_{c_{au}} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \\ \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix}}_{x_{au}}$$

Sobre este esquema, se aplican dos técnicas para calcular las ganancias (vectores fila) k_x y k_{im} que se describen a continuación.

3. Diseño por ubicación de polos

Mediante la función de MATLAB “place” que actúa de forma similar a la fórmula de Ackermann, es posible calcular las ganancias adecuadas para ubicar los polos o valores propios del sistema en cualquier lugar, aunque la documentación de la función recomienda ubicar los polos en lugares diferentes.

Con esto en mente, se selecciona de forma arbitraria en el semiplano negativo de s , un par de polos dominantes complejos conjugados, y dos polos reales de menor valor, luego se convierten a tiempo discreto mediante la fórmula $P_z = e^{P_s T_s}$ y se simula la respuesta al paso, repitiendo el proceso varias veces con el fin de realizar un ajuste manual de los polos complejos conjugados, los cuales determinan en mayor medida la dinámica del sistema, buscando minimizar el tiempo de respuesta y obtener el sobre pico necesario.

A partir de los principios del diagrama de lugar de las raíces, se parte de la aproximación de que el componente imaginario de los polos dominantes, es el principal responsable del sobre pico, mientras que la componente real, determina la velocidad del sistema, por lo cual se ajustan estos valores por separado hasta lograr alcanzar los requerimientos, luego se realiza un ajuste fino, encontrando finalmente los siguientes valores propios en tiempo discreto, como los valores aparentemente más óptimos que se pueden encontrar con un ajuste manual.

- 0.9802+0.0039j
- 0.9802-0.0039j
- 0.8187
- 0.7985

Mediante MATLAB se calculan las ganancias requeridas para lograr estos valores propios:

$$k_x = [0.9317 \quad 0.1106], \quad k_{im} = [-1.2854 \quad -0.6885]$$

En la figura 4 se muestra la respuesta al paso del lazo cerrado de todo el sistema, donde se refleja cero error de estado estacionario y sobre pico del 20% además de un tiempo de respuesta muy lento que no supone un problema con respecto al periodo de muestreo del controlador. Así mismo en la figura 5 se muestra la respuesta a la rampa, con cero error de estado estacionario.

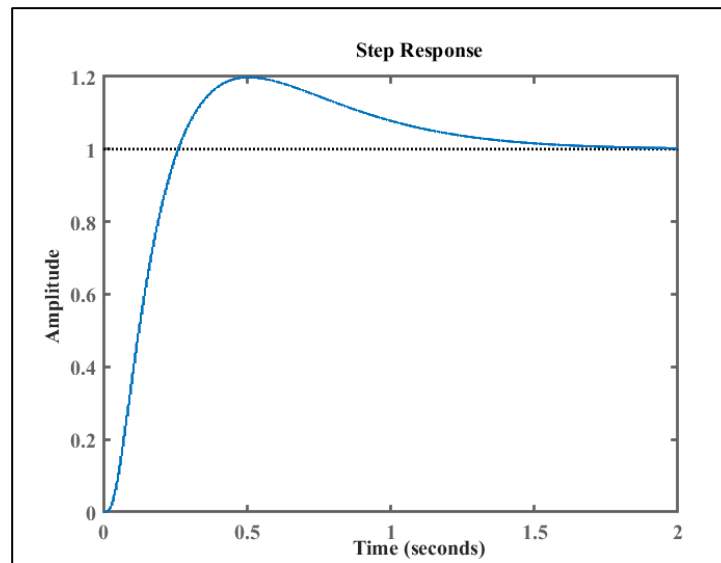


Figura 4: Respuesta al paso del lazo cerrado por ubicación de Polos.

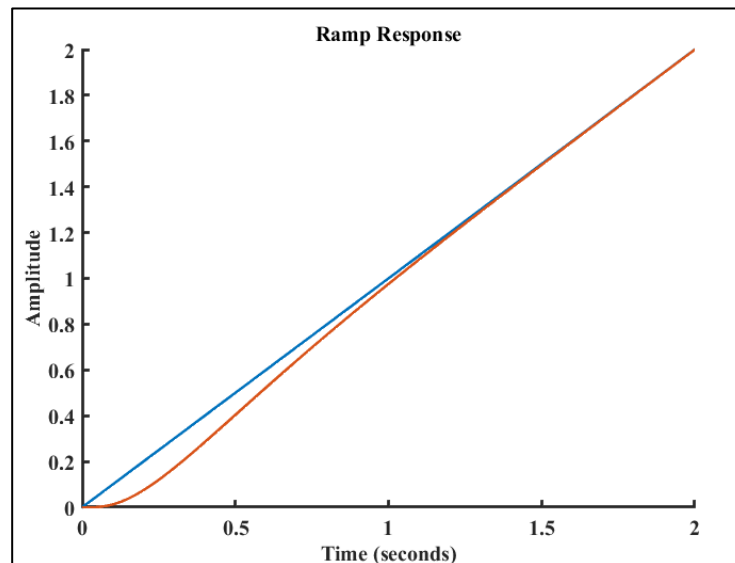


Figura 5: Respuesta a la rampa del lazo cerrado por ubicación de Polos.

4. LQR

La segundo método de diseño consistió en el calculo de las ganancias optimas para una función de costo cuadrático, con una matriz diagonal de costo Q que penaliza el mal desempeño de cada estado, y un escalar R que penaliza el esfuerzo de control, es decir su saturación.

La selección de los valores de la matriz Q y el escalar R, fue de naturaleza iterativa, modificando cada uno de los parámetros, hasta que un incremento o decrecimiento favorable, ya no lograra ninguna mejora en la respuesta del sistema.

Durante el proceso de implementación que se describe más adelante, se evidenciaron problemas al implementar este controlador al observar un comportamiento oscilatorio, a pesar de que la simulación no reflejaba este comportamiento, motivo por el cual fue necesario el incremento de la robustez del controlador mediante la reducción de la matriz Q, es decir la reducción de la penalización del mal desempeño de los estados. Esto se logró añadiendo un factor escalar de 0.2 a la matriz Q, con el inesperado resultado de no presentar cambios notables en el tiempo de respuesta, sobre pico, tiempo de establecimiento, etc. Las matrices de costo halladas son las siguientes:

$$Q = 0.2 \cdot \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 0.35 & 0 & 0 \\ 0 & 0 & 70 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, R = [5]$$

Mediante la función de MATLAB “dlqr” que permite ejecutar el algoritmo de optimización para calcular las ganancias de realimentación para un sistema discreto en variables de estado, se obtienen las siguientes ganancias:

$$k_x = [1.7314 \quad 0.2326], \quad k_{im} = [-2.7295 \quad -1.2244]$$

En la figura 6 se muestra la respuesta al paso del lazo cerrado de todo el sistema, donde se refleja cero error de estado estacionario y sobre pico algo mayor de 20%, además de un tiempo de respuesta muy lento que no supone un problema con respecto al periodo de muestreo del controlador. Así mismo en la figura 7 se muestra la respuesta a la rampa, con cero error de estado estacionario.

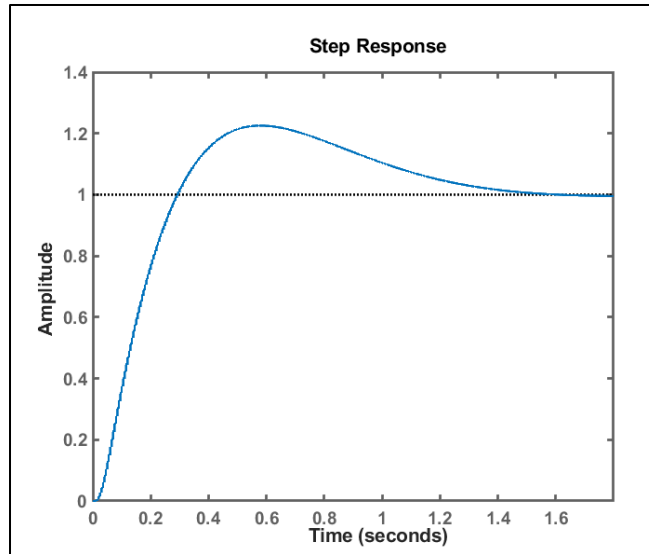


Figura 6: Respuesta al paso del LQR.

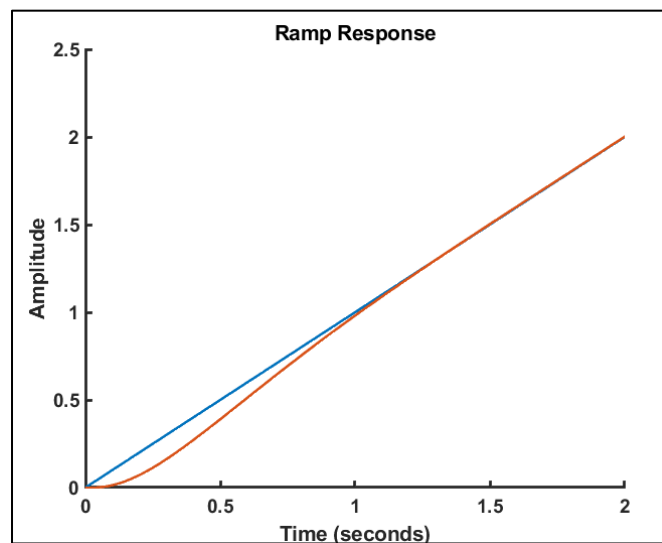


Figura 7: Respuesta a la rampa del LQR.

5. Simulaciones

Su utiliza Simulink para simular el comportamiento de todo el sistema, con el fin de verificar que la señal de control no presente saturación, dado que se esta trabajando en el dominio discreto, solo es necesario añadir un bloque de saturación en la señal de control, lo demás obedece al mismo diagrama de bloques de la figura 3. En la figura 8 se muestra este nuevo diagrama de bloques.

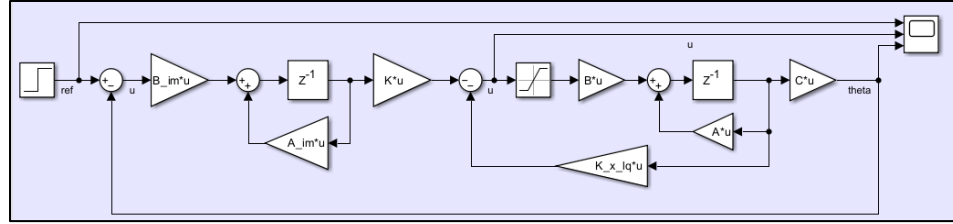
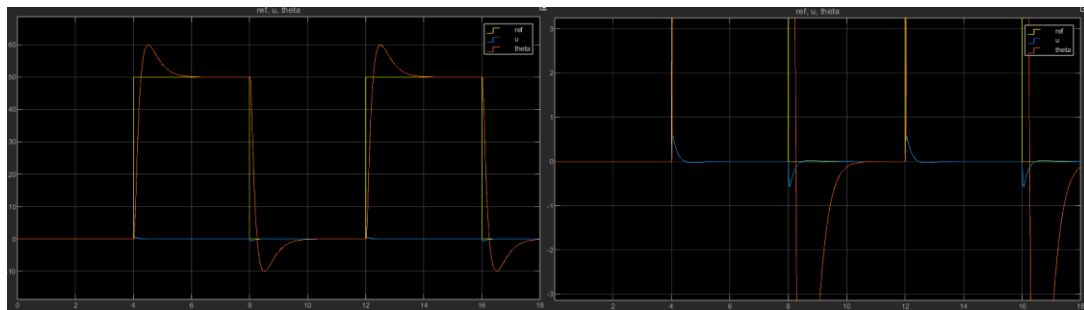


Figura 8: Diagrama de bloques de la simulación.

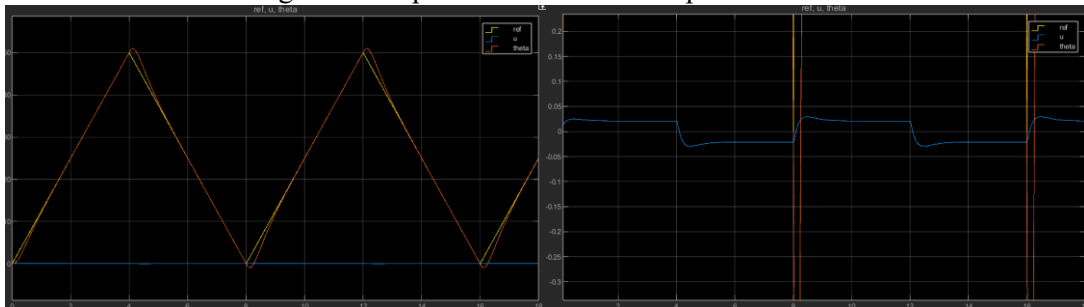
Se utiliza una referencia cuadrada y una rectangular, ambas de 0.125Hz variando de 0 a 50 grados, con el fin de reflejar un cambio de posición lo suficientemente grande como para notarlo audiblemente durante la etapa de implementación. De las figuras 9 a 12, se muestra las respuesta del controlador por ubicación de polos y el LQR.



a. Señal de Referencia y Salida.

b. Señal de Control.

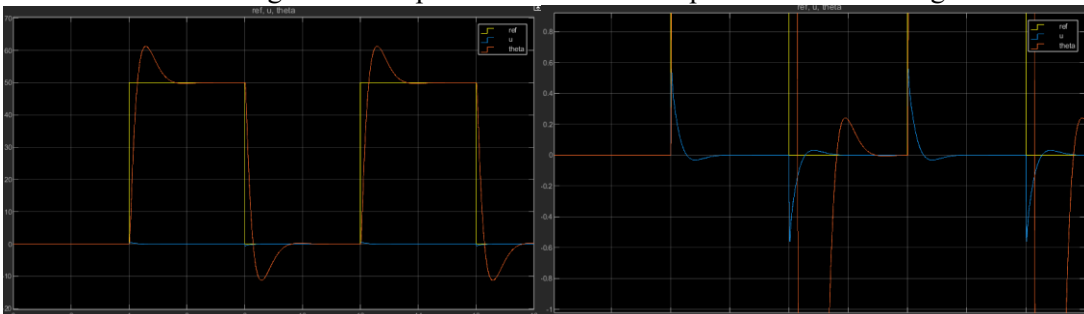
Figura 9: Respuesta de ubicación de polos a la señal cuadrada.



a. Señal de Referencia y Salida.

b. Señal de Control.

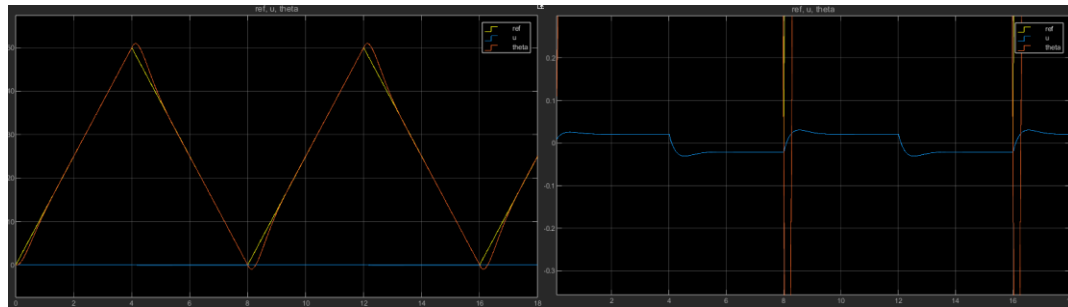
Figura 10: Respuesta de ubicación de polos a la señal triangular.



a. Señal de Referencia y Salida.

b. Señal de Control.

Figura 11: Respuesta de LQR a la señal cuadrada.



a. Señal de Referencia y Salida.

b. Señal de Control.

Figura 12: Respuesta de LQR a la señal triangular.

En las anteriores figuras, se evidencia que nunca se da la saturación, es decir la señal de control nunca alcanza el límite de 1, de igual forma se cumple con el error de estado estacionario cero tanto para el paso (señal cuadrada) como para la rampa (señal triangular), con un sobre pico de alrededor del 20% para ambos controladores, cumpliendo así con los requerimientos, por lo cual se procede a realizar la implementación de los controladores.

6. Implementación y Resultados

La implementación de estos controladores fue desarrollada en un lenguaje visual de bloques, en el software de Altair Embed, por lo que solo fue necesario convertir los elementos matriciales en ganancias unitarias aplicadas a los estados, los cuales se obtienen directamente sobre valores calculados mediante el encoder, sin necesidad de contar con un observador de estados, y además se aplican otras dos ganancias a las dos salidas de un integrador doble aplicado sobre la señal de error, que permiten la aplicación del principio del modelo interno. En la figura 13 se muestra una captura de la implementación de ambos controladores en el software.

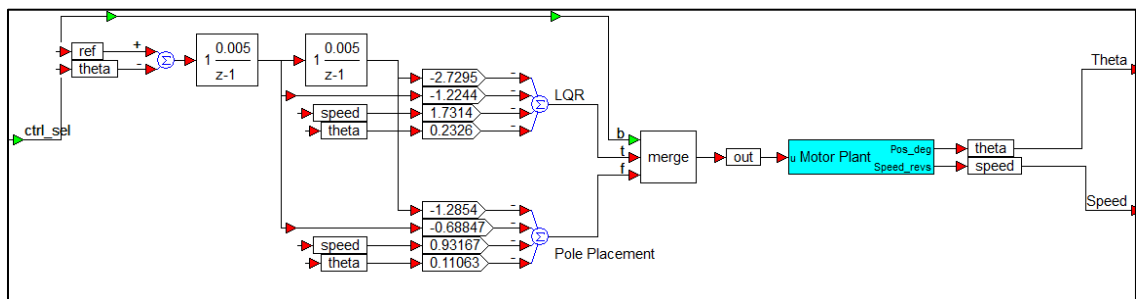


Figura 13: Implementación de Controladores.

De la figura 14 a la figura 17 se muestran las respuestas de los dos controladores a una señal triangular y a una señal cuadrada, ambas de 0.125Hz y de 0 a 50 grados, en todos los casos se evidencia que se cumple con los requerimientos de cero error de estado estacionario para ambas señales, y un sobre pico menor al 25%, siendo -12.5 y 62.5 grados, los límites de la gráfica, y dado que la señal no se sale de estos límites que corresponden a +25% y -25%, se demuestra que se cumple con el requerimiento. De igual forma no se presenta saturación en el motor, puesto que no se sobrepasan los límites de -1 y +1 para la señal PWM.

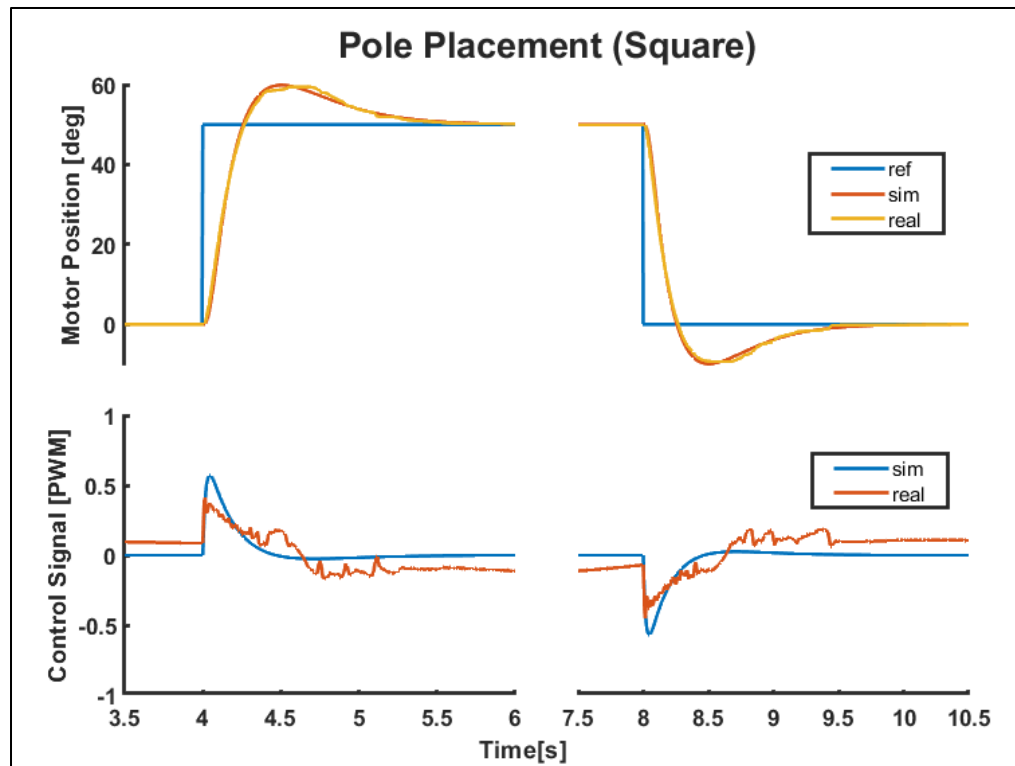


Figura 14: Respuesta de ubicación de polos a la señal cuadrada.

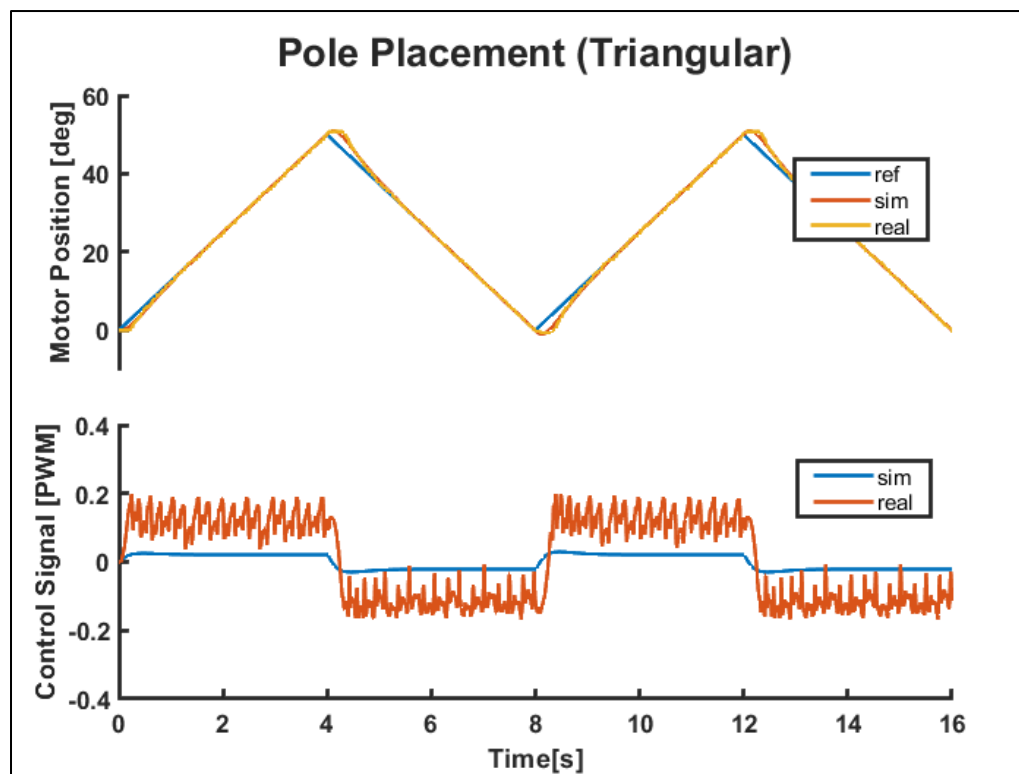


Figura 15: Respuesta de ubicación de polos a la señal triangular.

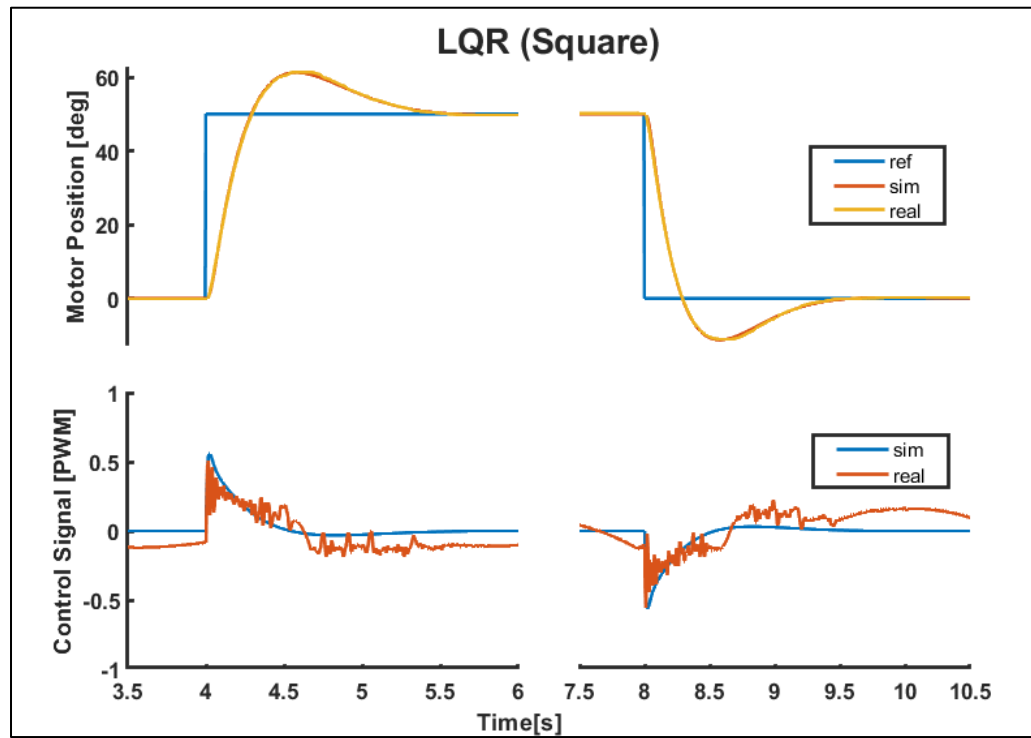


Figura 16: Respuesta de LQR a la señal cuadrada.

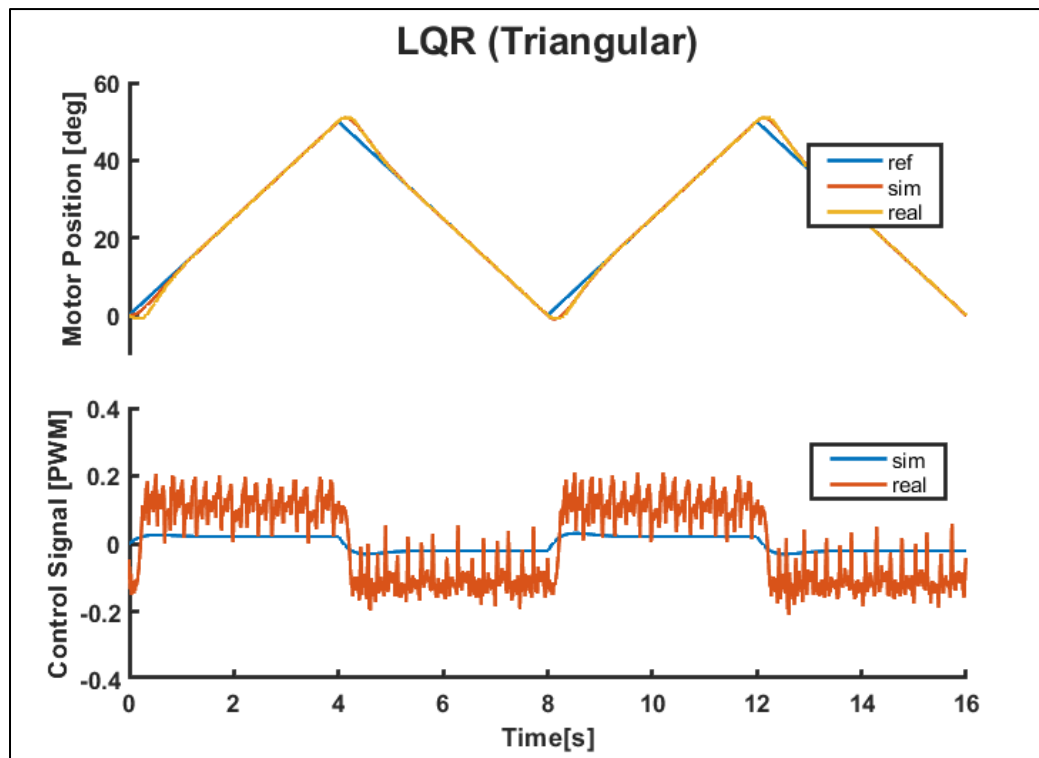


Figura 17: Respuesta de LQR a la señal triangular.

7. Análisis y Conclusiones

Al analizar los resultados es posible concluir que:

- Ambos controladores cumplieron con los requerimientos dados, y de hecho su comportamiento experimental es exactamente igual a la simulación, sin asimetría en el componente ascendente vs el descendente, como se vio en las practicas anteriores al controlar la posición mediante controladores clásicos, esto posiblemente se deba entre otras cosas, al incremento de precisión al usar variables de estado, al uso del software Altair Embed, al tiempo de respuesta tan lento con el que se tuvo que trabajar en comparación al control de velocidad, y al integrador implícito que se tiene al controlar la posición.
- El controlador LQR presenta un comportamiento con mayores oscilaciones y ruido en la señal de control, con respecto al controlador diseñado mediante ubicación de polos, y de hecho este fenómeno se convirtió en un problema reflejado en la señal de posición, cuando no se tenían ajustados apropiadamente los pesos de la función de costo cuadrático, sin embargo a pesar de las oscilaciones, e incluso cuando se tenían los pesos sin ajustar correctamente, la señal aun con ruido presentaba en la baja frecuencia un comportamiento muy similar al de la simulación, siendo evidencia de la robustez de este tipo de controlador.
- En general las señales de control PWM para el control de posición tienen una naturaleza mucho más ruidosa, en comparación al control de velocidad, evidenciando los integradores incluidos en el modelo interno dentro del controlador y en la propia planta.