

IMPLANTACIÓN DE CONTROLADORES DE ORDEN FRACCIONARIO MEDIANTE AUTÓMATAS PROGRAMABLES

Antonio José Calderón Godoy

E.II.II. Universidad de Extremadura (Badajoz), ajcalde@unex.es

Concepción Alicia Monje Micharet

E.II.II. Universidad de Extremadura (Badajoz), cmonje@unex.es

Blas Manuel Vinagre Jara

E.II.II. Universidad de Extremadura (Badajoz), bvinagre@unex.es

Vicente Feliu Batlle

E.II.II. Universidad de Castilla la Mancha (Ciudad Real), Vicente.Feliu@uclm.es

Resumen

El objeto de este trabajo es introducir la posibilidad de utilizar unos equipos empleados típicamente en tareas de automatización industrial, como es el caso de los autómatas programables, para la implementación de controladores cuya ley de control será la correspondiente a un PD, PI o PID de orden fraccionario. Para ello se ha desarrollado un módulo software que facilita dicha labor de implantación en el autómata S7_224 de Siemens. Igualmente, se han implementado controladores de orden no entero mediante la tarjeta de adquisición de datos PCL818. Los resultados obtenidos se presentan en este trabajo, validando la implementación de estos controladores.

Palabras Clave: Controlador fraccionario, autómata programable, discretización.

1 INTRODUCCIÓN

Tradicionalmente, los autómatas programables han sido utilizados en tareas orientadas a la automatización de procesos industriales. Estas aplicaciones estaban caracterizadas por la tipología de las operaciones realizadas por el controlador lógico programable (PLC), consistentes fundamentalmente en operaciones lógicas a nivel de bit (booleanas), de memoria, temporización, contaje, controles secuenciales, etc. Con el desarrollo de elementos especializados, las aplicaciones del autómata programable se extienden al control de procesos, al poder efectuar lazos de regulación trabajando con dispositivos de instrumentación. Estas últimas aplicaciones se han visto potenciadas por el incremento en la velocidad de respuesta (ciclo de ejecución) del autómata y el desarrollo de interfaces

de entrada/salida inteligentes, tales como servocontroladores, controladores PID, etc. Además, los lenguajes de programación de autómatas incluyen mejoras incorporando nuevas instrucciones más potentes que posibilitan los cálculos matemáticos con datos en notación de coma flotante.

Si bien actualmente los autómatas de las gamas media y alta incluyen módulos, tanto hardware como software, para poder insertarlos en lazos de regulación, éstos están limitados a la elaboración de las acciones básicas de control, a saber: P, PD, PI o PID. En este trabajo se desarrolla la implantación en un autómata programable de la generalización de las anteriores acciones básicas de control. Esta generalización se consigue utilizando operadores integro-diferenciales de orden fraccionario para obtener las acciones derivativa e integral de orden no entero. De esta forma, además de los tres grados de libertad de que disponen los PID clásicos (la ganancia proporcional, K_p , y los tiempos integral y derivativo, T_i y T_d), se añade la posibilidad de ajustar el orden de integración (λ) y derivación (μ), con lo que se incrementa en dos el número de parámetros susceptibles de ajuste. El PID fraccionario se formula como:

$$C(s) = k_p + \frac{1}{T_i s^\lambda} + T_d s^\mu \quad (1)$$

En resumen, en este trabajo se ha creado un módulo software para, utilizando un autómata programable, poder realizar un lazo de regulación con acción de control PID generalizada.

El resto del trabajo se ha estructurado como sigue. En la sección 2 se describe el método empleado para la sintonía de estos controladores fraccionarios. En la sección 3 se exponen los casos de estudio a tratar,

describiendo las plantas a controlar. La sección 4 presenta los resultados obtenidos tras la implementación de los controladores empleando la tarjeta de adquisición de datos PCL818 y el autómata programable. Igualmente se describe el módulo software desarrollado para programar el autómata. Finalmente, en la sección 5 se establecen las conclusiones de este trabajo.

2 MÉTODO DE SINTONÍA DE CONTROLADORES $PI^\lambda D^\mu$ FRACCIONARIOS

El uso de controladores del tipo $PI^\lambda D^\mu$ fraccionarios permite, frente al PID convencional, la consecución de hasta 5 especificaciones de diseño, ya que cuenta con 5 parámetros a sintonizar. El método de sintonía propuesto para estos controladores se basa en este caso en especificaciones de frecuencia de paso por cero, ω_{cg} , margen de fase, ϕ_m , robustez ante cambios en la ganancia de la planta, restricción en la función de sensibilidad y rechazo al ruido de alta frecuencia. Dicho método busca, mediante un algoritmo iterativo, el valor de los parámetros del controlador que minimiza estas cinco funciones. Este problema de minimización no lineal se ha resuelto empleando la función *fmincon* de Matlab (ver [1]). En la sección siguiente puede confirmarse la validez del método de diseño.

3 CASOS DE ESTUDIO

En este trabajo se ha llevado a cabo el diseño y la implementación de controladores $PI^\lambda D^\mu$ fraccionarios para el control de dos sistemas reales de primer orden, que son:

- Una planta de nivel de líquido, cuya función de transferencia medida experimentalmente es:

$$G'(s) = \frac{3.13}{433.33s + 1} \quad (2)$$

Se ha añadido por software un retardo (e^{-Ls}) a esta planta, de valor $L=5\text{seg}$, y se han diseñado los controladores correspondientes para estos dos casos (planta con y sin retardo).

- Un servomotor de velocidad, cuya caracterización experimental resulta en la función de transferencia (3):

$$G''(s) = \frac{0.544}{0.3s + 1} \quad (3)$$

A continuación se muestran los controladores obtenidos para cada una de estas plantas en función de las especificaciones fijadas para el sistema controlado.

3.1 PLANTA DE NIVEL DE LÍQUIDO SIN RETARDO

En este caso, la función de transferencia de la planta es:

$$G_1(s) = \frac{3.13}{433.33s + 1} \quad (4)$$

Las especificaciones requeridas para el sistema controlado son las siguientes:

- Frecuencia de paso por cero, $\omega_{cg}=0.01\text{rad/seg}$.
- Margen de fase, $\phi_m=70^\circ\text{deg}$.
- Robustez ante variaciones de la ganancia de la planta.
- Función de sensibilidad:
 $|S_1(j\omega)|_{\text{dB}} \leq -20\text{dB}, \forall \omega \leq \omega_s = 0.001\text{rad/seg}$.
- Rechazo al ruido de alta frecuencia:
 $|T_1(j\omega)|_{\text{dB}} \leq -20\text{dB}, \forall \omega \geq \omega_r = 100\text{rad/seg}$.

Empleando el método de sintonía mencionado en el apartado 2, el controlador fraccionario resultante es:

$$C_1(s) = 0.1173 + \frac{0.1005}{s^{0.5489}} + 1.9485s^{0.3971} \quad (5)$$

Debe tenerse en cuenta que el integrador fraccionario debe implementarse como:

$$\frac{1}{s^\lambda} = \frac{1}{s} s^{(1-\lambda)} \quad (6)$$

asegurando así el efecto de un integrador de orden entero a muy baja frecuencia y cancelando el error estacionario. La parte fraccionaria $s^{(1-\lambda)}$ se ha implementado en este caso mediante la aproximación continua de Oustaloup del derivador de orden no entero (ver [2]). Se han empleado cinco polos y cinco ceros en la aproximación y un rango de frecuencias para la misma de 0.01rad/seg a 100rad/seg . El derivador fraccionario, s^μ , se implementa igualmente con dicha aproximación de Oustaloup, empleando el mismo número de polos y ceros y el mismo rango de frecuencias.

Una vez obtenido el controlador continuo, se ha procedido a su discretización mediante la regla de Tustin, usando un periodo de muestreo de $t_s=1\text{seg}$.

Los diagramas de Bode obtenidos para el sistema en lazo abierto, $F_1(s)=C_1(s)G_1(s)$, se muestran en la figura 1.

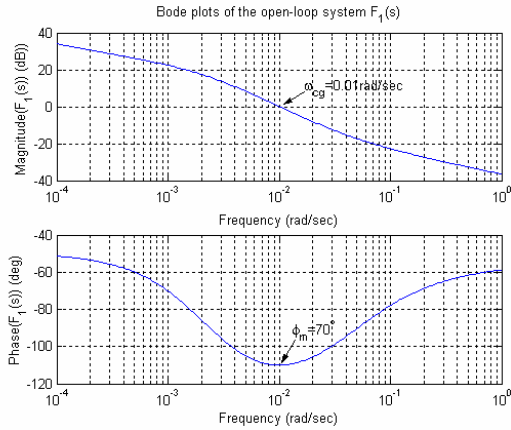


Figura 1: Diagramas de Bode del sistema en lazo abierto $F_1(s)$

Como puede observarse, se cumplen las especificaciones de frecuencia de paso por cero y margen de fase. Además, se fuerza a la fase del sistema a ser plana en un rango de frecuencias centrado en ω_{cg} , lo que se traduce en robustez ante cambios en la ganancia de la planta (dentro de unos límites de variación). Este hecho se observa en la figura 2, donde se representan las respuestas temporales del sistema en lazo cerrado a un escalón de 0.5 para distintos valores de ganancia.

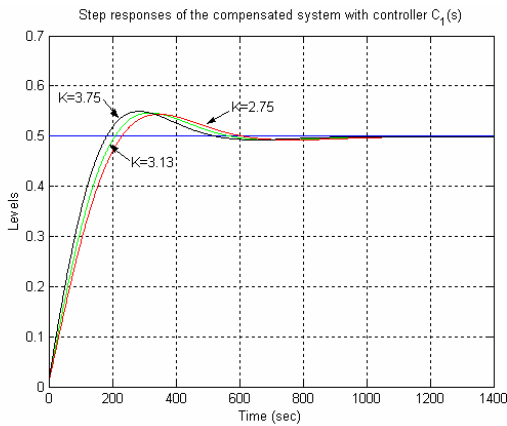


Figura 2: Respuestas temporales del sistema controlado con el controlador $C_1(s)$

Las figuras 3 y 4 muestran, respectivamente, las magnitudes de $S_1(j\omega)$ y $T_1(j\omega)$, cumpliéndose las especificaciones anteriormente citadas.

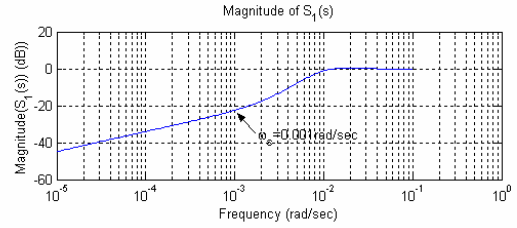


Figura 3: Magnitud de $S_1(s)$

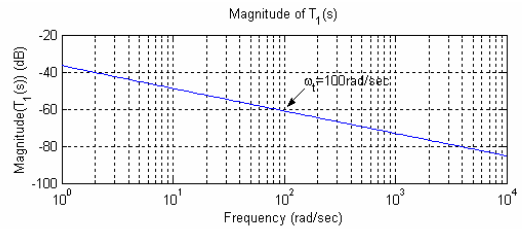


Figura 4: Magnitud de $T_1(s)$

3.2 PLANTA DE NIVEL DE LÍQUIDO CON RETARDO $L=5\text{seg}$

Añadiendo un retardo por software de 5seg a la planta de nivel de líquido del caso anterior, resulta la siguiente función de transferencia:

$$G_1(s) = \frac{3.13}{433.33s + 1} e^{-5s} \quad (7)$$

Ahora, las especificaciones requeridas para el sistema son:

- Frecuencia de paso por cero, $\omega_{cg}=0.01\text{rad/seg}$.
- Margen de fase, $\phi_m=50^\circ\text{deg}$.
- Robustez ante variaciones de la ganancia de la planta.
- Función de sensibilidad:
 $|S_1(j\omega)|_{\text{dB}} \leq -20\text{dB}, \forall \omega \leq \omega_s = 0.002\text{rad/seg}$.
- Rechazo al ruido de alta frecuencia:
 $|T_1(j\omega)|_{\text{dB}} \leq -20\text{dB}, \forall \omega \geq \omega_t = 10\text{rad/seg}$.

En este caso, el controlador fraccionario obtenido es:

$$C_2(s) = 0.0469 + \frac{0.0469}{s^{0.7333}} + 1.4747s^{0.3146} \quad (8)$$

La implementación continua de este controlador se realiza de la misma manera que en el caso anterior. Igualmente, la discretización del mismo se realiza mediante la regla de Tustin, con un periodo de muestreo de $t_s=1\text{seg}$.

Los diagramas de Bode del sistema en lazo abierto, $F_2(s)=C_2(s)G_2(s)$, se muestran en la figura 5,

cumpléndose las especificaciones de margen de fase y frecuencia de paso por cero.

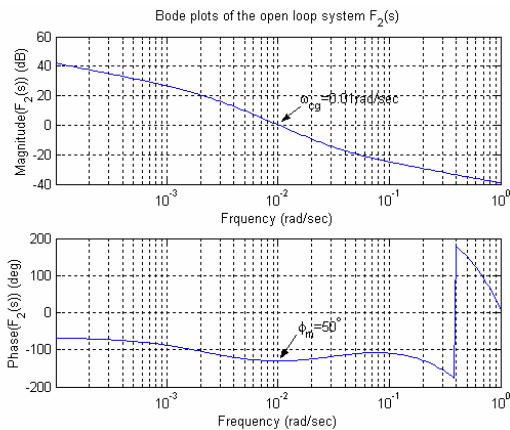


Figura 5: Diagramas de Bode del sistema en lazo abierto $F_2(s)$

La robustez ante cambios de ganancia se observa en la respuestas temporales de la figura 6.

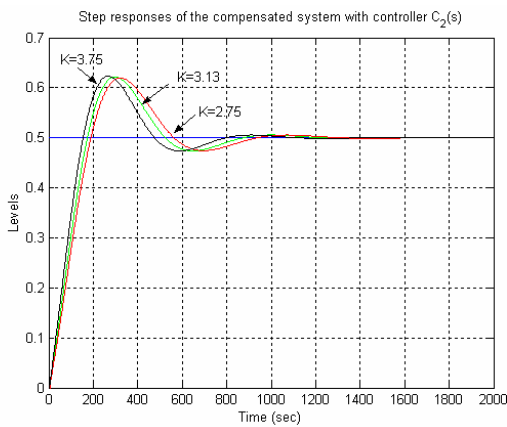


Figura 6: Respuestas temporales del sistema controlado con el controlador $C_2(s)$

Las magnitudes de $S_2(s)$ y $T_2(s)$ cumplen nuevamente las especificaciones (figuras 7 y 8, respectivamente).

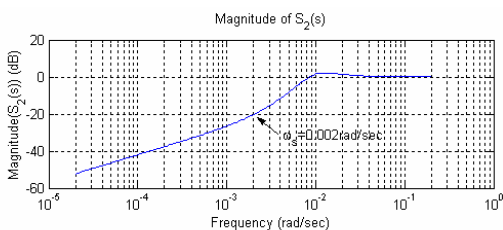


Figura 7: Magnitud de $S_2(s)$

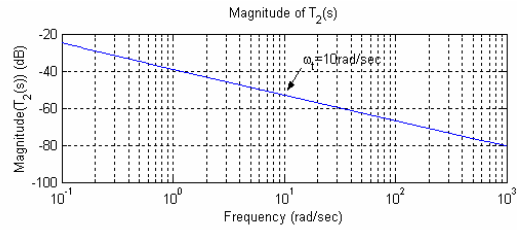


Figura 8: Magnitud de $T_2(s)$

3.3 SERVOMOTOR DE VELOCIDAD

Para el control del servomotor de velocidad se ha diseñado un PI^1 . La función de transferencia de la planta es:

$$G_3(s) = \frac{0.544}{0.3s + 1} \quad (9)$$

Las especificaciones requeridas para el sistema son:

- Frecuencia de paso por cero, $\omega_{cg}=10\text{rad/seg}$.
- Margen de fase, $\phi_m=70^\circ\text{deg}$.
- Robustez ante variaciones de la ganancia de la planta.

El controlador resultante es:

$$C_3(s) = 3.4545 \left(1 + \frac{0.1410}{s^{0.8121}} \right) \quad (10)$$

A diferencia de los controladores anteriores, este controlador se ha implementado mediante la definición de Grünwald-Letnikov de derivador discreto de orden no entero (ver [3]). Se ha empleado un número de 5 coeficientes y un periodo de muestreo de 0.1seg. En la sección 4.2.2. se explica con detalle cómo se implanta dicho controlador en el autómata partiendo de su función de transferencia en z (dominio discreto).

La figura 9 muestra los diagramas de Bode del sistema en lazo abierto $F_3(s)$, donde se observa el cumplimiento de las especificaciones.

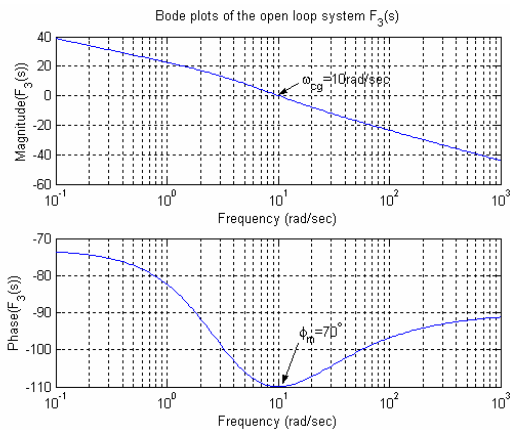


Figura 9: Diagramas de Bode del sistema en lazo abierto $F_3(s)$

4 IMPLANTACIÓN DE LOS CONTROLADORES FRACCIONARIOS

Para la implementación de los controladores fraccionarios obtenidos se han empleado los siguientes dispositivos:

- La tarjeta de adquisición de datos PCL818, bajo el entorno Matlab. Los dos controladores obtenidos para la planta de nivel de líquido ($C_1(s)$ y $C_2(s)$) se han implantado empleando este dispositivo. El experimento se ha realizado con un periodo de muestreo de $t_s=1$ seg.
- Autómata programable (PLC) S7_224, de Siemens. Con este dispositivo se ha implantado el controlador PI^λ fraccionario $C_3(s)$ para el servomotor de velocidad.

A continuación se mostrarán los resultados experimentales obtenidos en cada caso, explicando con detalle el proceso de implantación en el PLC.

4.1 TARJETA DE ADQUISICIÓN DE DATOS PCL818

Con este dispositivo se han implantado los controladores diseñados para la planta de nivel de líquido, empleando un periodo de muestreo de 1seg. Deben comentarse que el tanque de llenado parte de una posición inicial distinta de cero (vacío). Esto se debe a que en los momentos iniciales de llenado se crean en dicho tanque ciertas turbulencias que impiden emplear el modelo de primer orden especificado inicialmente para esta planta.

A continuación se muestran los resultados experimentales obtenidos.

4.1.1 Planta de nivel de líquido sin retardo

Las figuras 10, 11, 12 y 13 muestran los resultados obtenidos de la implantación del controlador $C_1(s)$. La comparativa entre las respuestas temporales y las leyes de control simuladas y experimentales concluyen que se han obtenido buenos resultados tras la implementación. Igualmente se comprueba que las pruebas de robustez, realizadas sobre el sistema variando la ganancia de la planta por software, han sido satisfactorias, como se observa en la figura 11.

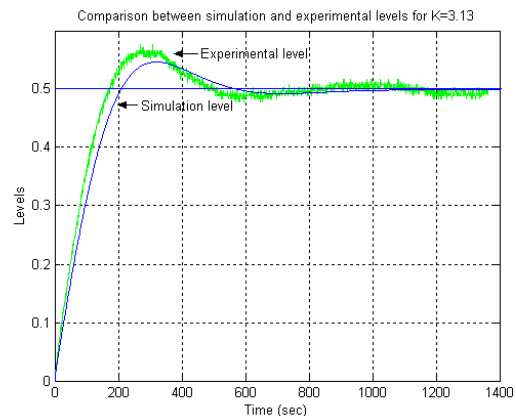


Figura 10: Comparación entre la respuesta temporal simulada y experimental con el controlador $C_1(s)$

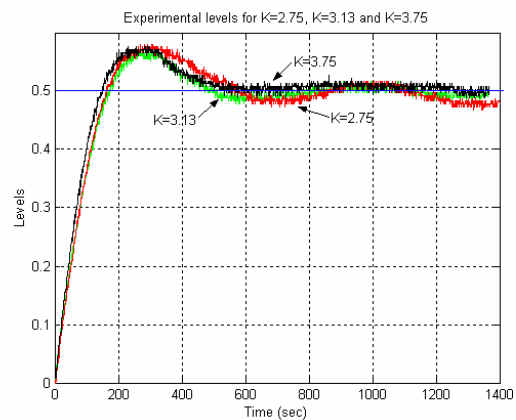


Figura 11: Respuestas temporales experimentales para distintos valores de ganancia con el controlador $C_1(s)$

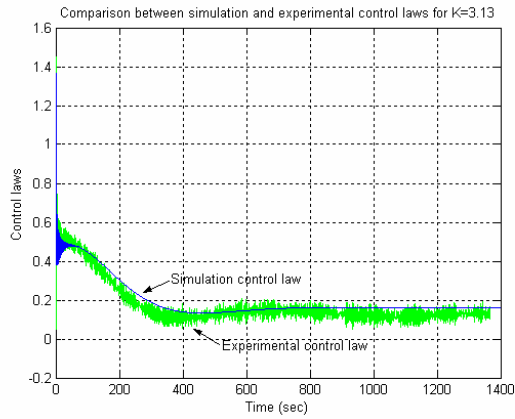


Figura 12: Comparación entre la ley de control simulada y experimental con el controlador $C_1(s)$

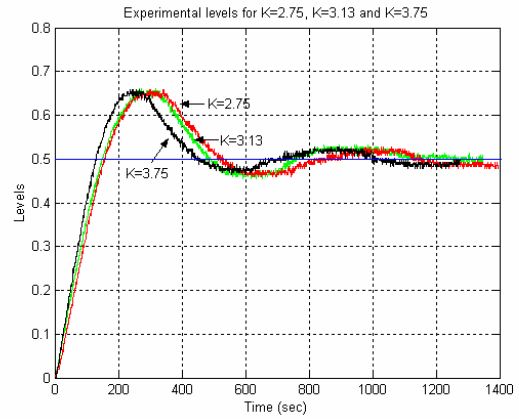


Figura 15: Respuestas temporales experimentales para distintos valores de ganancia con el controlador $C_2(s)$

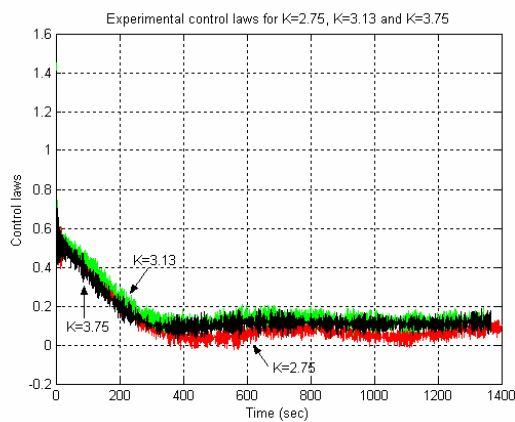


Figura 13: Leyes de control experimentales para distintos valores de ganancia con el controlador $C_1(s)$

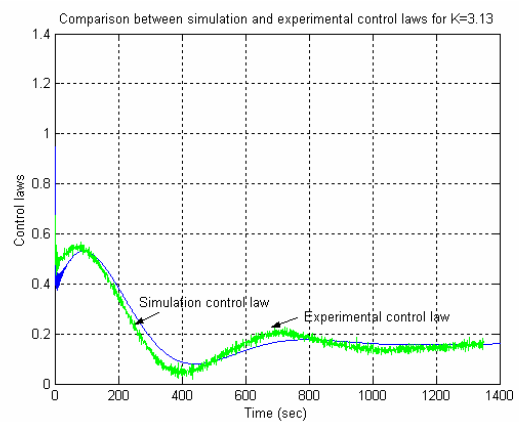


Figura 16: Comparación entre la ley de control simulada y experimental con el controlador $C_2(s)$

4.1.2 Planta de nivel de líquido con retardo $L=5\text{seg}$

Nuevamente, las gráficas de las figuras 14, 15, 16 y 17 muestran los buenos resultados experimentales obtenidos y la robustez del sistema real ante cambios en la ganancia de la planta.

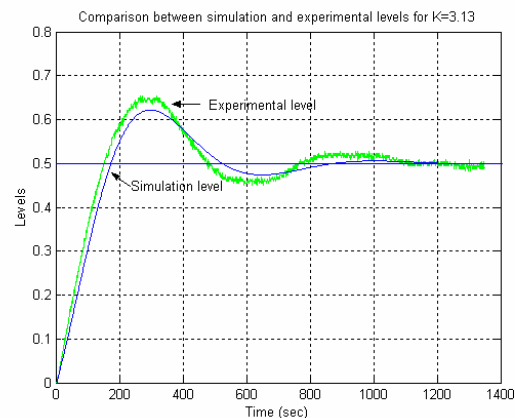


Figura 14: Comparación entre la respuesta temporal simulada y experimental con el controlador $C_2(s)$

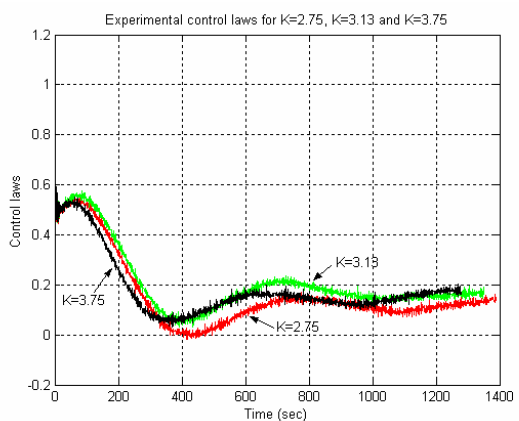


Figura 17: Leyes de control experimentales para distintos valores de ganancia con el controlador $C_3(s)$

4.2 AUTÓMATA PROGRAMABLE S7_224

En este apartado se presenta el módulo software para la implantación del controlador PID generalizado de orden fraccionario en el autómata programable.

Como se ha mencionado, los autómatas programables industriales suelen disponer de módulos, tanto software como hardware, que posibilitan su inserción en lazos de regulación resolviendo algoritmos de control PID. En ellos, la programación del controlador se lleva a cabo mediante la parametrización del módulo, que consiste en indicar el valor de las constantes del controlador PID y del periodo de muestreo utilizado. Si el algoritmo de control se resuelve mediante un módulo hardware, éste incluye también los interfaces de entrada/salida analógicos para su conexión a la planta bajo control. Si el algoritmo de control se resuelve mediante un módulo software, como en el caso que se expone, el PLC ha de equiparse con módulos de entradas y salidas analógicas para realizar la conexión con el proceso.

Lo ideal sería disponer de un módulo software igualmente parametrizable para la obtención del controlador fraccionario. Esto es, un módulo en el que se pudieran introducir como parámetros, además de las constantes del PID tradicional, los órdenes de las acciones integral y derivativa. No obstante, como un controlador fraccionario presenta una función de transferencia no racional en el dominio de Laplace, o una función de transferencia de orden ilimitado en el dominio Z , se han de obtener aproximaciones realizables de las funciones de transferencia que caracterizan los controladores para su implantación con dispositivos reales. Por ello, para la realización de estos controladores se ha de obtener una aproximación discreta de los mismos, y será la función de transferencia obtenida de tal aproximación la que resuelva el módulo software diseñado.

4.2.1 Descripción del sistema hardware

El sistema hardware sobre el que se ha desarrollado el módulo software de control fraccionario está basado en la CPU s7_224 de Siemens. La CPU utilizada cuenta con la posibilidad de ejecución aritmética en coma flotante (con una velocidad de ejecución de 100 μ s por instrucción aritmética frente a los 0.37 μ s por instrucción booleana), indispensable para efectuar las operaciones necesarias para resolver el algoritmo de control cometiendo el mínimo error.

Esta CPU se ha ampliado con un módulo de E/S analógicas EM235, que dispone de 4 entradas analógicas y 1 salida analógica, con un convertidor A/D de 12 bits de resolución y un tiempo de conversión analógica/digital menor a 250 μ s.

4.2.2 Descripción del módulo software

Para la programación del módulo software que implemente un controlador PID generalizado de orden fraccionario, se partirá del equivalente discreto de éste; esto es, de una función de transferencia discreta que de forma general se puede escribir como:

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}} \quad (11)$$

donde $a_i, b_i \in \mathfrak{R}$ y n es el orden.

Con objeto de evitar problemas de sensibilización de coeficientes, para implementar la función de transferencia $D(z)$ se realiza la descomposición en subsistemas de segundo orden del tipo:

$$D_j(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad (12)$$

La realización de la función de transferencia discreta, $D(z)$, se llevará a cabo utilizando una estructura en paralelo, cuyo diagrama de bloques se muestra en la figura 18. Para ello, es necesario hacer su desarrollo mediante fracciones parciales, obteniendo:

$$D(z) = A + D_1(z) + D_2(z) + \dots + D_m(z) \quad (13)$$

$$= A + \sum_{i=1}^j \frac{c_i}{1 + p_i z^{-1}} + \sum_{i=j+1}^m \frac{a_{i0} + a_{i1} z^{-1}}{1 + b_{i0} z^{-1} + b_{i1} z^{-2}}$$

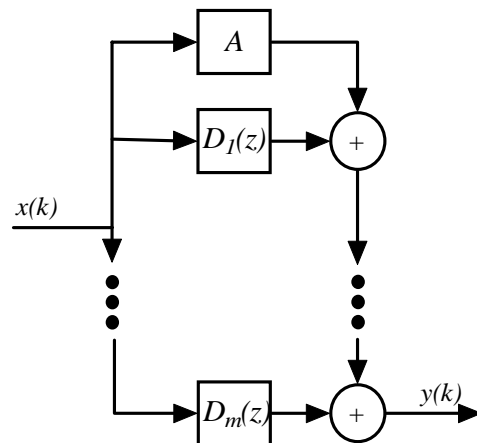


Figura 18: Realización en paralelo

Con esta estructura, teóricamente, se pueden calcular en un solo ciclo los valores de salida $y_i(k)$ para, en un segundo ciclo, obtener el valor de la salida $y(k)$. Sin embargo, para ello sería necesario disponer de un computador con capacidad de cálculo vectorial. En el caso de utilizar un autómata programable, el

algoritmo ha de ser secuencial, respondiendo el diagrama de flujo de la solución programada al que se muestra en la figura 19.

El programa de usuario para el autómata programable que resuelve el módulo software diseñado se ha estructurado en tres partes:

OB1: Bloque principal de ejecución cíclica, en el que se inicializan las variables, se programa el periodo de muestreo y se está a la espera del evento de interrupción que lanza la rutina que resuelve el algoritmo de control.

INT_0: Rutina de interrupción que se ejecuta periódicamente a la cadencia que indique el periodo de muestreo y donde se ejecutan las tareas que aparecen en el diagrama de la figura 20.

DB1: Bloque de datos, donde se almacenan los coeficientes a_i , b_i del controlador.

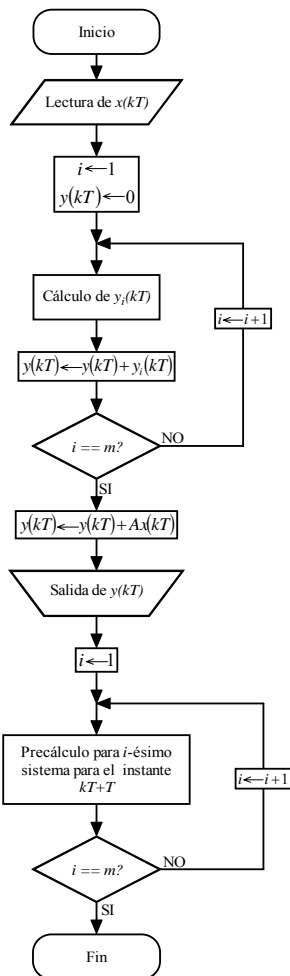


Figura 19: Diagrama de flujo para la programación del algoritmo de control

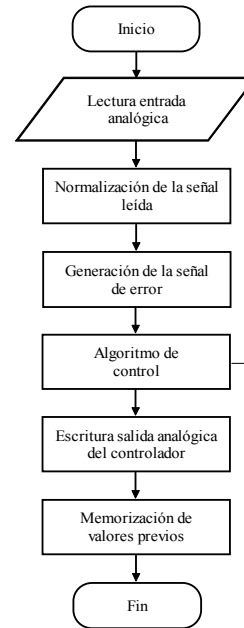


Figura 20: Diagrama de flujo del programa de usuario

A continuación se presentan los resultados obtenidos tras la implementación del controlador $C_3(s)$ con el PLC.

4.2.3 Servomotor de velocidad

Los resultados experimentales obtenidos para este caso se muestran en las figuras 21 y 22, donde se comparan la respuesta temporal y ley de control simuladas y experimentales. Los resultados obtenidos son buenos y validan la implantación.

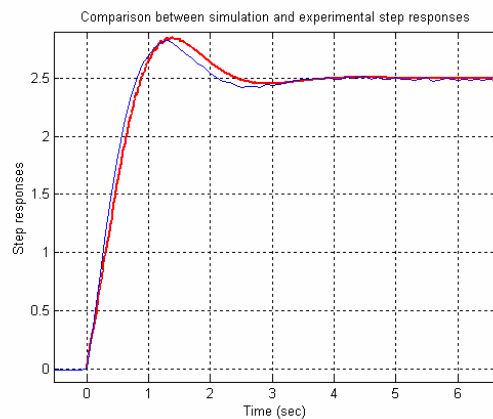


Figura 21: Comparación entre la respuesta temporal simulada y experimental para el controlador $C_3(s)$

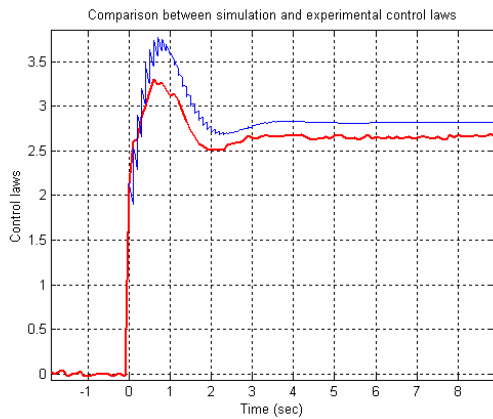


Figura 22: Comparación entre la ley de control simulada y experimental para el controlador $C_3(s)$

5 CONCLUSIONES

En este trabajo se ha llevado a cabo la implantación de controladores de orden no entero mediante el empleo de un autómata programable S7_224. Se ha desarrollado un módulo software para dicho propósito, facilitando al usuario la tarea de implementación. Igualmente, se han presentado los resultados obtenidos de la implementación de este tipo de controladores mediante la tarjeta de adquisición de datos PCL818. De los experimentos realizados puede concluirse que la implementación es factible y exitosa en ambos casos.

Referencias

- [1] Monje, C.A., Vinagre, B.M., Chen, Y.Q., Feliu, V., Lanusse, P., Sabatier, J., (2004) "Proposals for fractional $PI^{\lambda}D^{\mu}$ tuning", *FDA'04*, pp. 156-161, Bordeaux, France.
- [2] Oustaloup, A., (1995) *La dérivation non entière*, Hermes, Paris.
- [3] Oldham, K., Spanier, J., (1974) *The fractional calculus*, Academic Press, New York.