

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN



ENTREGABLE 2 - SECO

MODELADO E IMPLEMENTACIÓN

JAVIER LÓPEZ INIESTA DÍAZ DEL CAMPO
FERNANDO GARCÍA GUTIÉRREZ

12 de Abril de 2021

Índice

1. Introducción	2
2. Diseño e implementación de la arquitectura software en el hardware del RealLabo	3
2.1. Señales de entrada del encoder	3
2.2. Diseño del PWM	4
2.3. Generación y envío de datos	6
2.4. Cálculo de la reductora	6
3. Modelado experimental de un motor DC con la arquitectura hardware y software implementadas	7
3.1. Fase 1: Obtención del valor medio de velocidades angulares del régimen permanente.	9
3.2. Fase 2: Cálculo del polo medio de varianza mínima.	10
3.3. Fase 3: Cálculo de K y de la tensión de entrada equivalente.	11
3.4. Fase 4: Identificación de la función no lineal del modelo del motor mediante la técnica de interpolación polinómica.	12
4. Análisis, diseño e implementación de un controlador para un control de posición angular	14
4.1. Controlador Proporcional	15
4.2. Controlador Proporcional Derivativo	18
4.3. Controlador Proporcional Integral	22
4.4. Controlador Proporcional Integral Derivativo	25
5. Conclusiones	28

1. Introducción

El control automático ha tenido un rápido desarrollo en los últimos años. Ahora que las computadoras digitales se han vuelto más baratas y más compactas, se usan como parte integral de los sistemas de control. Las aplicaciones recientes de la teoría de control moderna incluyen sistemas ajenos a la ingeniería, como los biológicos y los económicos. Actualmente, las principales aplicaciones de controladores las podemos encontrar en control de motores, control de aire acondicionado, bombas de movimiento, seguidores solares, fuentes de alimentación, inversores fotovoltaicos, control de regadío en un explotación agrícola, etc.

Por ello, en el Entregable 2 de la asignatura de Sistemas Electrónicos de Control del itinerario de Sistemas Electrónicos del GITST se busca realizar el modelado e implementación de un controlador para el control de la posición angular de un motor DC. Para conseguir dicho objetivo, se realizará primero un diseño de un software orientado a la creación de un PWM que trabaje a frecuencias mayores a 20 kHz. Posteriormente, se modelará experimentalmente el motor DC del laboratorio. Finalmente, se realizará un diseño, análisis e implementación de un controlador para el control de la posición angular.

En la Figura 1 se puede observar el hardware utilizado del *Robolabo*, que consta de un Arduino Due [1], con un microcontrolador (*Atmel SAM3X8E ARM Cortex-M3*) [2]. Además, cuenta con una etapa de potencia [4], que posee un puente en H (*L6206*) [3]. Por último, en el *RealLabo* podemos encontrar un motor DC (*Pololu 25 mm 12 V*) [7], un encoder magnético [8], una reductora y una rueda.

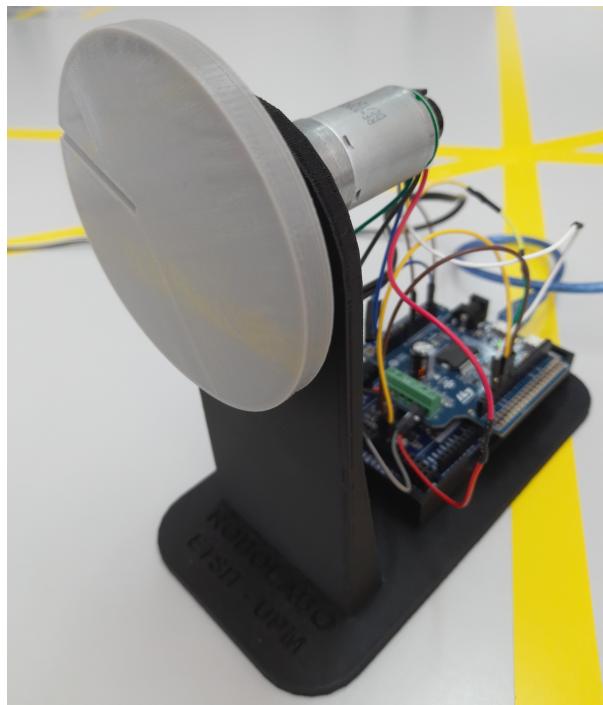


Figura 1: Hardware del RealLabo

2. Diseño e implementación de la arquitectura software en el hardware del RealLabo

La metodología seguida para la realización de este apartado fue la siguiente:

1. Conseguir familiarizarse con el software de Arduino Due, con el objetivo de poner en funcionamiento el motor.
2. Lectura de las señales de entrada de los canales A y B del encoder.
3. Realización de la modulación de anchura de pulsos del microcontrolador en el canal A y B del puente en H de la etapa de potencia a través de registros para elevar la frecuencia de modulación al menos a 20 kHz.
4. Generación y envío de datos.
5. Cálculo de la reductora.

2.1. Señales de entrada del encoder

El primer paso realizado fue conseguir poner en funcionamiento el motor realizando el conexiónado del puente en H con el Arduino Due. Este se realiza mediante los pines *IN1A*, *IN2A* y *enable* (activo a nivel alto), los cuales son los pines 5, 4 y 2 respectivamente. Inicialmente todos estos pines fueron declarados como salidas.

Seguidamente, el siguiente paso fue la lectura de las señales de entrada a través de las interrupciones hardware del microcontrolador.

Para poder calcular la posición del eje del motor, es necesario conocer el número de pulsos total que han leído las señales de entrada de ambos canales del encoder. Con el objetivo de calcular el número total de pulsos, se han creado 2 interrupciones, una para cada canal del encoder¹, que se activan cuando ocurre un evento en el flanco de subida o bajada. Cada interrupción, llama a una función la cual se encarga de incrementar o decrementar un pulso a *pulses_counter*, dependiendo del sentido de giro de nuestro motor (horario o antihorario).

```
attachInterrupt(digitalPinToInterrupt(portChannelA), channelA_ISR, CHANGE);

attachInterrupt(digitalPinToInterrupt(portChannelB), channelB_ISR, CHANGE);
```

A continuación, se detalla el procedimiento seguido para poder conocer el sentido de giro. El encoder magnético utilizado en *RealLabo* es un encoder en cuadratura, es decir, ambos canales tienen la misma forma de onda pero uno de ellos está atrasado/adelantado respecto al otro. Cuando se activa la interrupción del canal A, si el sentido de giro es horario, la señal del canal B es igual a la del canal A (Figura 2). En cambio, si el sentido es antihorario la señal del canal B será la inversa de la señal del canal A (Figura 3).

¹Los pines 3 y 7 han sido declarados como entradas para leer las señales de los canales del encoder.

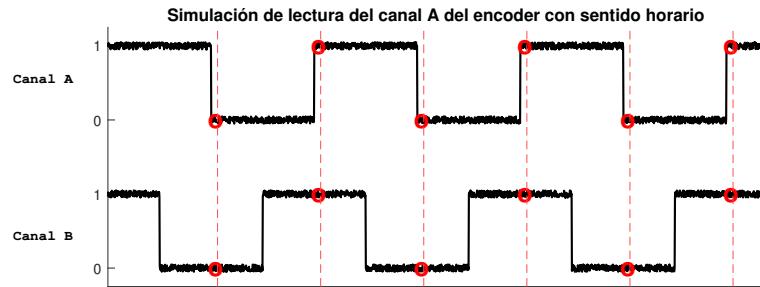


Figura 2: Simulación de lectura del canal A del encoder con sentido horario

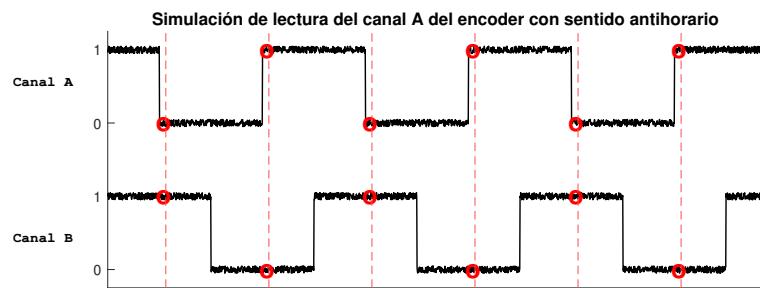


Figura 3: Simulación de lectura del canal A del encoder con sentido antihorario

De forma contraria a la interrupción del canal A, cuando se produce algún evento en el flanco de subida o bajada del canal B si el sentido de giro del motor es horario la señal del canal A es opuesta a la del canal B. Finalmente, si el sentido de giro es antihorario ambas señales de los encoders serán idénticas.

Por lo tanto, cuando el sentido de giro es horario se incrementa `pulses_counter` y si el sentido es antihorario se decrementa. Cabe destacar, que en el momento donde se inicia el sistema las señales de ambos encoders deben ser “0”. Para poder inicializar el encoder, se crea un booleano, `estado_inicial` que se pone a *true* cuando `pulses_counter` y las señales de los canales A y B son “0”. Una vez que el encoder está inicializado, cuando se produce una interrupción de los canales A o B, se modifica `pulses_counter` dependiendo del sentido de giro.

2.2. Diseño del PWM

En este apartado se describirá el procedimiento seguido para implementar el manejo de las señales de salida del controlador, mediante el PWM del microcontrolador en el canal A y B del puente en H de la etapa de potencia.

Para modificar la frecuencia de modulación del PWM al menos a 20 *KHz*, tenemos que conectar el periférico con el microcontrolador. En este caso se usaron los pines 35 y 37, que son los relacionados con dos de los canales de los ocho disponibles en el PWM Hardware del microcontrolador, el canal 0 y el 1. Además, se usó el periférico B y no se han declarado los pines 5 y 4 (*IN1A* y *IN2A*) que anteriormente habían sido declarados como salidas, ya que se deben conectar a los 2 canales elegidos del PWM Hardware.

La implementación del software para realizar la modulación por anchura de pulsos se ha realizado mediante la modificación de los registros del PWM Hardware [2]. Para ello, se han creado dos funciones, *configure_PWM* y *change_PWM*. La primera de ellas, se encarga de la configuración inicial de los registros, mediante un valor de frecuencia y ciclo de trabajo concretos. Por defecto, la frecuencia del PWM es de 20 kHz e inicialmente el motor se encuentra parado ($duty_cycle=0\%$). Posteriormente, para poder modificar correctamente el ciclo de trabajo de nuestro PWM se utiliza la función *change_PWM*, que únicamente modifica los registros que modifican el ciclo de trabajo, sin modificar el resto de registros. Por lo tanto, se han modificado los siguientes registros en la función *configure_PWM*:

- **PMC Write Protect Mode Register:** Registro que controla la protección contra escritura, el cual se ha modificado para deshabilitarlo.
- **PMC Peripherical Clock Enable Register 1:** Se encarga de habilitar los relojes de los periféricos. Se ha modificado para habilitar el reloj del periférico PWM Hardware.
- **PIO Peripheral AB Select Register:** Registro utilizado para seleccionar con qué periférico se va a trabajar. En este caso, hacer que los pines del PWM mencionados anteriormente sean controlados por el periférico B.
- **PIO Controller PIO Disable Register:** Modifica que pines dejan de ser controlados por el PIO y pasa a ser controlado por los periféricos, en este caso los pines anteriores del PWM Hardware.
- **PWM Enable Register:** Indica los canales del PWM activos. En nuestro caso los canales 0 y 1.
- **PWM Clock Register:** Registro que modifica el valor del reloj del periférico B, el cual se ha hecho para ser el reloj del sistema dividido por un factor de 4.
- **PWM Channel Mode Register:** Se encarga de controlar la configuración del PWM, en este caso modificado para los canales 0 y 1 (con la misma configuración).
- **PWM Channel Period Register:** Registro que controla el valor del periodo del PWM en función de una constante (*value_CPRD*) cuyo valor es:

$$value_CPRD = \frac{84 \times 10^6 [\text{Hz}]}{f [\text{Hz}]} \quad (1)$$

Donde, 84 MHz es la frecuencia de reloj del Arduino Due, y f es la frecuencia de modulación, en nuestro caso 20 kHz .

- **PWM Channel Duty Cycle Register:** Este registro introduce a cada canal utilizado el valor PWM deseado, el cual es el producto del valor del valor absoluto del *duty_cycle* por el *value_CPRD*.

$$pwm_level = |duty_cycle| \cdot value_CPRD \quad (2)$$

Donde $duty_cycle \in [0, 1]$ y *value_CPRD* es el valor calculado en la fórmula 1. Este valor se introduce en uno de ellos y en el otro un valor 0 en función de si el motor debe de girar en sentido horario o antihorario.

En la función *change_PWM* ha sido modificado el siguiente registro:

- **PWM Channel Duty Cycle Update Register:** Este registro realiza la misma función que el PWM Channel Duty Cycle Register.

Posteriormente, con la ayuda del osciloscopio se comprobó que la frecuencia de modulación del PWM era de 20 kHz. La Figura 4 muestra uno de los canales del PWM Hardware para una tensión de 5 V, es decir, aplicando la fórmula 3, un ciclo de trabajo del 41,67 %.

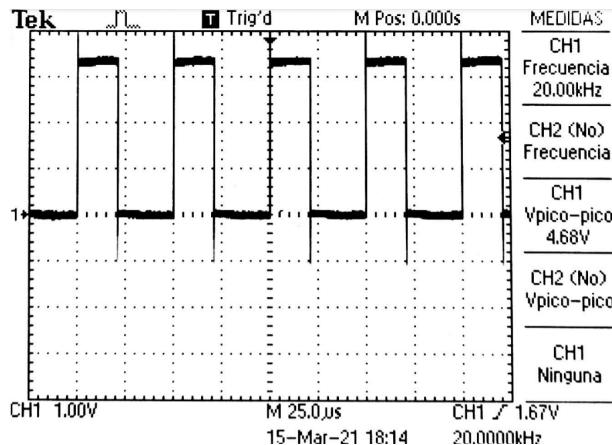


Figura 4: Modulación de anchura de pulsos vista desde el osciloscopio del *RealLabo*

2.3. Generación y envío de datos

Para poder recopilar los datos necesarios para el modelado del motor y los posteriores controladores, se ha configurado el Timer 3 para que genere una interrupción cada 1 ms. Esta interrupción llama a una función la cual almacena el valor de los pulsos que lleva contados en ese preciso instante en un array (buffer). Además, se suma uno a la variable *timer_counter* que representa el tiempo de muestreo.

La función de atención a la interrupción, es distinta para el caso del modelado que para los distintos controladores implementados. En el caso del modelado y como se explicará más adelante, se recogen 1200 muestras (1200 ms) y en el caso de los controladores se recogen 7000 muestras (7 s).

Una vez almacenadas todas las muestras en el array, la variable booleana *imprimir_pantalla* pasa a *true*. Si esta variable es *true*, en el *loop* se llama a una función *imprime*. Esta función, se encarga de imprimir los datos almacenados en el array por el puerto serial, con una velocidad de 115200 baudios para su recogida por el usuario.

2.4. Cálculo de la reductora

Con la intención de calcular la posición angular del motor DC, se ha calculado el valor de la reductora del motor. Para ello, se movió la rueda aproximadamente una vuelta, y se contó el número de pulsos del encoder, cuyo valor es 3509. Como sabemos que el encoder utilizado es de 48 CPR (*Cycles Per Revolution*), en una vuelta del motor (sin contar la reductora) se cuentan 48 pulsos. Por lo tanto, obtenemos una reductora de 73,1:1. Observando, las características del motor y el encoder utilizados comprobamos que la reductora del *RealLabo* es de **r = 74,83:1**, donde nos proporciona un total de 3591,84 pulsos por vuelta.

3. Modelado experimental de un motor DC con la arquitectura hardware y software implementadas

En esta sección, se ha realizado el modelado experimental del motor DC del *RealLabo*, con la arquitectura hardware comentada previamente y el software implementado, para contar el número de pulsos que se producen en los canales del encoder situado en eje del motor, donde:

$$q = 48 \text{ pulsos/vuelta}$$

Los experimentos realizados para modelar el motor DC han consistido en introducir una señal de entrada cuadrada periódica, durante $P = 10$ repeticiones, para 12 valores distintos de tensión de entrada ($Q = 12$). Las tensiones de entrada V_j voltios han sido los siguientes:

$$V \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

Por otro lado, la duración del escalón a nivel alto con una tensión de V_j voltios ha sido de 0,6 s y la del escalón a nivel bajo con 0 V, también ha sido de 0,6 s. Por último, el periodo de muestreo era de $T = 1 \text{ ms}$. En la Figura 5 se puede observar la señal de entrada:

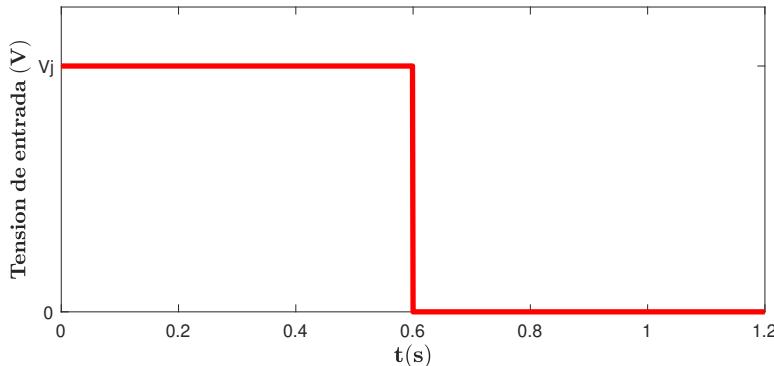


Figura 5: Señal de entrada cuadrada periódica

Cabe destacar, que la tensión máxima que admite el motor DC utilizado es de 12 V. Por lo tanto, si la tensión a la entrada es mayor que este valor, entra en saturación y la entrada es 12 V. En el software implementado, el parámetro de entrada de las funciones para modificar el PWM es el duty cycle. Por lo que, para cada valor de la tensión de entrada, se ha calculado el correspondiente duty cycle del motor con la fórmula 3:

$$\text{duty_cycle} = \frac{V_j}{12 \text{ V}} \cdot 100 [\%] \quad (3)$$

Una vez realizados todos los experimentos, 10 repeticiones de 12 experimentos con valores de entrada un escalón de diferente amplitud, se han promediado el número de pulsos de los 12 experimento en cada instante de tiempo, desde 0 ms hasta 1200 ms. En la Figura 6 se pueden observar las curvas del número de pulsos $N_j(t)$ promediadas del motor DC del *RealLabo* (Pololu 25 mm 12 V):

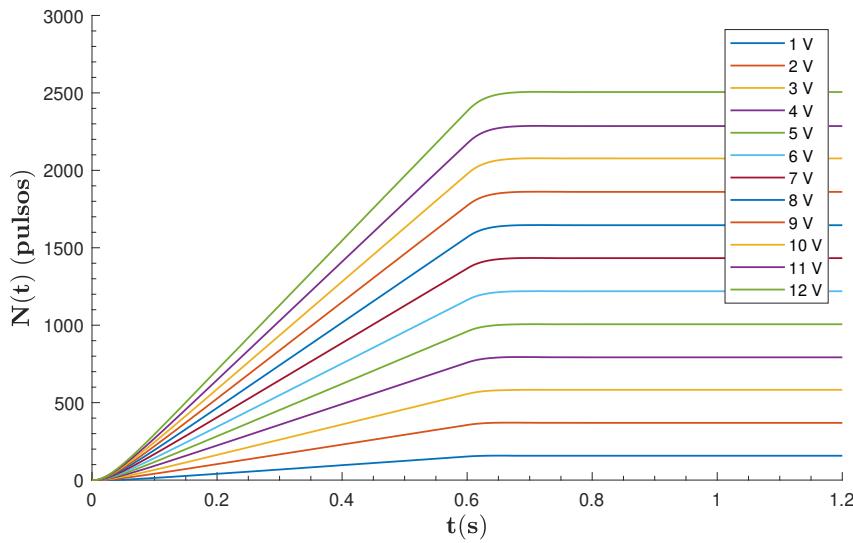


Figura 6: Número de pulsos en el eje del motor DC

Mediante la fórmula 4 podemos convertir los pulsos a la salida, en posición angular θ [rad].

$$\theta = \frac{\text{pulses} \cdot 2\pi}{3591,84} [\text{rad}] \quad (4)$$

En la Figura 7, podemos ver la posición angular media que gira el motor, cuando se introducen diferentes valores de la tensión de entrada V_j voltios.

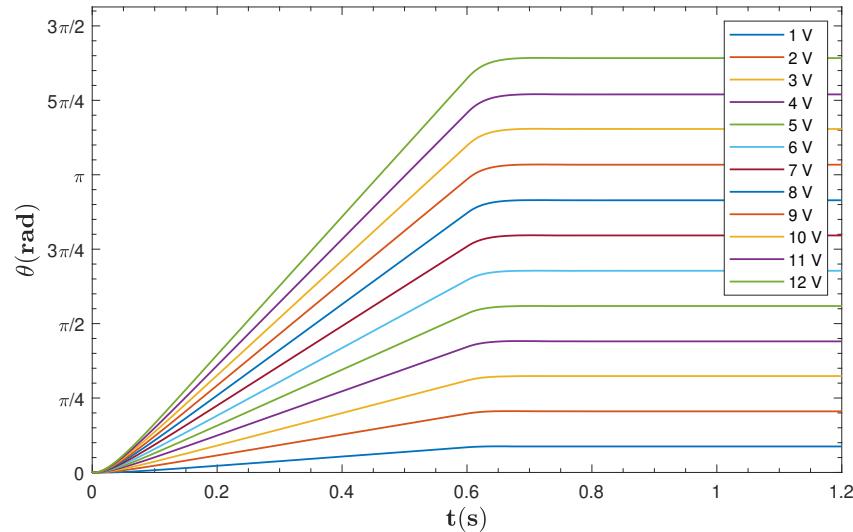


Figura 7: Posición en radianes a la salida del motor DC

Se puede observar, que si introducimos la tensión máxima que admite el motor a la entrada (12 V) durante 0,6 s el motor gira aproximadamente $3\pi/4$ [rad]. Mientras, que si introducimos solo 1 V, el motor apenas gira.

A continuación, se ha calculado la velocidad angular en el eje del motor DC del laboratorio, con la siguiente fórmula:

$$\dot{\theta}_j(t_k) = \frac{2\pi(N_j(t_k) - N_j(t_k - 1))}{q \cdot T} \quad (5)$$

Donde, $N_j(t_k)$ es el número de pulsos obtenido con el encoder, q es igual a 48 [pulsos/vuelta] y T es el periodo de muestreo, 0,001 [s]. En la Figura 8 se pueden ver las curvas de velocidad angular $\dot{\theta}_j(t_k)$ promediadas del motor DC del *RealLabo*:

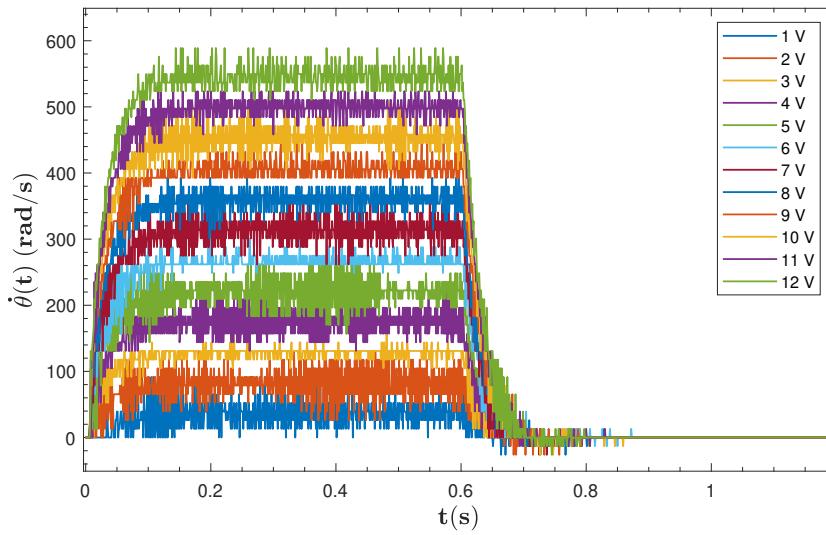


Figura 8: Velocidad angular en el eje del motor DC

Para la obtención experimental del modelo simplificado del motor, se debe obtener sus parámetros únicos p , K de la parte lineal del modelo, y la función no lineal f . Se ha seguido el algoritmo visto en clase [6], de cuatro fases mediante el programa de *Matlab* para el modelado de un motor DC [5]:

- **Primera fase:** Obtención del valor medio de velocidades angulares del régimen permanente.
- **Segunda fase:** Cálculo del polo medio de varianza mínima.
- **Tercera fase:** Cálculo de K y de la tensión de entrada equivalente.
- **Cuarta fase:** Identificación de la función no lineal del modelo del motor mediante la técnica de interpolación polinómica.

3.1. Fase 1: Obtención del valor medio de velocidades angulares del régimen permanente.

En esta primera fase, se ha calculado la velocidad angular en régimen permanente, $\bar{\dot{\theta}}_j(\infty)$, con la fórmula 6:

$$\bar{\dot{\theta}}_j(\infty) = \frac{\sum_{i=1}^N \dot{\theta}_{N_f-N+i}}{N} [\text{rad/s}] \quad (6)$$

Donde $N_f = 601$ ($t_f = 0,6\text{ s}$) y $N = 200$, es decir, se ha calculado el valor medio de los últimos $0,2\text{ s}$ del escalón.

3.2. Fase 2: Cálculo del polo medio de varianza mínima.

En esta fase, se realiza para cada uno de los 12 experimentos realizados utilizando el valor de $\bar{\theta}_j(\infty)$ obtenido en la fase anterior. A continuación, se calculará los polos en el transitorio de subida (pS_j) con la fórmula 7:

$$pS_j(k) = \frac{N_j(k) - N_j(k-1)}{T \left(\frac{q\bar{\theta}_j(\infty)}{2\pi} kT - N_j(k) \right)} [\text{s}^{-1}] \quad (7)$$

Por un lado, se encontrará el polo de subida a partir de calcular dicho polo a partir de distintos intervalos dentro del escalón. Para escoger el valor de pS_j , se escogerá el valor adecuado dentro del intervalo con menor varianza. Aplicando, los intervalos que se describen en el documento de *Modelado de SECO*, $k_{I_m} = 2$ ($t_{I_m} = 0,001\text{ s}$), $\Delta k = 60$ y $n = 2$ (600 primeras muestras). La Figura 9 muestra la varianza y el polo medio pS_j en cada intervalo de tiempo.

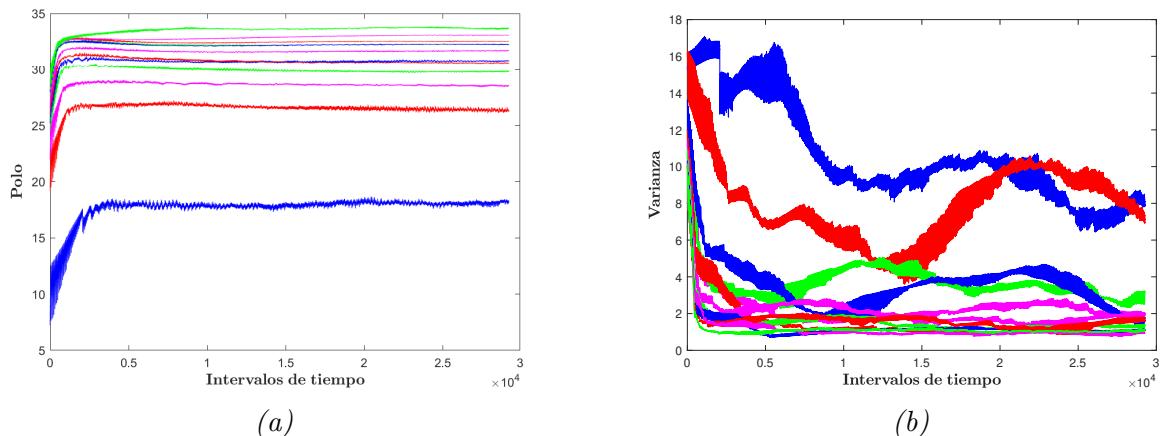


Figura 9: (a) Polo medio pS_j (b) Varianza para los diferentes intervalos de tiempo evaluados.

Por otro lado, para el cálculo de los polos en el transitorio de bajada (pB_j) se utilizará la fórmula 8, donde se calcula a partir de las muestras de final de subida ($k_0 = 601$) y un instante donde el motor se encuentra parado ($k_f = 1201$):

$$pB_j = \frac{q\bar{\theta}_j(\infty)}{2\pi} \frac{1}{N_j(k_f) - N_j(k_0)} [\text{s}^{-1}] \quad (8)$$

Finalmente, se calculará aplicando la fórmula 9 el polo de la función de transferencia como la media entre el polo de subida (pS_j) y el polo de bajada (pB_j) con $\alpha = 0,5$:

$$p_j = \alpha pS_j + (1 - \alpha)pB_j [\text{s}^{-1}] \quad (9)$$

En la Tabla 1 se muestran los valores de pS_j , pB_j y p_j calculados mediante *Matlab*:

V_j	pS_j	pB_j	p_j
1	17,91	54,63	36,27
2	26,53	45,78	36,15
3	28,61	36,23	32,42
4	30,15	39,44	34,79
5	30,76	36,13	33,45
6	30,60	35,88	33,24
7	31,61	35,80	33,71
8	32,25	35,40	33,83
9	32,35	34,92	33,64
10	32,56	34,27	33,41
11	33,10	33,75	33,43
12	33,25	33,61	33,43

Tabla 1: Polos de subida, bajada y medio

3.3. Fase 3: Cálculo de K y de la tensión de entrada equivalente.

En esta fase, primero se ha calculado el conjunto de valore K_j para cada experimento ($j \in \{1, 2, \dots, 12\}$ mediante la fórmula 10:

$$K_j = \frac{p_j \bar{\theta}_j(\infty)}{V_j} \left[\frac{\text{rad}}{V \cdot s^2} \right] \quad (10)$$

Además, se ha calculado la tensión equivalente V_{eq_j} para los 12 distintos experimentos con la fórmula 11:

$$V_{eq_j} = \frac{K_j V_j}{p_j} \cdot \frac{p}{K} [V] \quad (11)$$

En la Tabla 2, se muestra los valores de K_j y V_{eq_j} obtenidos mediante *Matlab*:

V_j	K_j	V_{eq_j}
1	1638,60	0,80
2	1633,28	1,83
3	1464,57	2,85
4	1571,90	3,88
5	1511,05	4,89
6	1501,61	5,93
7	1522,80	6,96
8	1528,17	7,98
9	1519,76	9,02
10	1509,43	10,05
11	1510,06	11,05
12	1510,27	12,10

Tabla 2: K_j y tensiones equivalentes de entrada (V_{eq_j})

Finalmente, se han calculado los valores de p y K del motor. El error cuadrático ha sido $J = 0,0701$.

$$\mathbf{p = 33,54}$$

$$\mathbf{K = 1515,10}$$

Por lo tanto, la función de transferencia del motor en velocidad (8) y posición angular (13), teniendo en cuenta la reductora, $r = 74,83 : 1$, será:

$$G_{\dot{\theta}_m} = \frac{K \cdot r}{s + p} = \frac{113374,93}{s + 33,54} \left[\frac{\text{rad}}{\text{V} \cdot \text{s}} \right] \quad (12)$$

$$G_{\theta_m} = \frac{1}{s} G_{\dot{\theta}_m} = \frac{K \cdot r}{s(s + p)} = \frac{113374,93}{s(s + 33,54)} \left[\frac{\text{rad}}{\text{V}} \right] \quad (13)$$

En la Figura 10 se representa en azul, las respuestas de la parte lineal del motor a las señales de entrada cuadrada, con los valores calculados del escalón de subida (pS_j) y bajada (pB_j). En rojo, la respuesta a las señales de entrada cuadrada, con el valor de p y K calculados. Por último, se han superpuesto en color verde las velocidades angulares medidas previamente, y representadas también en la Figura 8.

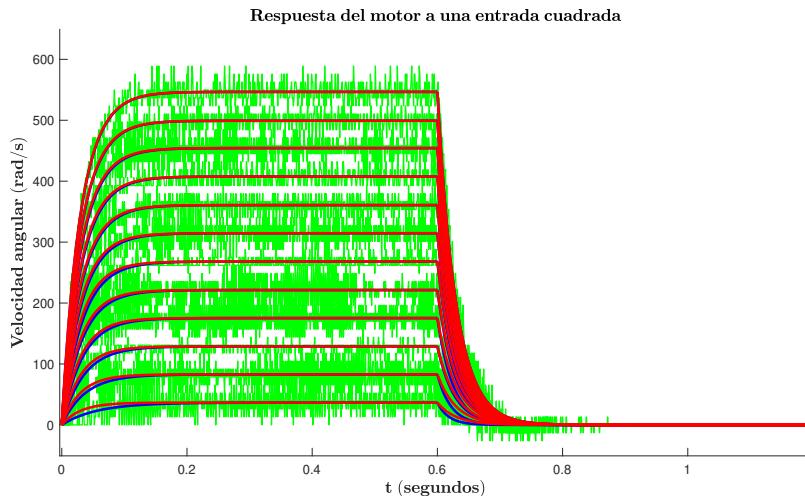


Figura 10: Respuesta del motor a una entrada cuadrada

3.4. Fase 4: Identificación de la función no lineal del modelo del motor mediante la técnica de interpolación polinómica.

Finalmente, la última fase consiste en identificar la función no lineal del motor del motor, f . Así, nuestro objetivo es encontrar dichos polinomios ($j \in \{1, 2, \dots, 12\}$):

$$V_{eq,j}(t) = f(V(t)) \quad (14)$$

$$V_j(t) = f^{-1}(V_{eq,j}(t)) \quad (15)$$

El polinomio, $V_{eq,j}$, y sus coeficientes serán los siguientes:

$$V_{eq,j} = a_1 V_j + a_3 V_j^3 + \dots + a_{23} V_j^{23} \quad (16)$$

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_{23} \end{bmatrix} = 10^{-20} \begin{bmatrix} 73114973702519971840,00 \\ 9336180225946857472,00 \\ -1739455712356216832,00 \\ 179597073613118016,00 \\ -11025273628171132,00 \\ 423622151169251,06 \\ -10495339798143,45 \\ 169400259493,80 \\ -1762230572,35 \\ 11346268,53 \\ -40986,58 \\ 63,32 \end{bmatrix} \quad (17)$$

Por último, los valores de los parámetros del polinomio f^{-1} son:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_{12} \end{bmatrix} = 10^{-20} \begin{bmatrix} 130757495275606671360,00 \\ -12422583637838217216,00 \\ 2528261971647230464,00 \\ -275715900488109600,00 \\ 17550823579963456,00 \\ -691244336456999,62 \\ 17417336661070,94 \\ -284276076622,34 \\ 2977259498,07 \\ -19231420,07 \\ 69496,02 \\ -107,14 \end{bmatrix} \quad (18)$$

La Figura 11 muestra los polinomios de interpolación impar de grado 23:

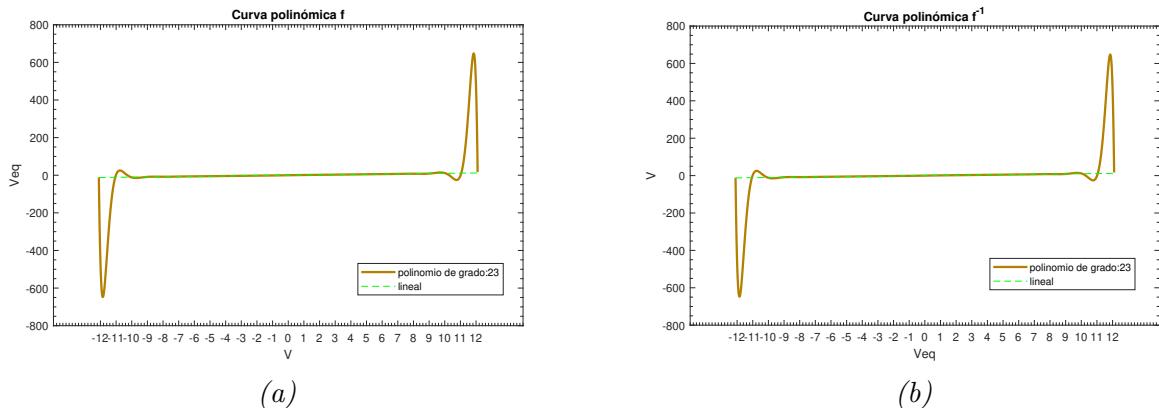


Figura 11: Polinomios de interpolación de grado 23 (a) f (b) f^{-1} .

4. Análisis, diseño e implementación de un controlador para un control de posición angular

Finalmente, en esta última sección se realizará el diseño e implementación de los 4 tipos de controladores vistos en clase: proporcional, proporcional derivativo, proporcional integral y proporcional integral derivativo. Para cada uno de ellos, se realizará el análisis matemático, mediante la tabla de Routh y el lugar de raíces. Posteriormente, en el *RealLabo* se implementará dicho controlador y se evaluará el comportamiento. Para evaluar el comportamiento, se tendrá en cuenta especificaciones de régimen permanente y de régimen transitorio (la sobreelongación máxima, M_p , el tiempo de establecimiento, t_s , el tiempo de pico, t_p y el tiempo de subida, t_r) para una entrada escalón.

El sistema a controlar es el motor modelado en la Sección 3 cuya función de transferencia, incluyendo la reductora, r , es:

$$G(s) = \frac{K \cdot r}{s(s + p)} = \frac{113375}{s(s + 33, 54)} \quad (19)$$

En la Figura 12 se muestra un sistema de control realimentado continuo con un controlador, $G_c(s)$:

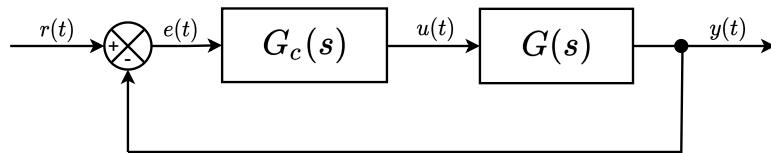


Figura 12: Estructura de control de lazo directo continuo con un controlador, $G_c(s)$.

La función de transferencia en lazo cerrado será:

$$H(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} \quad (20)$$

El funcionamiento del controlador, se basa en introducir un señal de referencia ($r(t)$), que en nuestro caso será un ángulo dado en radianes. La señal de error, $e(t)$, será la diferencia entre la señal de salida y la señal de referencia (fórmula 21). Finalmente, la señal de control, $U(s)$ como se observa en la fórmula 22 será el producto de $E(s)$ y $G_c(s)$.

$$e(t) = y(t) - r(t) \quad (21)$$

$$U(s) = E(s) \cdot G_c(s) \quad (22)$$

Todos los experimentos, se han realizado para un giro de 2π radianes, es decir, una vuelta completa. En la implementación del controlador se ha usado interrupciones periódicas en Timer 3, cada 1 ms.

4.1. Controlador Proporcional

Para el caso del controlador proporcional:

$$G_c(s) = K_p \quad (23)$$

Donde la señal de control, $u(k)$ en el caso discreto será:

$$u(k) = K_p \cdot e(k) \quad (24)$$

La Figura 13 muestra el sistema de control realimentado continuo mediante un controlador proporcional en lazo directo continuo.

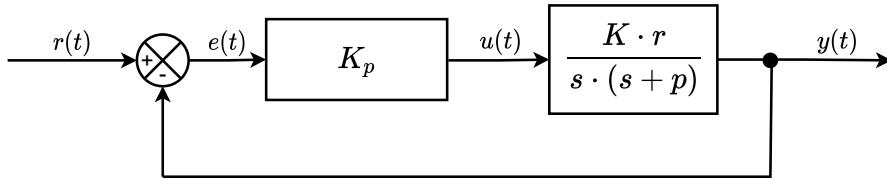


Figura 13: Controlador proporcional en una estructura de control de lazo directo continuo.

Por otro lado, la función de transferencia en lazo cerrado queda como:

$$H(s) = \frac{K_p Kr}{s^2 + ps + K_p Kr} \quad (25)$$

La Tabla de Routh es:

$$\begin{array}{c|cc} s^2 : & 1 & K_p \cdot K \cdot r \\ s^1 : & s \cdot p & \\ s^0 : & K_p \cdot K \cdot r & \end{array} \quad (26)$$

Observando la Tabla de Routh, y sabiendo que K , r y p son positivos, la condición que debe cumplir K_p para que el sistema sea estable es:

$$K_p > 0 \quad (27)$$

A continuación, conociendo los valores de K_p que hacen inestable el sistema, se ha representado en la Figura 14 el lugar de raíces de la función de transferencia en lazo abierto.

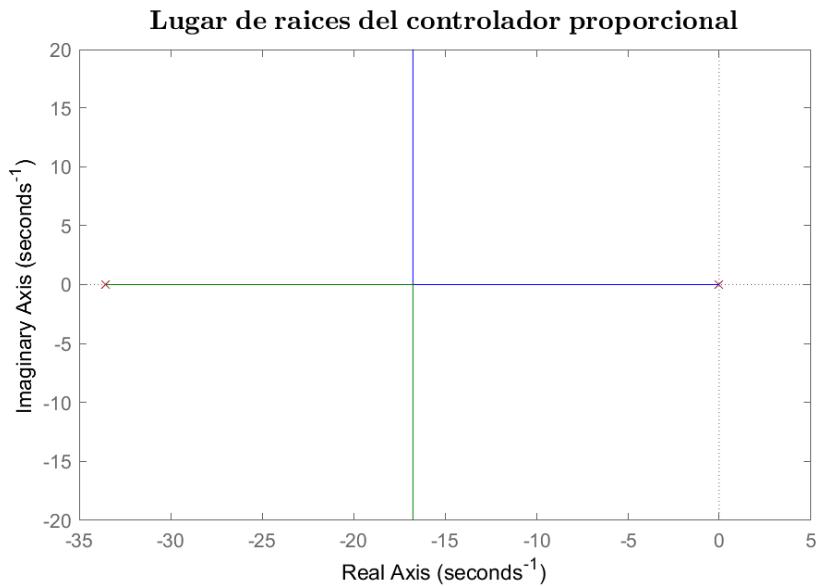


Figura 14: Lugar de raíces del controlador proporcional.

Se puede observar en el lugar de raíces de la Figura 14 que el sistema es estable, independientemente del valor de K_p elegido, como se ha explicado mediante la Tabla de Routh. Existen 2 polos reales, en $-33,54$ y en 0 . En la función de transferencia de lazo cerrado, a medida que va aumentando el valor de K_p , aumenta la componente imaginaria de los polos, y por lo tanto se comportan como polos complejos conjugados. Al producirse esto, se empiezan a producir oscilaciones en la señal de salida $y(t)$ y el sistema se comporta como sub-amortiguado.

A continuación, se ha representado la respuesta en lazo cerrado para distintos valores de K_p . Los valores elegidos de K_p han sido los siguientes:

$$K_p \in \{1, 3, 5, 10, 20, 50, 100\}$$

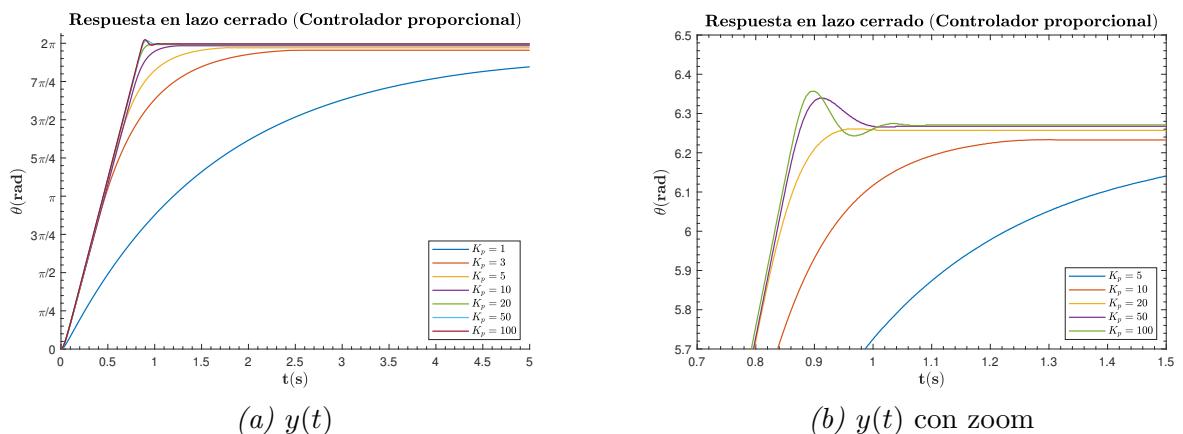


Figura 15: Señal de salida del controlador proporcional.

En la Tabla 3 se pueden encontrar los parámetros de los distintos experimentos realizados con el controlador proporcional:

K_p	t_r (ms)	M_p (rad)	t_p (ms)	t_s (ms)
1	-	-	-	-
2	-	-	-	1772
5	-	-	-	1189
10	-	-	-	914
20	-	-	-	836
50	882	0.0558	908	831
100	872	0.0733	894	828

Tabla 3: Parámetros de interés del controlador proporcional.

A vista de los distintos parámetros de la Tabla 3, se puede observar que a medida que aumenta el valor de K_p el tiempo de establecimiento, t_s y el tiempo de subida, t_r disminuyen. Sin embargo, la sobreexcitación máxima, M_p aumenta si K_p también aumenta. Por lo tanto, cuanto mayor es la ganancia proporcional, K_p más rápida es la respuesta.

Se puede observar, que con $K_p = 1$ no se consigue llegar al valor de referencia de 2π en 7 s, ya que existe un pequeño error de estado estacionario. Para los valores de $K_p \in \{1, 2, 5, 10, 20\}$ la respuesta del sistema es sobre-amortiguado. En cambio, para $K_p \in \{50, 100\}$ la respuesta es sub-amortiguada.

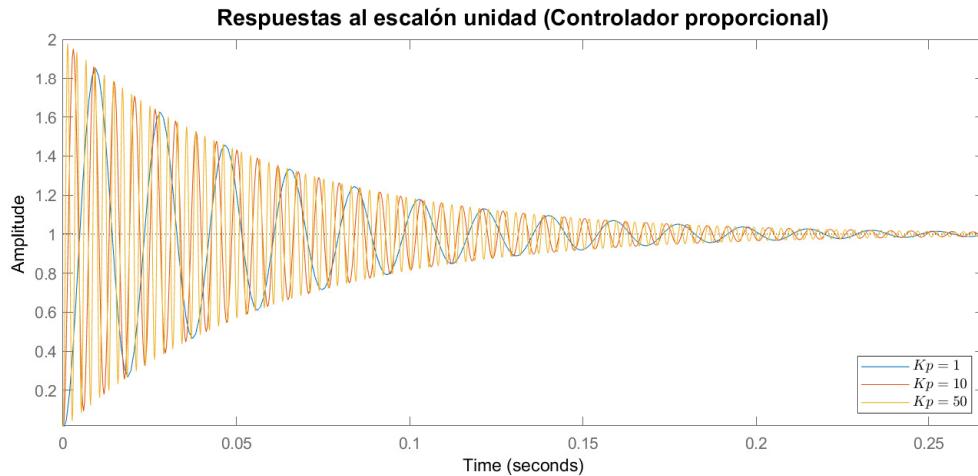


Figura 16: Respuestas al escalón unidad del controlador proporcional variando K_p .

En la Figura 16 se muestran las respuestas al escalón unidad variando K_p para el sistema formado por el controlador proporcional.

Por lo tanto, cuanto más grande sea el error, $e(t)$ más rápida será la respuesta, y conforme el error se vaya reduciendo la señal de control será más pequeña, hasta que el error sea 0.

4.2. Controlador Proporcional Derivativo

Para el controlador proporcional derivativo:

$$G_c(s) = K_p(1 + \tau_d s) \quad (28)$$

En la Figura 17 se muestra el sistema de control realimentado continuo mediante el controlador proporcional derivativo en lazo directo en cuestión.

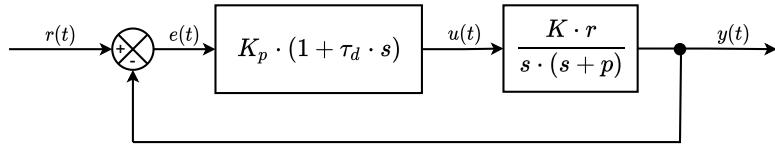


Figura 17: Controlador proporcional derivativo en una estructura de control de lazo directo continuo.

La acción de control del controlador proporcional derivativo (PD) es:

$$u(t) = K_p \left(e(t) + \tau_d \frac{de(t)}{dt} \right) \quad (29)$$

Por otro lado, la señal de control, $u(k)$ en el dominio de tiempo discreto será:

$$u(k) = K_p \cdot e(k) + K_p \cdot \tau_d \cdot \dot{e}(k) \quad (30)$$

Donde \dot{e} es la derivada del error, y se calcula mediante la fórmula 31:

$$\dot{e}(k) = \frac{e(k) - e(k-1)}{T} \quad (31)$$

Y su función de transferencia es la siguiente:

$$H(s) = \frac{K_p K r (1 + \tau_d s)}{s^2 + (p + K_p K r \tau_d) s + K_p K r} \quad (32)$$

Cuya tabla de Routh es:

$$\begin{array}{c|cc}
s^2 : & 1 & K_p \cdot K \cdot r \\
s^1 : & p + K_p \cdot K \cdot \tau_d \cdot r & \\
s^0 : & K_p \cdot K \cdot r &
\end{array} \quad (33)$$

Por lo tanto, sabiendo que K , r y p son positivos, las condiciones que deben cumplir K_p y τ_d son:

$$K_p > 0 \quad (34)$$

$$\tau_d > \frac{-p}{K_p K r} \quad (35)$$

Sabiendo los valores que hacen el sistema estable, se pasa a representar en la Figura 18 el lugar de raíces de la función de transferencia en lazo abierto. Se ha usado $K_p = 10$ y se ha ido variando el valor de τ_d elegido.

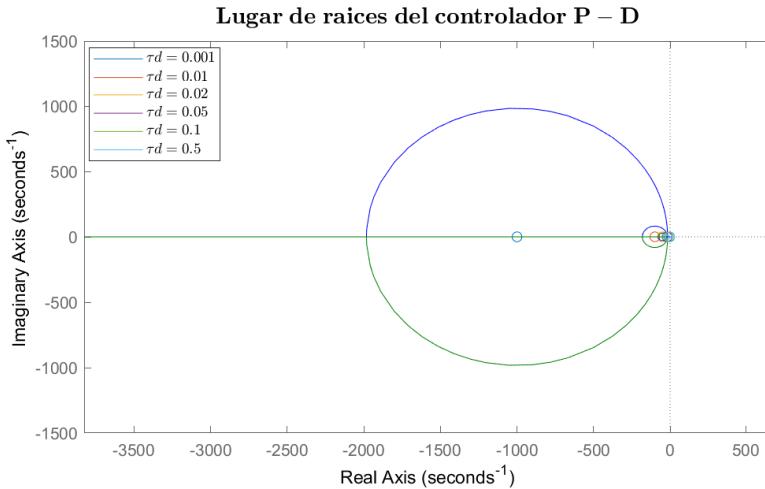


Figura 18: Lugar de raíces del controlador P-D para distintos valores de τ_d .

Observando el lugar de raíces de la Figura 18, se observa con respecto al caso anterior del controlador proporcional como se ha añadido un cero. Al añadir un cero, cerca del origen se hace más estable el sistema. Cuando se aumenta el valor de τ_d el efecto derivador del sistema también aumenta.

Posteriormente, se ha representado la señal de salida del controlador proporcional para los siguientes valores:

$$K_p \in \{10, 100\}$$

$$\tau_d \in \{0,001; 0,01; 0,002; 0,05; 0,1; 0,5\}$$

La Figura 19 muestra la señal de salida del controlador proporcional derivativo para $K_p = 10$ variando el valor de τ_d , y en la Tabla 4 se muestran sus principales parámetros.

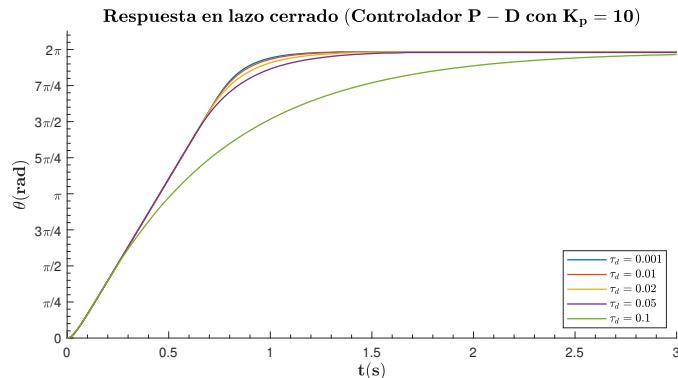
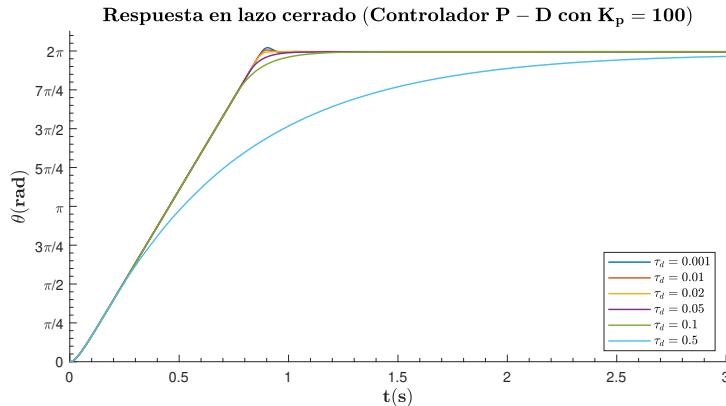
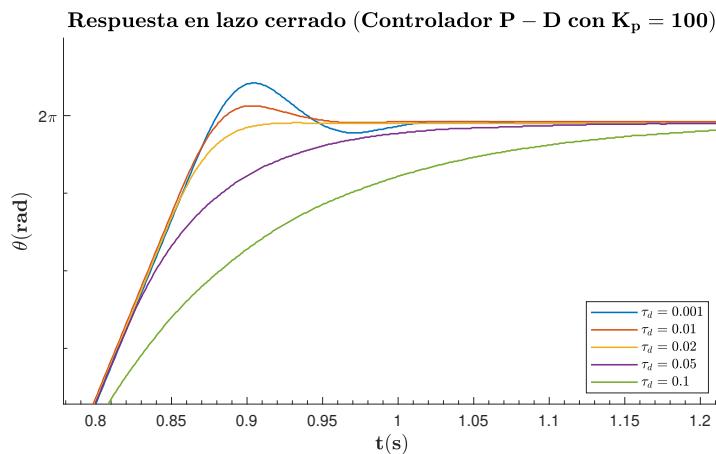


Figura 19: Señal de salida del controlador proporcional derivativo para $K_p = 10$ variando τ_d .

τ_d	t_r (ms)	M_p (rad)	t_p (ms)	t_s (ms)
0,01	-	-	-	913
0,02	-	-	-	930
0,05	-	-	-	983
0,1	-	-	-	1080
0,5	-	-	-	2084

Tabla 4: Parámetros de interés del controlador P-D con $K_p = 10$ variando τ_d .

Por otro lado, en la Figura 20 y 21 se muestran la señal de salida al igual que en el caso anterior variando τ_d , pero con $K_p = 100$. Finalmente, en la Tabla 5 se muestran sus parámetros de interés.

Figura 20: Señal de salida del controlador proporcional derivativo para $K_p = 100$ variando τ_d .Figura 21: Señal de salida con zoom del controlador proporcional derivativo para $K_p = 100$ variando τ_d .

τ_d	t_r (ms)	M_p (rad)	t_p (ms)	t_s (ms)
0,001	881	0.0663	903	836
0,01	887	0.0191	898	834
0,02	-	-	-	835
0,05	-	-	-	839
0,1	-	-	-	881
0,5	-	-	-	2084

Tabla 5: Parámetros de interés del controlador P-D con $K_p = 100$ variando τ_d .

A vista de los resultados, cuanto mayor sea el valor de τ_d más lento será el sistema, siendo el tiempo de establecimiento, t_s mucho mayor. La respuesta del sistema es sobre-amortiguada para todos los valores con los que se han realizado las simulaciones en el *RealLabo*, a excepción de τ_d igual a 0,001 y 0,01 para $K_p = 100$ que es sub-amortiguada.

La sobreenlogación máxima es mayor para el caso de $\tau_d = 0,001$ que para $\tau_d = 0,01$, debido a que a medida que aumenta τ_d la sobreenlogación máxima disminuye.

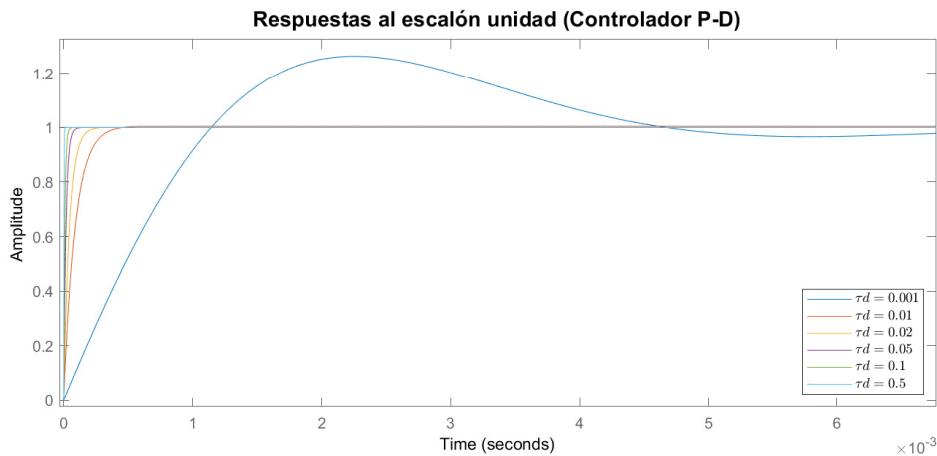


Figura 22: Respuestas al escalón unidad del controlador proporcional derivativo con $K_p = 100$ y variando τ_d .

En la Figura 22 se muestran las respuestas al escalón unidad con $K_p = 100$ y variando τ_d para el sistema formado por el controlador proporcional derivativo.

En conclusión, la función del controlador proporcional derivativo es ralentizar la señal de salida del sistema. Así, cuanto más cerca esta del valor de referencia, se hace más lento.

4.3. Controlador Proporcional Integral

Para el controlador proporcional integral:

$$G_c(s) = K_p \left(1 + \frac{1}{\tau_i s} \right) \quad (36)$$

En la Figura 23 se muestra el sistema de control realimentado continuo mediante el controlador proporcional integrador en lazo directo.

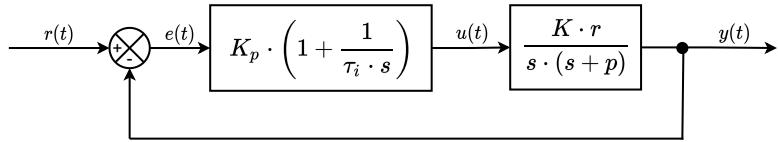


Figura 23: Controlador proporcional integral en una estructura de control de lazo directo continuo.

En este caso, el término integral es proporcional a la magnitud y a la duración del error. La acción de control viene del controlador proporcional integral (PI) por:

$$u(t) = K_p \left(e(t) + \tau_i \int_0^t e(t) dt \right) \quad (37)$$

Esta acción de control en tiempo discreto es:

$$u(k) = K_p \cdot e(k) + \left(\frac{K_p}{\tau_i} \right) \cdot \int_0^t e(k) dt \quad (38)$$

Donde la integral del error, se puede calcular a partir de:

$$\int_0^t e(k) dt = \left(\sum_0^t e(k) \right) \cdot T \quad (39)$$

Por otro lado, su función de transferencia es:

$$H(s) = \frac{K_p Kr (1 + \tau_i s)}{\tau_i s^3 + \tau_i p s^2 + \tau_i K_p K r s + K_p K r} \quad (40)$$

Cuya tabla de Routh como la siguiente:

$$\begin{array}{c|cc}
s^3 : & \tau_i & \tau_i \cdot K_p \cdot K \cdot r \\
s^2 : & \tau_i \cdot p & K_p \cdot K \\
s^1 : & K_p \cdot K \cdot r \cdot (\tau_i - \frac{1}{p}) & \\
s^0 : & K_p \cdot K \cdot r &
\end{array} \quad (41)$$

Al igual que en los controladores anteriores se va a estudiar el lugar de raíces (Figura 24). Se puede observar, que a medida que se va aumentando el valor de τ_i las ramas (verde y roja) se van desplazando hacia la izquierda. Además, aparece un polo en el origen, junto a un cero. Cuando aumentamos el valor de τ_i el efecto integrador del sistema disminuye.

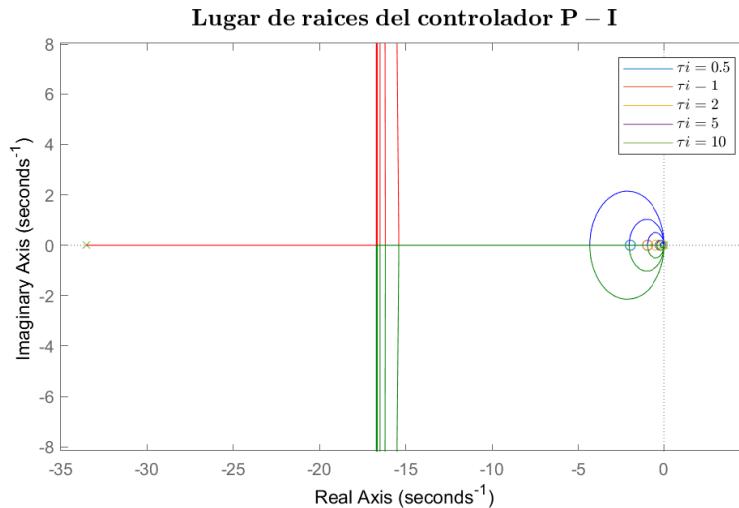


Figura 24: Lugar de raíces del controlador proporcional integral

En la Figura 25 se puede observar la salida del controlador proporcional integral con un valor de $K_p = 10$ con variación en los valores de τ_i y en la Tabla 6 los principales parámetros.

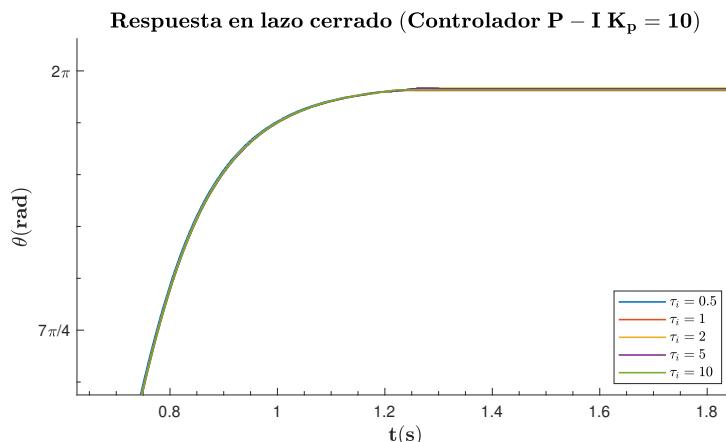


Figura 25: Señal de salida del controlador proporcional derivativo para $K_p = 10$ y variando τ_i .

τ_i	t_r (ms)	M_p (rad)	t_p (ms)	t_s (ms)
0,5	-	-	-	893
1	-	-	-	897
2	-	-	-	895
5	-	-	-	897
10	-	-	-	896

Tabla 6: Parámetros de interés del controlador P-I con $K_p = 10$ variando τ_i .

Y a continuación, en la Figura 26 se observan la salida del controlador proporcional integral con un valor de $K_p = 100$ con variación en los valores de τ_i y en la Tabla 7 los principales parámetros.

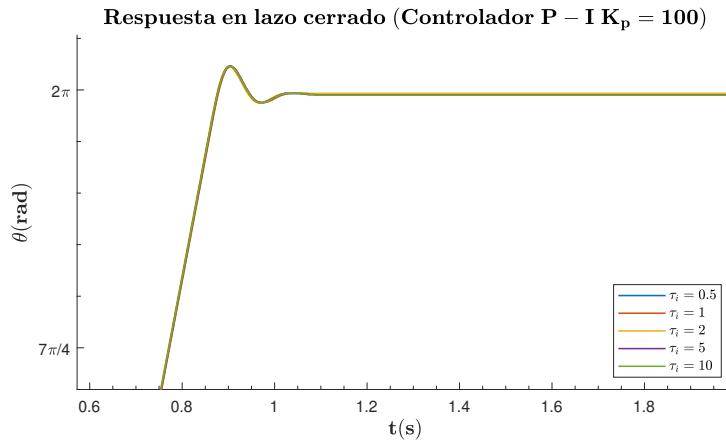


Figura 26: Señal de salida del controlador proporcional derivativo para $K_p = 100$ y variando τ_i .

τ_i	t_r (ms)	M_p (rad)	t_p (ms)	t_s (ms)
0,5	901	0.0715	901	835
1	903	0.0715	903	836
2	899	0.0715	899	833
5	902	0.0733	902	834
10	900	0.0715	900	834

Tabla 7: Parámetros de interés del controlador P-I con $K_p = 10$ variando τ_i .

A la vista de los resultados anteriores, no se puede llegar a una conclusión clara debido a su similitud, por lo que pasamos a representar la respuesta al escalón unidad para poder determinar el funcionamiento de este controlador. En la Figura 27 se muestran las respuestas al escalón unidad con $K_p = 100$ y variando τ_i para el sistema formado por el controlador proporcional integral.

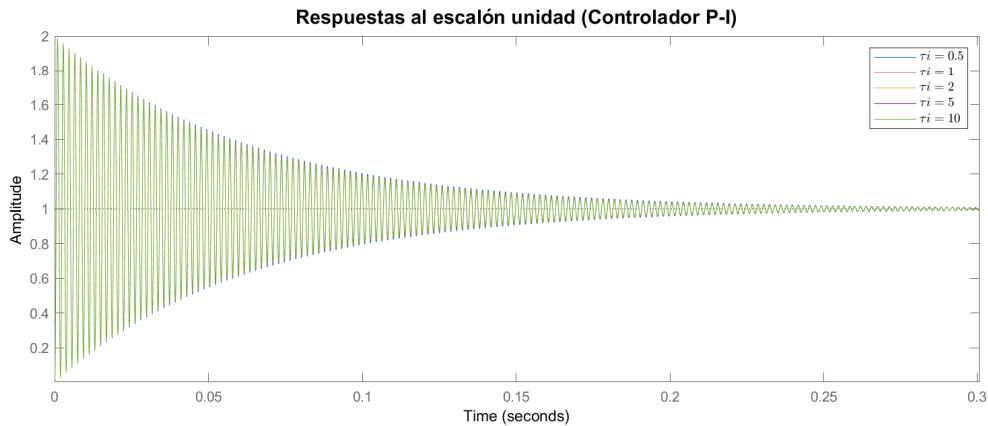


Figura 27: Respuestas al escalón unidad del controlador proporcional integral con $K_p = 100$ y variando τ_i .

A vista de los resultados, a medida que τ_i desciende, el controlador se hace más integrador, y por lo tanto el tiempo de subida baja. En cambio, la sobreexcitación máxima y el tiempo de establecimiento aumentan. Al usar el controlador integral junto al proporcional se aumenta la rapidez del sistema, lo que permite llegar antes al valor deseado, es decir, menor tiempo de respuesta.

4.4. Controlador Proporcional Integral Derivativo

Finalmente, se estudiará el caso de los tres anteriores combinados (Figura 28, donde se suman los términos proporcional, integral y derivativo por lo que:

$$G_c(s) = K_p(1 + \tau_d s + \frac{1}{\tau_i s}) \quad (42)$$

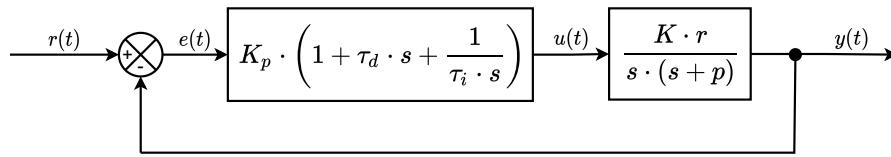


Figura 28: Controlador proporcional integral derivativo en una estructura de control de lazo directo continuo

Para el controlador PID, la acción de control es:

$$u(k) = K_p \cdot e(k) + K_p \cdot \tau_d \cdot \dot{e}(k) + \left(\frac{K_p}{\tau_i}\right) \cdot \int_0^t e(k) dt \quad (43)$$

Donde, la derivada del error, $\dot{e}(k)$ y la integral del error, $\int_0^t e(k) dt$ son las calculadas en la fórmula 31 y 39. Por lo que la función de transferencia en lazo cerrado queda como:

$$H(s) = \frac{K_p Kr (1 + \tau_i s + \tau_d \tau_i s^2)}{\tau_i s^3 + (\tau_i p + K_p Kr \tau_d \tau_i) s^2 + \tau_i K_p Kr s + K_p Kr} \quad (44)$$

Su tabla de Routh es:

$$\begin{array}{c|cc} s^3 : & \tau_i & \tau_i \cdot K_p \cdot K \cdot r \\ s^2 : & \tau_i \cdot p & K_p \cdot K \\ s^1 : & K_p \cdot K \cdot r \cdot \left(\tau_i - \frac{1}{p + K_p Kr \tau_d}\right) & \\ s^0 : & K_p \cdot K \cdot r & \end{array} \quad (45)$$

Al igual que en los casos anteriores, para que el sistema sea estable si consideramos que K_p y τ_d son positivas, la condición que debe cumplir τ_i es:

$$\tau_i > \frac{1}{p + K_p Kr \tau_d} \quad (46)$$

El lugar de raíces de la Figura 29 muestra 3 ramas (3 polos en lazo cerrado). Si se aumenta, el valor de alguna de las constantes los polos del sistema se hacen más negativo y por lo tanto el sistema es más estable. Como se ha explicado previamente, si aumentamos τ_d aumenta el efecto derivador y si se aumenta τ_i disminuye el efecto integrador.

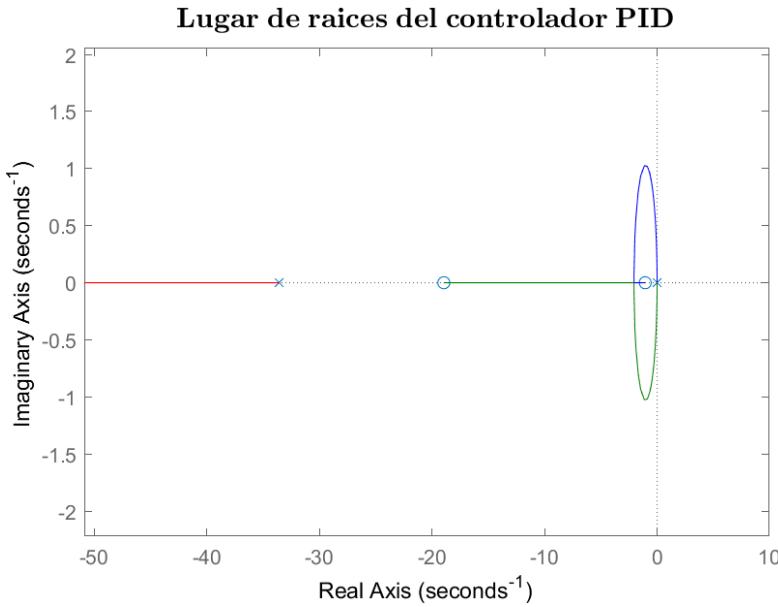


Figura 29: Lugar de raíces del controlador proporcional integral derivativo

La Figura 30 muestra la señal de salida del controlador PID para distintos valores de τ_d y τ_i siendo $K_p = 100$. Cabe destacar que debido a la inexperiencia diseñando controladores, los valores se han escogido en función de los otros controladores analizados previamente. Al igual que en los experimentos anteriores, la variación de los valores de τ_d afecta más al tiempo de establecimiento que los de τ_i .

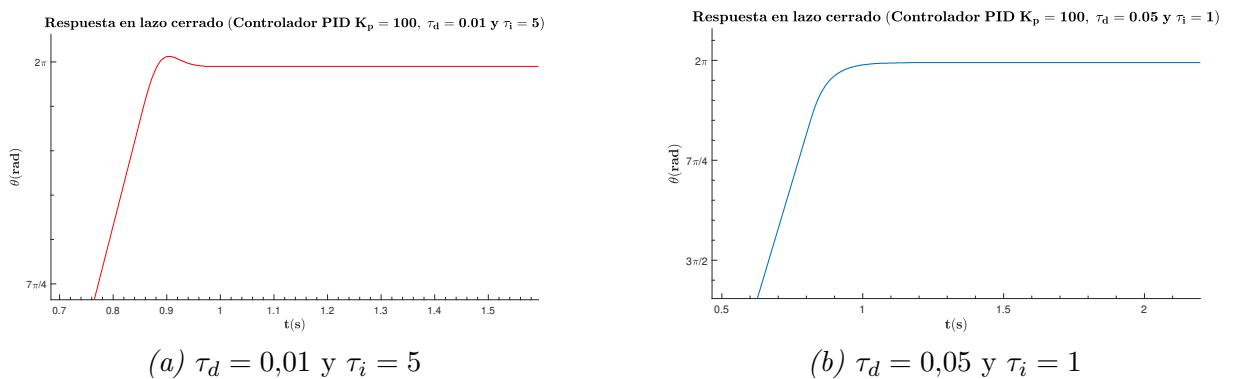


Figura 30: Señal de salida del controlador proporcional integral derivativo con $K_p = 100$.

En la Figura 31 se muestra la respuesta al escalón unidad con $K_p = 100$, $\tau_d = 0,05$ y $\tau_i = 1$ para el sistema formado por el controlador proporcional integral derivativo.

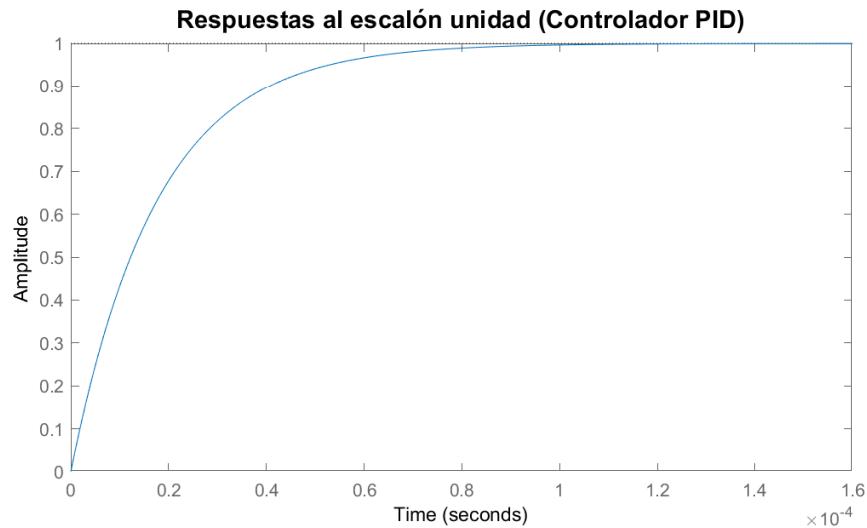


Figura 31: Respuestas al escalón unidad del controlador proporcional integral derivativo.

5. Conclusiones

En este trabajo, hemos podido modelar el motor disponible en el *RealLabo* e implementar distintos controladores para el control de su posición angular. A lo largo de este proyecto se ha aprendido bastante sobre sistemas sistemas de control y se ha puesto en práctica los conocimientos vistos en clase de modelado de un motorDC y análisis. Sin embargo, nos hemos enfrentado a diversos problemas, como ha podido ser la implementación del PWM mediante registros, la recogida de datos a través del puerto serie o conseguir que el Timer interrumpa correctamente.

Además, hemos llegado a las siguientes conclusiones:

- Cuando se produce la interrupción periódica del Timer cada 1 ms, para almacenar el número de pulsos en el eje del motor es muy importante que no se vea interrumpido por ningún otro proceso. Además, es necesario no imprimir estos datos por el puerto serie, cada milisegundo, sino almacenarlos correctamente en un array e imprimirlos cuando estén todos los datos recogidos, dado que este no puede alcanzar tal velocidad.
- Aunque todavía no se ha aprendido a diseñar controladores, mediante las herramientas de análisis aprendidas (Tabla de Routh y lugar de raíces) se ha podido analizar la respuesta en lazo cerrado y en lazo abierto de los controladores, de forma teórica y en el *RealLabo* para ver la evolución de las distintas especificaciones de régimen permanente y transitorio, demostrándonos así la utilidad de estas herramientas.
- Se ha observado, que para algunos valores de K_p pequeños no se conseguía superar el par de fricción inicial del motor. Por consiguiente, el motor no se movía. Esto nos demostraba que aunque realicemos aproximaciones teóricas, en el laboratorio se deben de seguir teniendo en cuenta todos los parámetros.
- En cuanto a los controladores, hemos podido comprender que se consigue una respuesta más rápida aumentando el valor de K_p en el caso del controlador proporcional, disminuyendo el valor de τ_d en el caso del controlador diferencial o aumentando el valor de τ_i en el caso del controlador integral.
- En lo relacionado al controlador proporcional integral derivativo, se ha visto que es el más adecuado, ya que podemos modificar los diversos parámetros para diseñar el controlador según nuestras necesidades. Por esta razón, es el más usado en la actualidad.

Referencias

- [1] Información Arduino DUE. [Disponible online] Accedido por última vez el 30/03/2021.
- [2] ATMEL SAM3X8E ARM CORTEX-M3. *Datasheet*. Microcontroller. [Disponible online] Accedido por última vez el 30/03/2021.
- [3] DMOS DUAL FULL BRIDGE DRIVER. *L6206*. [Disponible online] Accedido por última vez el 07/03/2021.
- [4] DUAL BRUSH DC MOTOR DRIVER EXPANSION BOARD BASED ON L6206 FOR STM32 NUCLEO. *X-NUCLEO-IHM04A1* . [Disponible online] Accedido por última vez el 15/03/2021.
- [5] MONASTERIO-HUELIN F. *simuladoModeladoMotor*. Programas Matlab para el modelado de un motor DC. [Disponible online] Accedido por última vez el 26/03/2021.
- [6] MONASTERIO-HUELIN F., GUTIÉRREZ A. *Modelado de un motor DC*. 2021. [Disponible online] Accedido por última vez el 30/03/2021.
- [7] POLOLU. ROBOTICS & ELECTRONICS. *Datasheet Motor 25 mm 12 V. Low Power*. [Disponible online] Accedido por última vez el 17/03/2021.
- [8] POLOLU. ROBOTICS & ELECTRONICS. *Magnetic Encoder. 75:1 Metal Gearmotor 25Dx66L mm LP 12V with 48 CPR Encoder*. [Disponible online] Accedido por última vez el 21/03/2021.