Control Digital

Trabajo Práctico Final

Alumno: Ing. Lucas Constantino (constantino.lucas12@gmail.com)
Carrera: Maestría en Sistemas Embebidos

Análisis teórico

El análisis teórico se encuentra desarrollado en la primera sección del script de Matlab con nombre **calculos_planta.m**.

En la **Figura 1** se puede ver la planta a controlar. Esta cuenta con dos capacitores por lo que se puede saber a prori que estará representada por un sistema de segundo orden.

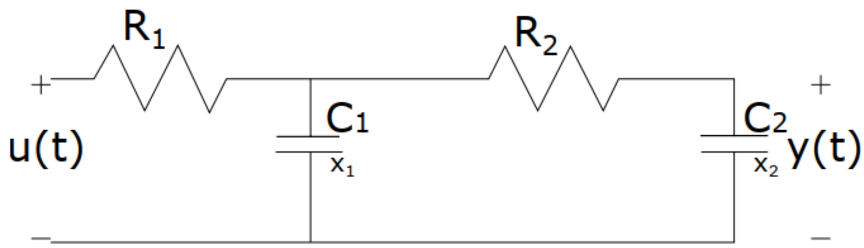

Figura 1: Planta a controlar

Siendo los valores de los elementos pasivos

$$R_1 = 10\ k\Omega$$
$$R_2 = 27\ k\Omega$$
$$C_1 = C_2 = C = 1\ \mu F$$

Calculando la transferencia desde la entrada U(s) a Y(s) (dominio de la frecuencia S) resulta

$$\frac{Y(s)}{U(s)} = \frac{1}{s^2 C_1 C_2 R_1 R_2 + s[R_1(C_1 + C_2) + C_2 R_2] + 1}$$

Reemplazando por los valores en la ecuación se obtiene

$$\frac{Y(s)}{U(s)} = \frac{3703.7}{s^2 + 174.1s + 3703.7}$$

Mirando a la expresión resultante ya se puede concluir que la respuesta del sistema será de tipo pasabajos con ganancia plana en continua (0 dB).
Los polos de la planta resultan

$$\omega_1 = -149.26\ \frac{rad}{seg}$$
$$\omega_2 = -24.81\ \frac{rad}{seg}$$

El signo negativo de los polos significa que estos se encuentran en el semiplano izquierdo en el plano S, por lo tanto, se sabe que la planta es estable. El valor de estos polos en Hz resulta

$$p_1 = 23.7\ Hz$$
$$p_2 = 4\ Hz$$

El valor de los polos indica que el polo p2 es más rápido que p1, esto significa que este último será el polo que más aportará a la respuesta transitoria del sistema ya que p2 se apagará más rápido y sus efectos serán más efímeros.
De todas formas, su separación es menor a un orden de magnitud por lo que p2 **no** se puede despreciar cuando se realice un análisis práctico del sistema.
En la **Figura 2** se puede ver la posición de estos polos en el plano S

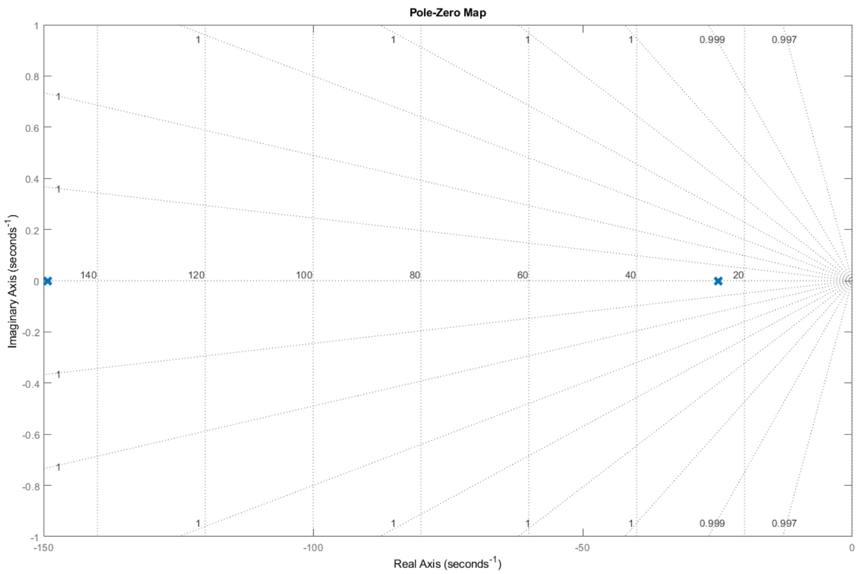

Figura 2: Posición de polos en el plano S

En la **Figura 3** se puede ver la respuesta al escalón del sistema. Esta respuesta resulta sobreamortiguada ya que los polos se encuentran sobre el eje real y separados entre sí.

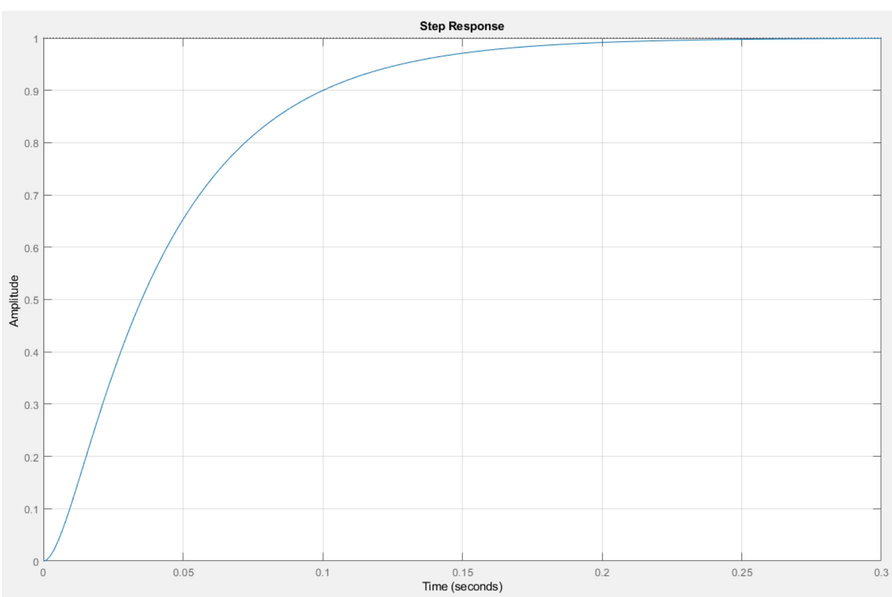

Figura 3: Respuesta al escalón del sistema

Medición del rise time

El script para graficar los valores de entrada y salida de la planta y para el cálculo del rise time encuentra desarrollado en Matlab con nombre **graficador_rt.m**.

Para poder calcular el rise time del sistema se aplicó a la planta una señal cuadrada entre 1V y 2V mediante el DAC. La señal cuadrada tiene un período aproximado de 1.6 segundos y los valores de salida se muestrean cada 8 mseg. En el programa en C la tarea encargada de esto es la denominada **TaskOLResponse.**

En la **Figura 4** se pueden ver tres escalones aplicados a la planta

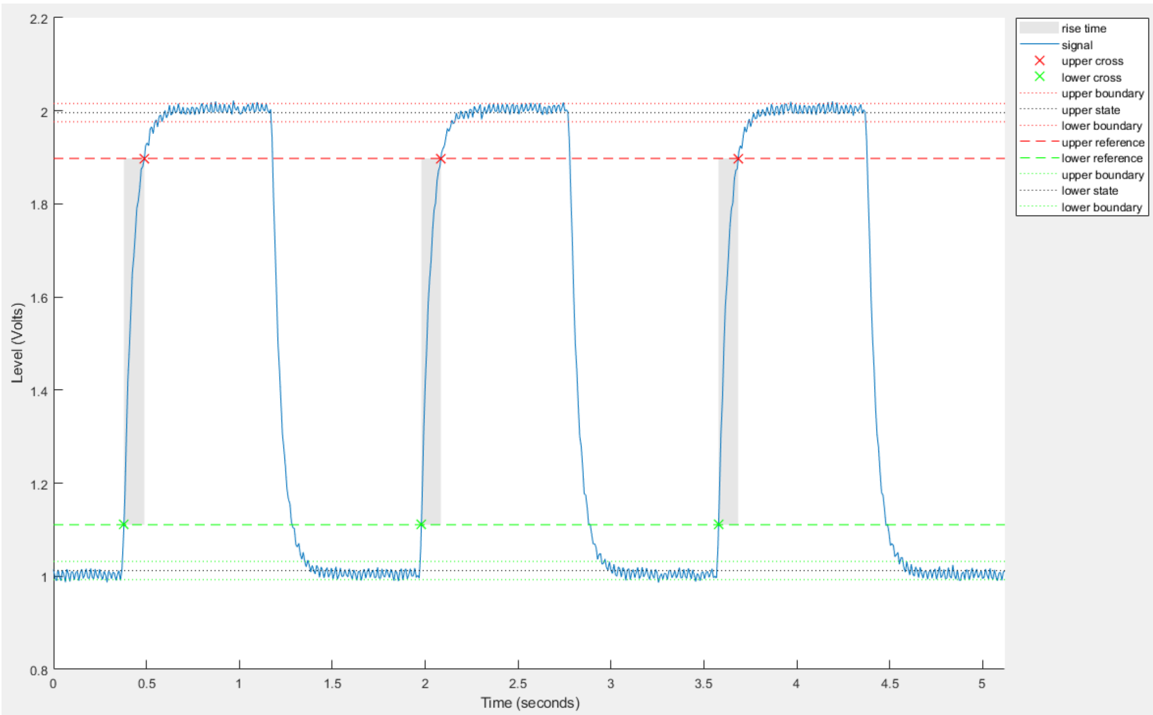

Figura 3: Respuesta al escalón del sistema

Nota: La pequeña oscilación que se puede observar cuando la salida alcanza el estado estacionario es ruido de la red de alta tensión (20 mV de amplitud).

El rise time se computa realizando un promedio de los tres valores de cada escalón resultando

$$rt = 107\ mseg$$

Identificación de la planta

Los cálculos de la planta en tiempo discreto se encuentran en la segunda y tercera sección del script de Matlab con nombre **calculos_planta.m**. En el programa en C la identificación se realiza en la tarea denominada **TaskIdentification**.

Para realizar la identificación del sistema se utiliza el método de cuadrados mínimos (LS) proponiendo un sistema de segundo orden (dos polos y dos ceros). Para ello, se genera una secuencia de niveles de tensión aleatorios con una frecuencia de 66 Hz (cada 15 mseg). Siendo el rise time de 107 mseg, se computa la relación entre ambos

$$N > \frac{107\,mseg}{15\,mseg} = 7.13$$

Por lo tanto, la cantidad de muestras del ensayo vienen dadas por

$$L > 2^N - 1 = 140$$

Luego de correr el ensayo con estos parámetros las constantes del vector theta resultan

$$\Theta = [-a_1\ {-a_0}\ b_2\ b_1\ b_0]$$

$$\begin{gathered} a_1 = -0.856 \\ a_0 = 0.094 \\ b_2 = 0 \\ b_1 = 0.164 \\ b_0 = 0.073 \end{gathered}$$

Se calculan los polos correspondientes a esta planta discreta (muestreando cada 15 mseg con un Zero Order Hold) resultando

$$\begin{gathered} \widehat{pd}_1 = 0.73 \\ \widehat{pd}_2 = 0.13 \end{gathered}$$

En la **Figura 4** se pueden ver los polos y el cero ubicados en el plano Z

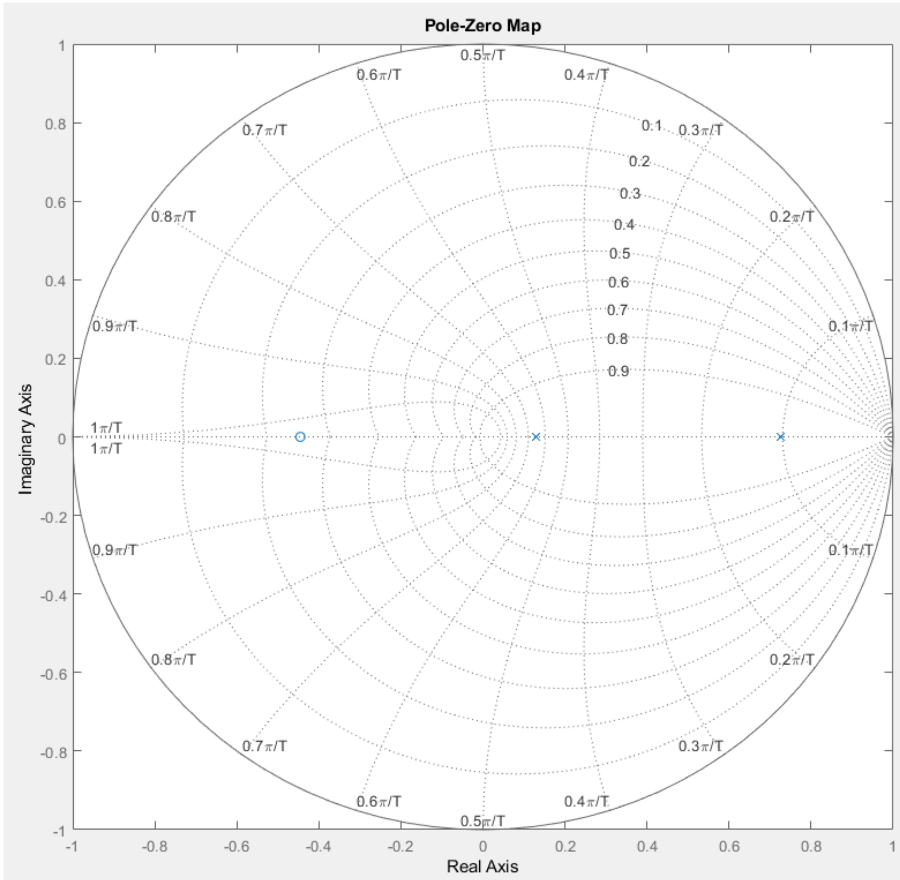

Figura 4: Polos del sistema discreto

Para verificar si la estimación del sistema es buena se pasan los polos discretos a tiempo continuo mediante la fórmula

$$\hat{p} = \frac{\ln(\widehat{pd})}{T_s}$$

Los valores de los polos entonces resultan

$$\hat{p}_1 = 3.39\ Hz$$
$$\hat{p}_2 = 21.7\ Hz$$

La desviación de estos valores con respecto a los teóricos se debe principalmente a dos motivos: en primer lugar, el algoritmo de estimación introduce un error ya que para poder ajustar la curva de los valores estimados a los medidos se debería usar un modelo de planta de orden muy alto, lo cual es muy costoso computacionalmente. Por otro lado, la tolerancia de los componentes pasivos utilizados introduce un desvío en sus valores reales y por consiguiente en los parámetros de la planta.

Controlador PID

En el programa en C el control PID se realiza en la tarea denominada **TaskPIDControl**.

El control PID se aplica para que la respuesta transitoria cumpla con los siguientes requisitos:

1. Tener error de estado estacionario nulo
2. Tener un sobrepico menor al 8%
3. Mejorar el rise time en un 30% respecto a lazo abierto. Para este caso el nuevo rise time resulta de aproximadamente 75 mseg

Antes de aplicar el control PID a la planta se debe establecer una frecuencia de muestreo. Para ello se deben considerar el límite superior e inferior para el tiempo entre muestras.

Como límite máximo se considera la regla de pulgar: se deben tomar entre 4 y 20 muestras durante el intervalo del rise time. Si se toman menos no se llegaría a ver los efectos de los polos y se perdería información. Si se toman más, no brinda información extra y a demás puede volver inestable a la planta (los polos se acercan al círculo unitario en el plano Z).

Como límite mínimo se tiene el tiempo de procesamiento del microcontrolador. El MCU debe ser capaz de actualizar ambos el DAC, el ADC y correr el algoritmo del PID entre dos muestras consecutivas.

Para este caso, como se dijo antes, el polo rápido no se despreciará y es deseable ver sus efectos durante el rise time. Para esto se debe cumplir Nyquist, por lo tanto, se debe muestrear a por lo menos 44 Hz. Si se tiene en cuenta que el nuevo rise time es de 75 mseg, muestrear a 44 Hz implicaría tomar unas 3 muestras durante este intervalo. Por esta razón, se decide muestrear cada 8 mseg (125 Hz) lo que equivale a tomar unas 9 muestras durante el rise time.

Para la sintonización del PID se realizó de forma iterativa el proceso de modificar las constantes y medir la respuesta al escalón hasta llegar a cumplir con los valores. Para ello primero se comienza con un controlador puramente integrativo hasta que se observa que la respuesta alcanza error de estado estacionario nulo con un rise time algo cercano al deseado. Luego se prende la constante proporcional con un valor muy pequeño hasta ajustar más el rise time. En este momento se debe considerar colocar un límite para la acumulación de la variable integradora (wind up). Por último, se prende la constante derivativa con un valor muy pequeño (esta variable puede hacer oscilar al sistema fácilmente) para disminuir el sobrepico.

En la **Figura 5** se puede ver la respuesta al escalón del sistema con el controlador PID implementado.

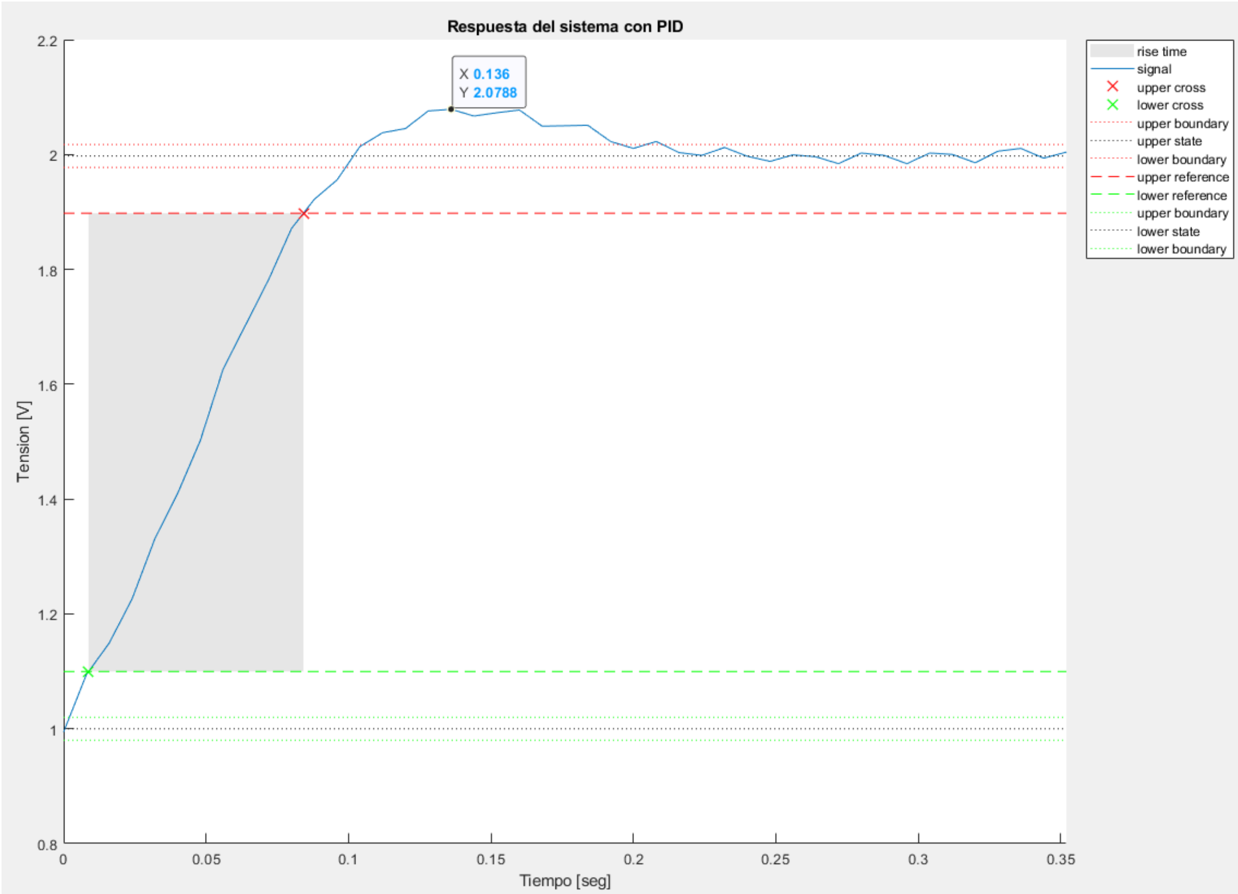

Figura 5: Respuesta al escalón con PID

Los valores obtenidos de la señal controlada son

$$rt = 75.8\ mseg$$
$$OS = 7.88\ \%$$

Los valores de las constantes del PID son

$$P = 1$$
$$I = 32$$
$$D = 0.005$$

En la **Figura 6** se puede ver la respuesta al escalón del sistema junto con la señal de control que sale del PID implementado.

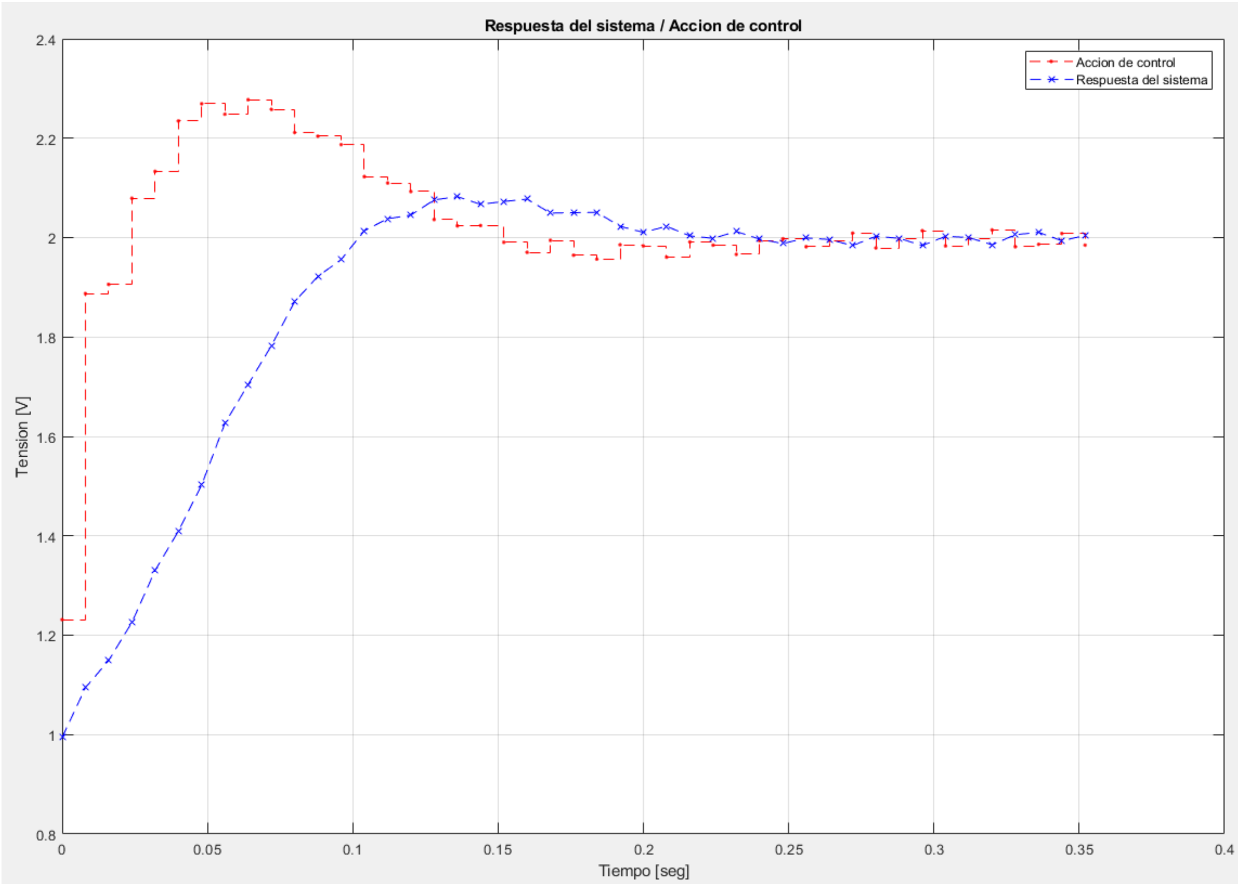

Figura 6: Respuesta al escalón con PID y acción de control

Lo que se puede ver en la figura anterior es que para alcanzar una respuesta más rápida que la de lazo abierto la acción de control debe excitar a la planta con una tensión mayor a la del setpoint. De esta forma se fuerza temporalmente al sistema para que la velocidad de subida sea mayor. Luego, para evitar overshoot la señal de control disminuye su valor (incluso por debajo del setpoint) para compensar.

Control por realimentación de estados (pole placement)

Los cálculos de la planta en tiempo discreto para la realimentación de estados se encuentran en el script de Matlab con nombre **pole_placement.m**. En el programa en C el control mediante esta técnica se realiza en la tarea denominada **TaskPolePlacement**.

Para realizar un control por realimentación de estados primero se debe obtener el modelo del sistema en espacio de estados. Siendo las ecuaciones de estado del sistema en tiempo continuo

$$\frac{dx}{dt} = Ax + Bu$$

$$y = Cx + Du$$

Siendo el vector de estados

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Donde x_1 es la tensión en el capacitor C_1 y la tensión x_2 la tensión de salida sobre el capacitor C_2 (ver **Figura 1**).

Para determinar las matrices A, B, C y D no se puede trabajar directamente sobre la función transferencia porque eso implicaría que las variables de estado sean una la derivada de la otra, lo cual no es cierto para el circuito en cuestión. Por ello, se debe obtener otra representación del sistema utilizando el método de mallas.

Por lo tanto, recorriendo ambas mallas se obtienen las expresiones x_1 y x_2

$$X_1 = \frac{(V_i - X_1)}{R_1}\frac{1}{sC_1} + \frac{(X_2 - X_1)}{R_2}\frac{1}{sC_1}$$

$$X_2 = \frac{(X_1 - X_2)}{R_2}\frac{1}{sC_2}$$

Multiplicando ambos a lados por S y reagrupando términos resulta

$$\dot{X_1} = X_1\left(-\frac{1}{R_1C_1} - \frac{1}{R_2C_1}\right) + \frac{X_2}{R_2C_1} + \frac{V_i}{R_1C_1}$$

$$\dot{X_2} = \frac{X_1}{R_2C_2} - \frac{X_2}{R_2C_2}$$

Teniendo las ecuaciones de esta forma ya se pueden construir las matrices A, B, C y D

$$A = \begin{bmatrix} -\frac{1}{R_1C_1} - \frac{1}{R_2C_1} & \frac{1}{R_2C_1} \\ \frac{1}{R_2C_2} & -\frac{1}{R_2C_2} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{R_1C_1} \\ 0 \end{bmatrix}$$

$$C = [0 \quad 1]$$

$$D = 0$$

Una vez obtenido el modelo en espacio de estados en tiempo continuo se obtiene su equivalente en tiempo discreto utilizando la frecuencia de muestreo determinada anteriormente (8 ms).

Para realizar la realimentación de estados se multiplica el vector X por un vector de ganancia K y se lo realimenta a la señal de referencia, quedando el diagrama del sistema como se ve en la **Figura 7**

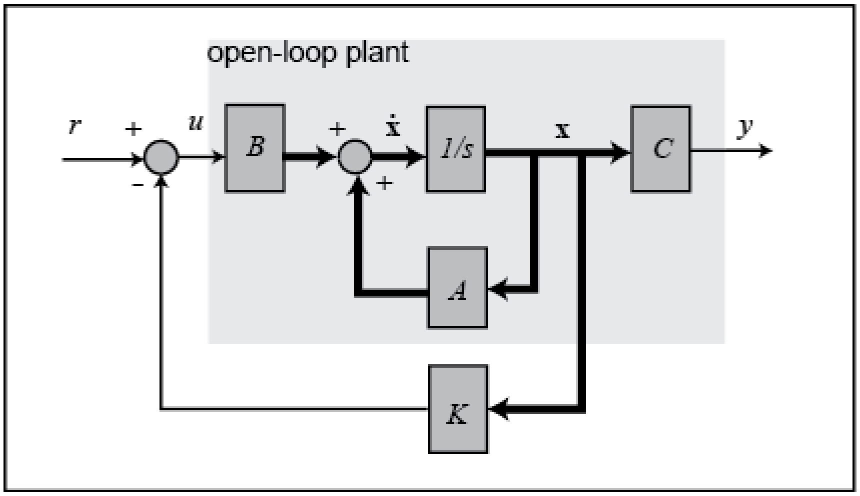

Figura 7: Diagrama de realimentación de estados

Luego se debe hallar el vector de constantes K tal que los dos polos del sistema a lazo cerrado resulten los deseados. Para el valor estos polos se consideraron dos juegos de valores. Para el primero se consideró que toda la respuesta transitoria se deba a un solo polo como si el sistema fuese de primer orden, por lo que se fijó uno de ellos a alta frecuencia (cercano a 0 en el plano Z) para que sus efectos sean despreciables. El valor del otro polo se lo computa utilizando la fórmula que relaciona al rise time y el ancho de banda de un sistema (en Hz)

$$BW = \frac{0.35}{rt}$$

Teniendo en cuenta el valor del rise time hallado anteriormente el valor del polo resulta

$$BW = \frac{0.35}{75mseg} = 4.67\ Hz$$

Pasando el valor del polo al plano Z resulta

$$p = e^{-2\pi \mathrm{BW} T_S} = 0.79$$

Por lo tanto, el par de polos resulta

$$P = [0.79 \quad 0.1]$$

Ahora se puede computar el valor del vector de realimentación K

$$K = [0.51 \quad 0.283]$$

Además del vector K se debe calcular la constante de la referencia Kf para eliminar el error de estado estacionario y que la salida converja a la referencia. Esto se debe a que el sistema a lazo cerrado ahora posee otros polos y por lo tanto otra ganancia en continua. Para calcular esta constante se utiliza la fórmula

$$K_f = \frac{1}{C(I-(A-BK))^{-1}B}$$

$$K_f = 1.79$$

Esta variable también se puede calcular de forma práctica simplemente observando el valor en estado estacionario de la salida del sistema para Kf = 1. Luego se la calcula tomando la inversa de este valor, compensando la salida.

La ley de control resulta entonces

$$u = K_f r - Kx$$

$$u = 1.79r - \begin{bmatrix} 0.51 \\ 0.283 \end{bmatrix} x$$

Teniendo todas las constantes halladas se realiza la simulación de la salida para una entrada escalón, resultando la respuesta de la **Figura 8**

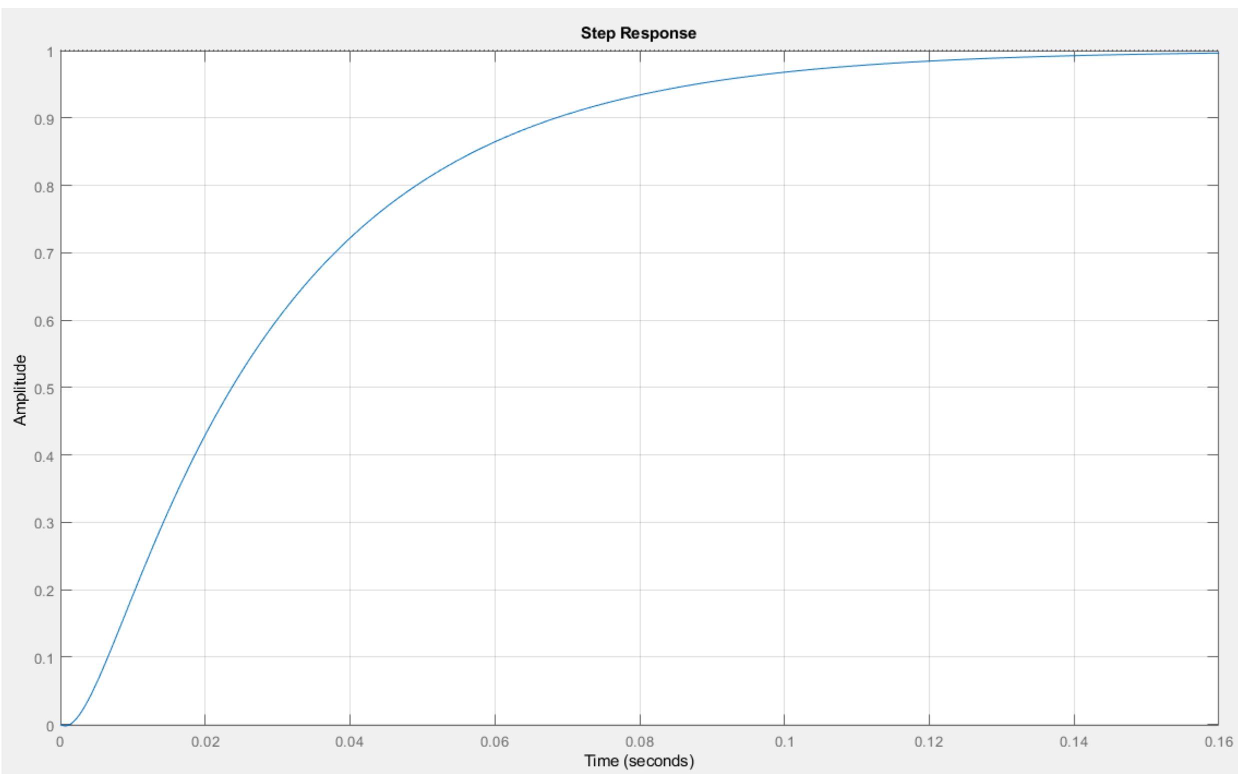

Figura 8: Respuesta al escalón con realimentación de estados

En la **Figura 9** se puede ver las dos variables de estado X1, X2 y la señal de control u entrante a la planta

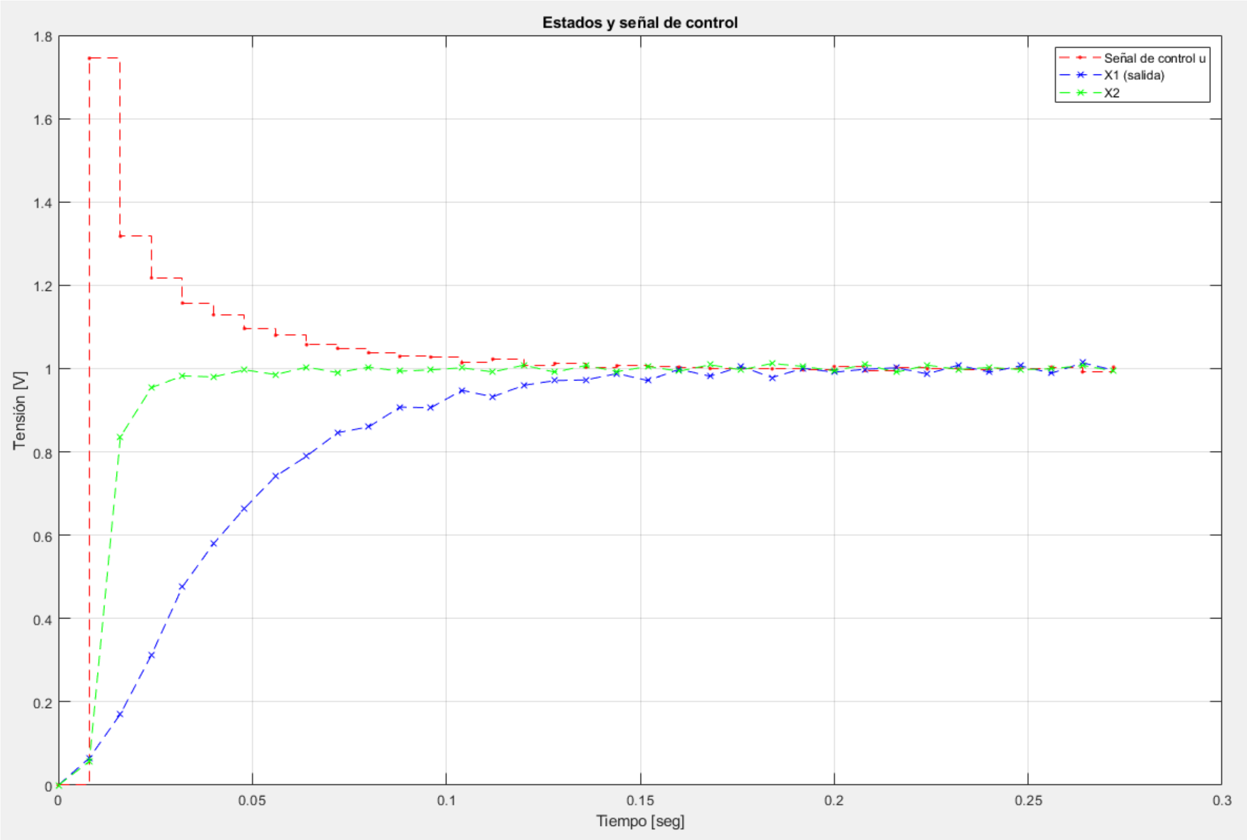

Figura 9: Respuesta transitoria de estados y señal de control (polo dominante)

El rise time obtenido es

$$rt = 74.2\ mseg$$

Cabe destacar que el valor calculado para el polo (0.79) no generaba una salida con las características esperadas (unos 15 mseg más lenta) por lo que se tuvo que hacer un ajuste manual del valor. El polo resultante debe ubicarse ligeramente por debajo del calculado, en 0.75.

Esta discrepancia se atribuye principalmente a que la planta difiere ligeramente en algunos de sus parámetros al modelo teórico. En este caso, la planta real resulto ser un poco mas lenta que la teórica. Por eso es lógico que el polo haya tenido que ser corregido a una frecuencia más alta.

Por otro lado, además de los calculados se buscaron otros valores para los polos de lazo cerrado de tal forma que en lugar de que solamente uno aporte toda la respuesta transitoria (polo dominante), los dos lo hagan de forma equitativa.

Estos valores se buscaron de forma iterativa variando las constantes y midiendo la respuesta de la planta. El par de polos resultaron

$$P' = [0.643 \quad 0.642]$$

Las constantes son

$$K = [-0.394 \quad 0.412]$$

$$K_f = 1.02$$

El rise time medido resulta

$$rt = 73.3\, mseg$$

En la **Figura 10** se pueden ver los valores transitorios de las variables de estado y de la señal de control.

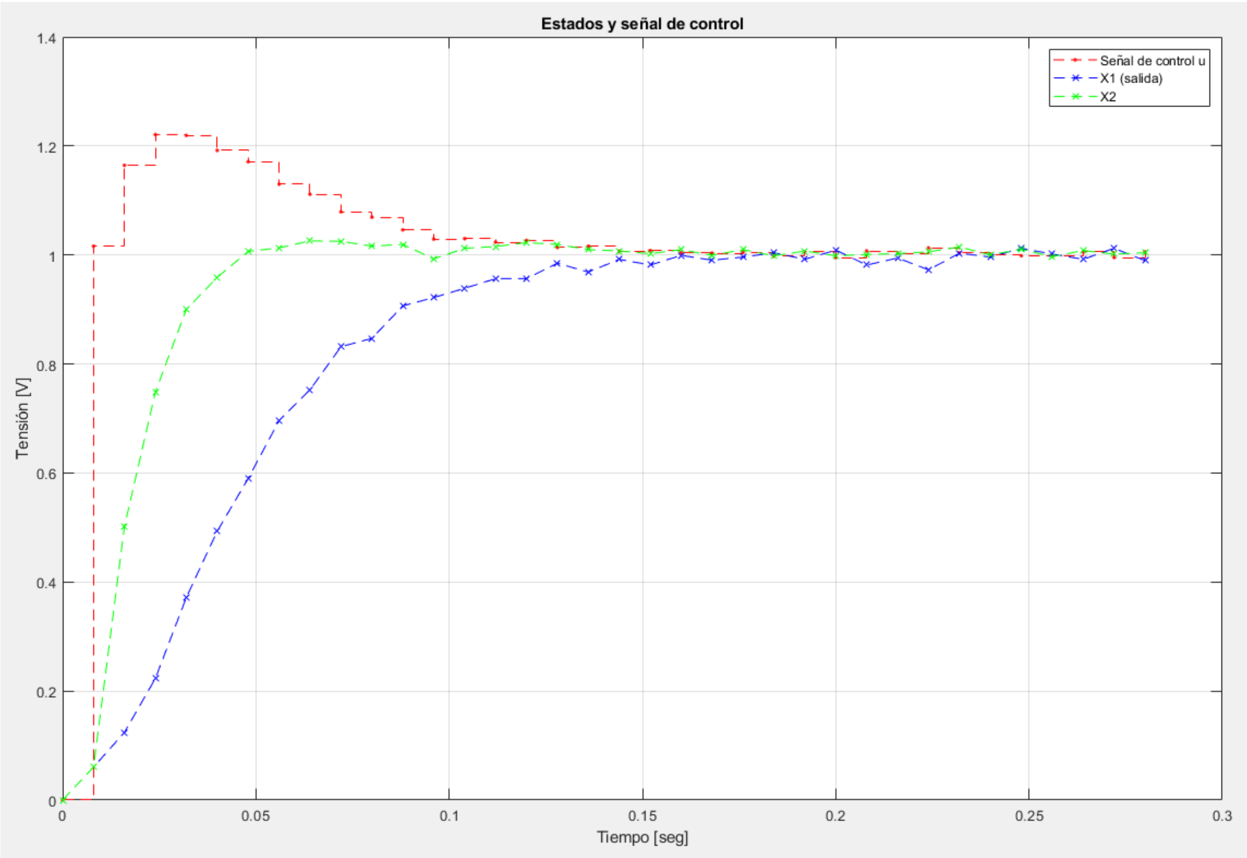

Figura 10: Respuesta transitoria de estados y señal de control

Nótese que en este caso la señal de control a penas llega a los 1.2V mientras que en el caso anterior casi alcanzaba los 1.8V. Esto se debe a que en este caso los dos polos aportan a la respuesta transitoria del sistema y por lo tanto no se lo debe forzar tanto.

Control por realimentación de estados estimados (observador)

Los cálculos de la planta en tiempo discreto para la realimentación de estados estimados mediante observador se encuentran en el script de Matlab con nombre **observador.m** (basado en el provisto por la cátedra). En el programa en C el control mediante esta técnica se realiza en la tarea denominada **TaskObserver**.

Para realizar un control por observador se mide solamente la salida de la planta (X_2) y en base a esta medición se estiman el resto de las variables de estado, haciendo uso del modelo en espacio de estados discreto de la planta.

Luego, el vector de estados que en pole placement se realimentaba a la entrada de referencia es reemplazado por el vector de estados estimados. Así, el problema se divide en dos partes que funcionan en conjunto, pero se diseñan de forma individual (realimentación de estados y observador), quedando el diagrama completo del controlador y el proceso como se muestra en la **Figura 11.**

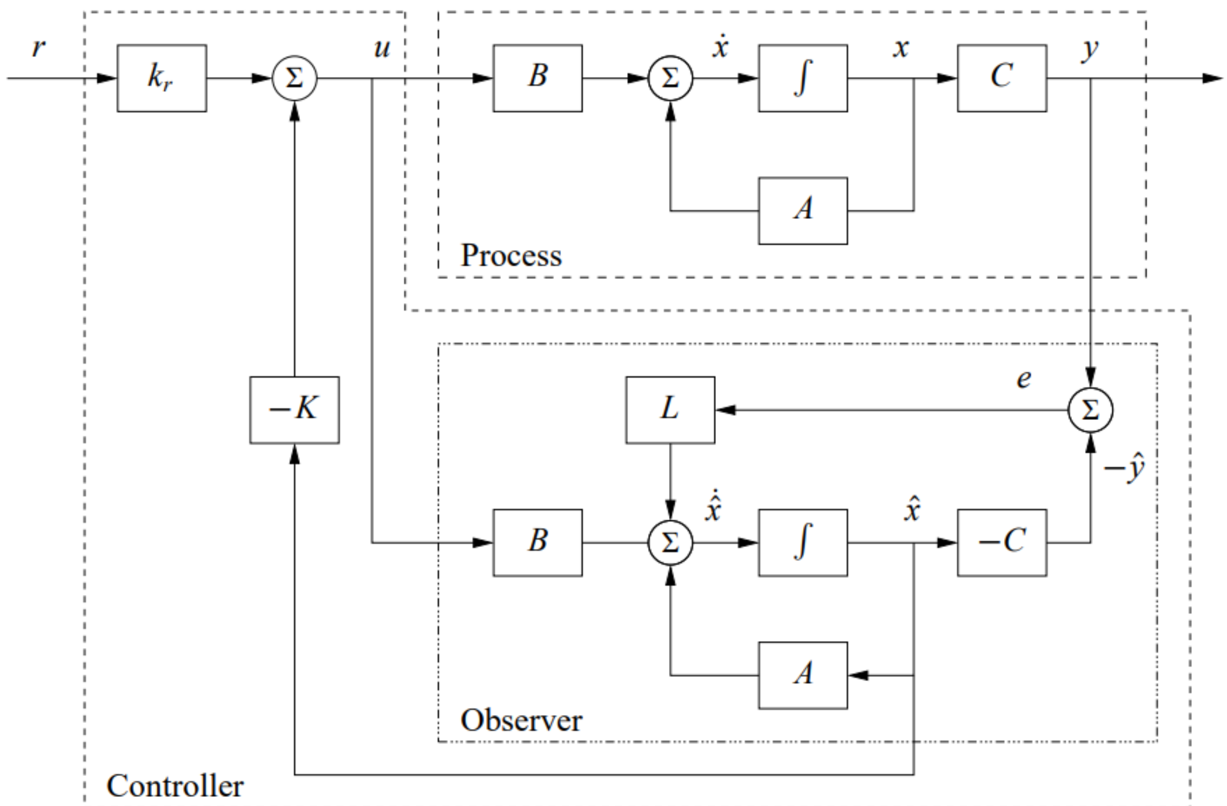

Figura 11: Diagrama en bloques del observador y realimentación de estados *(Feedback Systems, Astrom – Murray, p.213)*

Ahora la ecuación de estados incluyendo el observador resulta

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x})$$

Siendo y la medición de la salida del sistema y u la ley de control definida en el apartado anterior.

El valor del vector L determina la posición de los polos del observador y por lo tanto la respuesta del vector de estados estimados. El valor de este vector se desea que converja mucho más rápido que el vector de estados de la planta física. Para ello la regla que en general se aplica es que los polos del observador sean por lo menos 3 o 4 veces mas rápidos que los polos de la planta a lazo cerrado.

Para este caso se usa el polo doble en 0.64 (con las constantes K y K_f halladas en el punto anterior) para la realimentación de estados y aplicando la regla antes mencionada se establecen entonces los polos del observador en

$$P_O = \begin{bmatrix} 0.1 \\ 0.11 \end{bmatrix}$$

Estos polos son aproximadamente 5 veces mas rápidos que los polos de lazo cerrado. Para determinar el valor del vector L se realiza un pole placement (mediante Matlab) con las matrices A y C resultando

$$L = \begin{bmatrix} 0.56 \\ 0.91 \end{bmatrix}$$

Corriendo el script en Matlab la simulación del vector de estados estimados resulta como se ve en la **Figura 12**.

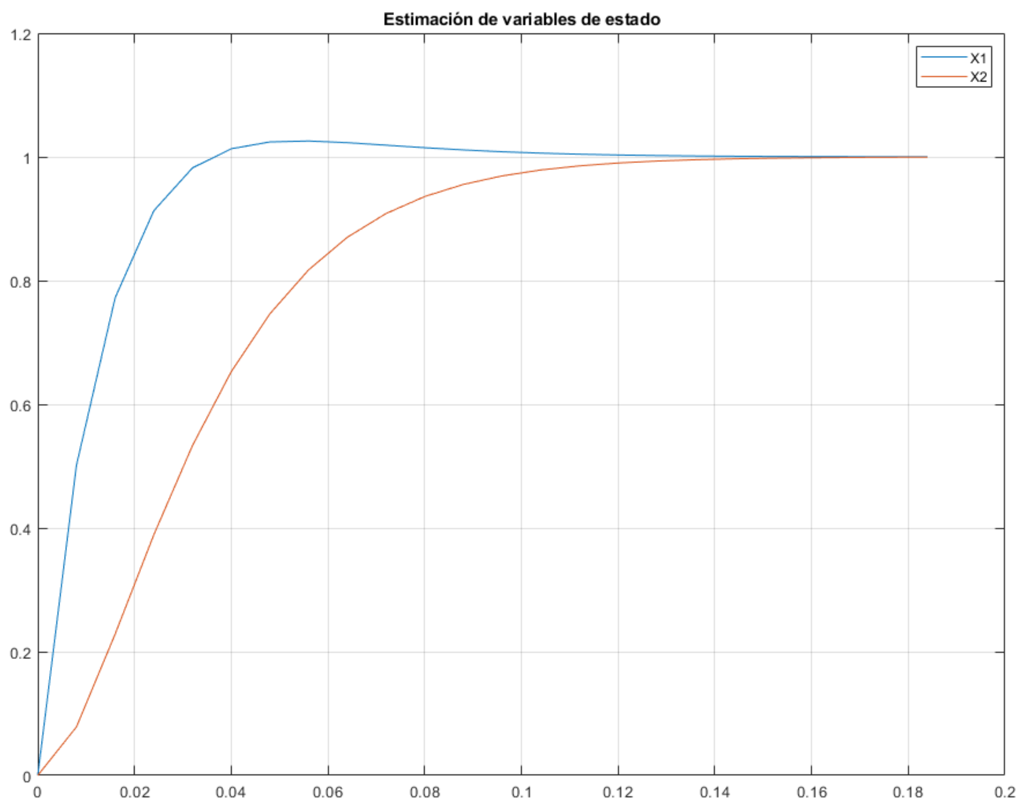

Figura 12: Simulación de estimación de estados

Implementando la ecuación de estados antes definida en el microcontrolador se obtiene la salida mostrada en la **Figura 13.**

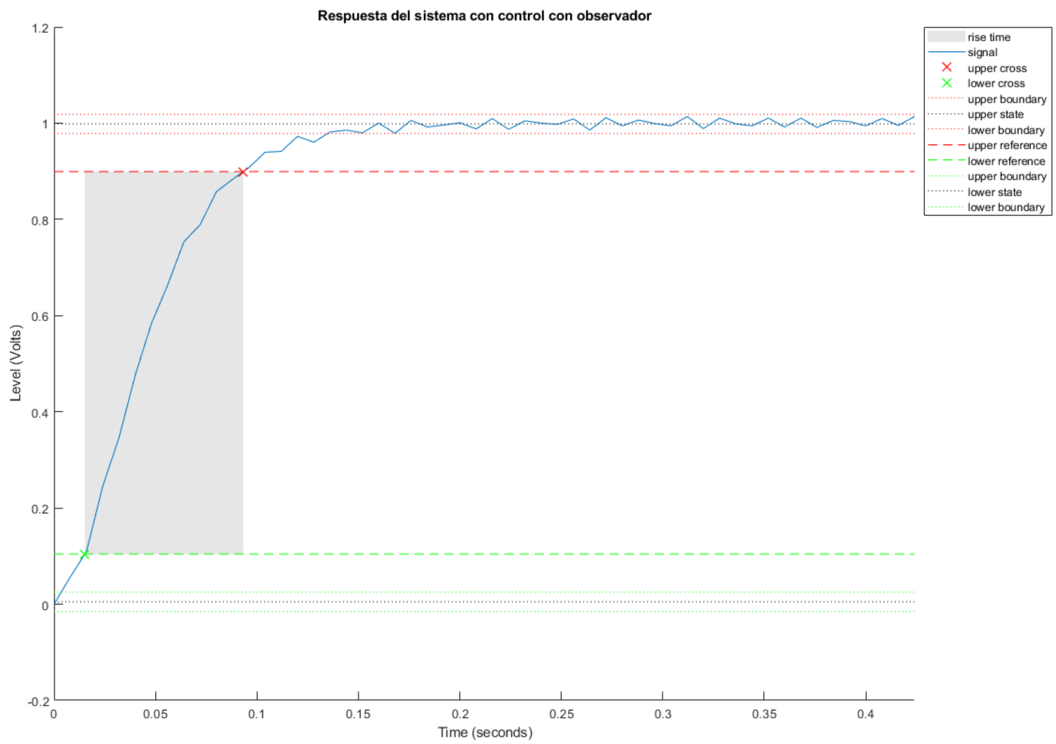

Figura 13: Salida del sistema con control por estimación de estados

El rise time medido resulta

$$rt = 78.1\ mseg$$

En la **Figura 14** se puede ver la salida del sistema medida, la estimación de las variables de estado y la señal de control u.

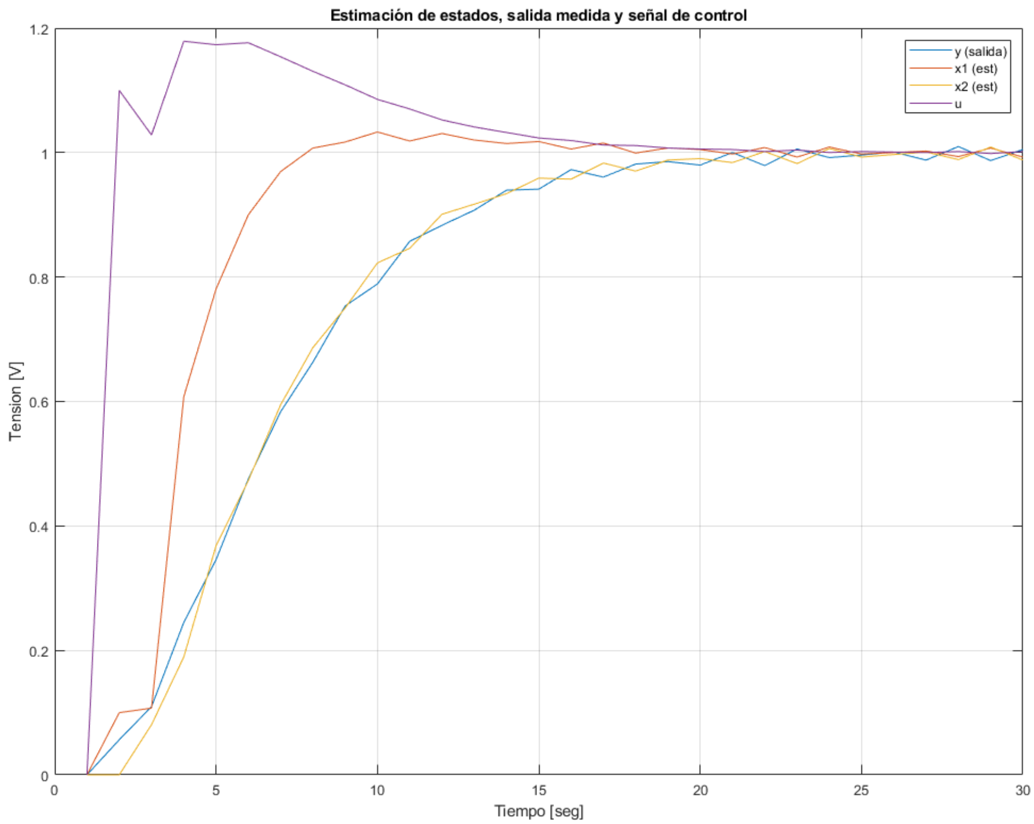

Figura 14: Salida del sistema medida, estados estimados y señal de control

Como se puede ver, la señal medida y su estimación $\hat{X}_2$ tienen cierta diferencia al principio y una vez que el sistema avanza hacia el estado estacionario la estimación converge rápidamente al valor medido. Este tiempo es el que viene determinado por los polos del observador y el vector L.

Por otro lado, la estimación de la variable X_1 coincide bastante bien con su valor teorico mostrado en la figura anterior, presentando un porcentaje de sobrepico similar.

Por último, la señal de control también resulta muy similar a la correspondiente al control por realimentación de estados (Figura 10), alcanzado un pico de 1.2V. Esto es lógico ya que la ley de control u es igual en ambos casos, con la diferencia que en este caso se calcula utilizando el vector de estados estimados.