

# Diseño de Reguladores para el Balanceo y Estabilización de un Péndulo Invertido

Implementación en *Arduino* con *MATLAB* y *Simulink*

**Alejandro Casanova Martín**

Ingeniería Electrónica Industrial y Automática



Escuela Técnica Superior de Ingeniería y Diseño Industrial

Universidad Politécnica de Madrid

España

Diciembre de 2020

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Sistema Físico a Controlar . . . . .	2
1.2. Herramientas Empleadas . . . . .	3
<b>2. Modelado Matemático de la Planta</b>	<b>4</b>
2.1. Modelado del Péndulo Invertido . . . . .	4
2.2. Modelado del Motor de CC y el Sistema de Tracción del Carro . . . . .	6
2.3. Modelo del Conjunto . . . . .	10
<b>3. Diseño del Regulador</b>	<b>10</b>
3.1. PID Clásico . . . . .	10
3.2. Asignación de Polos por Realimentación del Estado . . . . .	13
3.2.1. Asignación Directa . . . . .	14
3.2.2. Regulador Óptimo Cuadrático (LQR) . . . . .	15
3.2.3. Servocontrolador Óptimo . . . . .	18
<b>4. Diseño del Observador</b>	<b>20</b>
4.1. Diseño de Observador por Asignación Directa de Polos . . . . .	20
4.2. Diseño de Observador Óptimo - Filtro de Kalman . . . . .	22
<b>5. Controlador para el Balanceo del Péndulo (Swing-Up)</b>	<b>23</b>
<b>6. Demo - Paso a Paso</b>	<b>25</b>
<b>7. Implementación de los Reguladores</b>	<b>26</b>
7.1. PID Clásico . . . . .	26
7.2. Regulador por Asignación Directa de Polos y Estimador de Kalman . . . . .	28
7.3. Servocontrolador Lineal Cuadrático Gaussiano con Acción Integral (LQG) . . . . .	30
7.4. Swing-Up y Servocontrolador LQG . . . . .	31
<b>8. Conclusiones</b>	<b>32</b>

# 1. Introducción

La meta de este trabajo es poner en práctica los conocimientos adquiridos en las asignaturas de Regulación Automática e Ingeniería de Control. Para lograr la estabilización de un péndulo invertido se implementarán distintos tipos de reguladores, recurriendo a conceptos y recursos de tanto la teoría de control moderna, como de la clásica. Se analizarán las fortalezas de ambas y se compararán los resultados obtenidos con cada una.

Para la ejecución del trabajo se hará uso extensivo de la herramienta de cálculo numérico *MATLAB*, así como de *Simulink* y las diferentes toolbox disponibles para identificación, modelado y diseño de sistemas de control. A pesar de que la mayoría de los cálculos serán realizados computacionalmente, se tratará de explicar y justificar cada concepto empleado y resultado obtenido.

## Comentarios y Aclaraciones

La memoria se dividirá principalmente en tres partes: modelado, diseño e implementación. En la parte de diseño se abordarán tanto los reguladores como los observadores individualmente, estudiando su comportamiento mediante simulaciones; mientras que en la parte de implementación se combinarán los diferentes reguladores y observadores y se compararán los resultados experimentales.

Es importante señalar que en muchos casos hubo que reajustar los reguladores a la hora de implementarlos en la planta real. La mayoría de las veces se obtuvo mejores resultados aumentando las ganancias de los reguladores obtenidos mediante simulación, probablemente debido a las simplificaciones realizadas en el modelado.

## Videos Demostrativos:

PID clásico y Servoregulador LQG: <https://youtu.be/Yro0ylzs2SM>

Balanceo y Estabilización con LQG: <https://youtu.be/7lcDM7xUucg>

## Todo el Código Disponible en:

<https://github.com/Alejandro-Casanova/Ingenieria-de-Control---Pendulo-Invertido>

### 1.1. Sistema Físico a Contrólar

La planta está formada principalmente por el raíl de una impresora de inyección de tinta. A lo largo del raíl se mueve la base sobre la que se ha montado el péndulo, formado por una barra de aluminio con una masa adherida al extremo. Un motor de corriente continua transmite el movimiento a la base (carro a partir de ahora) a través de una cinta.

Para medir la posición lineal del carro, se ha empleado un encoder óptico incremental obtenido de la impresora, mientras que para registrar la posición angular del péndulo se ha incorporado un encoder incremental rotatorio. El microcontrolador empleado será un *Arduino Mega 2560*, que nos permitirá implementar dos rutinas de interrupción externa por

cada encoder para lograr una mayor resolución. Para controlar el motor se ha empleado un driver *Cytron* de 13 Amperios, alimentado por una fuente ATX.

$N^{\circ}$	<i>Componente</i>
1	Arduino Mega 2560
2	Encoder Óptico Incremental Lineal (1850 pulsos)
3	Encoder Óptico Incremental Rotatorio (600 PPR)
4	Controlador Motor DC Cytron 13 A
5	Fuente de Alimentación ATX 3,3-12 V
6	Péndulo y Masa
7	Guía y Carro de impresora (32 cm de carrera)

Tabla 1: Resumen de Componentes.

## 1.2. Herramientas Empleadas

1	<i>MATLAB &amp; Simulink</i> 2020b
2	<i>Simulink</i> Control Design Toolbox
3	Control System Toolbox
4	<i>MATLAB &amp; Simulink</i> Support Package for <i>Arduino</i> Hardware
5	Rensselaer <i>Arduino</i> Support Package Library
6	<i>Simulink</i> Desktop Real-Time
6	<i>Arduino</i> IDE

Tabla 2: Resumen de Herramientas Utilizadas.

## 2. Modelado Matemático de la Planta

### 2.1. Modelado del Péndulo Invertido

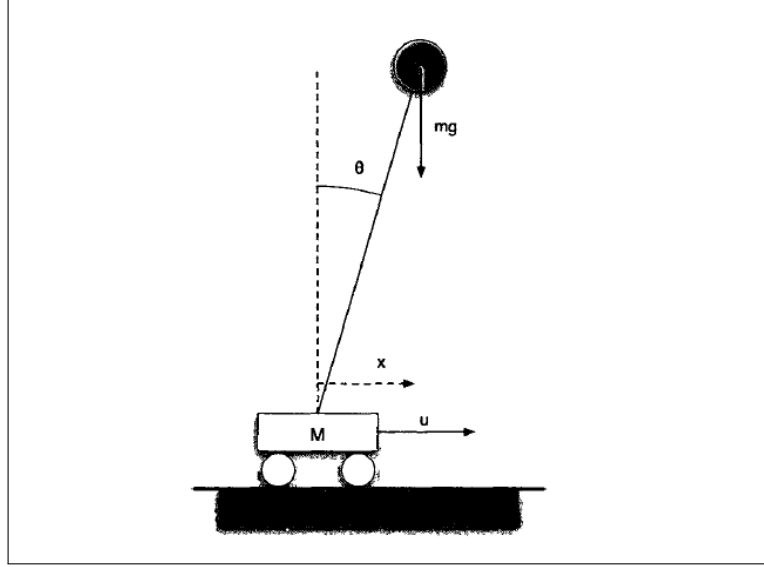


Figura 1: Diagrama del péndulo invertido a modelar.[1]

Como se puede observar en la Figura 1, el sistema físico está formado por las masas del péndulo y el carro, respectivamente  $m$  y  $M$ . Se considerará que la masa del péndulo está concentrada en el extremo del mismo. La variable  $u$  es la fuerza aplicada sobre el carro, mientras que  $\theta$  y  $x$  serán el ángulo del péndulo y el desplazamiento del carro. A continuación se muestran las ecuaciones físicas del sistema:

$$M\ddot{x} = u - T\sin\theta \quad (1)$$

$$m \cdot \frac{d^2}{dt^2}(l\cos\theta) = T\cos\theta - mg \quad (2)$$

$$m \cdot \frac{d^2}{dt^2}(x + l\sin\theta) = T\sin\theta \quad (3)$$

siendo la variable  $T$  la fuerza ejercida recíprocamente entre el carro y el péndulo. Eliminando dicha variable intermedia y derivando las expresiones se obtiene:

$$ml\sin\theta \cdot (\ddot{\theta}\sin\theta + \dot{\theta}^2\cos\theta) = M\ddot{x}\cos\theta - u\cos\theta + mg\sin\theta \quad (4)$$

$$m(\ddot{x} + l\ddot{\theta}\cos\theta - l\dot{\theta}^2\sin\theta) = u - M\ddot{x} \quad (5)$$

Definiendo las siguientes variables de estado:  $x_1 = x$ ,  $x_2 = \dot{x}$ ,  $x_3 = \theta$ ,  $x_4 = \dot{\theta}$  y linealizando en torno a  $x_1 = x_2 = x_3 = x_4 = 0$  se llega al siguiente modelo en variables de estado:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(M+m)}{lM} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix} u \quad (6)$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (7)$$

Existe también la posibilidad de, siguiendo la teoría clásica de control, obtener el modelo en funciones de transferencia. Linealizando las ecuaciones (4) y (5) como en el caso anterior y aplicando la transformada de Laplace a ambos lados se obtiene:

$$s^2 x = \frac{u - mg\theta}{M} \quad (8)$$

$$(M + m)xs^2 = u - ml\theta s^2 \quad (9)$$

Eliminando  $x$  y  $\theta$  por separado se obtiene la función de transferencia de cada salida respecto a la entrada  $u$ :

$$\frac{\Theta(s)}{U(s)} = \frac{1}{-Mls^2 + g(M + m)} \quad (10)$$

$$\frac{X(s)}{U(s)} = \frac{ls^2 - g}{lMs^4 - (M + m)gs^2} \quad (11)$$

A continuación se muestran los valores numéricos de los diferentes parámetros del modelo:

<b>Parámetros del Péndulo</b>		
<i>Parámetro</i>	<i>Valor</i>	<i>Unidad</i>
M	0.1	Kg
m	0.5	Kg
g	9.81	m/s <sup>2</sup>
l	0.4	m

Tabla 3: Valores numéricos de los parámetros del modelo, correspondientes al péndulo real que se desea controlar.

Sustituyendo los valores de la tabla 3 en (6), (7), (10) y (11) se obtiene:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -49,05 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 147,1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 10 \\ 0 \\ -25 \end{bmatrix} u \quad (12)$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (13)$$

$$\frac{\Theta(s)}{U(s)} = \frac{-1}{0,04s^2 - 5,886} = \frac{-25}{(s - 12,13)(s + 12,13)} \quad (14)$$

$$\frac{X(s)}{U(s)} = \frac{0,4s^2 - 9,81}{0,04s^4 - 5,886s^2} = \frac{10(s - 4,952)(s + 4,952)}{s^2(s - 12,13)(s + 12,13)} \quad (15)$$

## 2.2. Modelado del Motor de CC y el Sistema de Tracción del Carro

En el apartado anterior se modeló el comportamiento del péndulo invertido considerando como entrada la fuerza aplicada sobre el carro. Para poder conocer el comportamiento de la planta en función de la tensión de alimentación, se procederá a modelar el motor junto con el sistema de poleas encargado de transmitir al carro el par entregado por el motor.

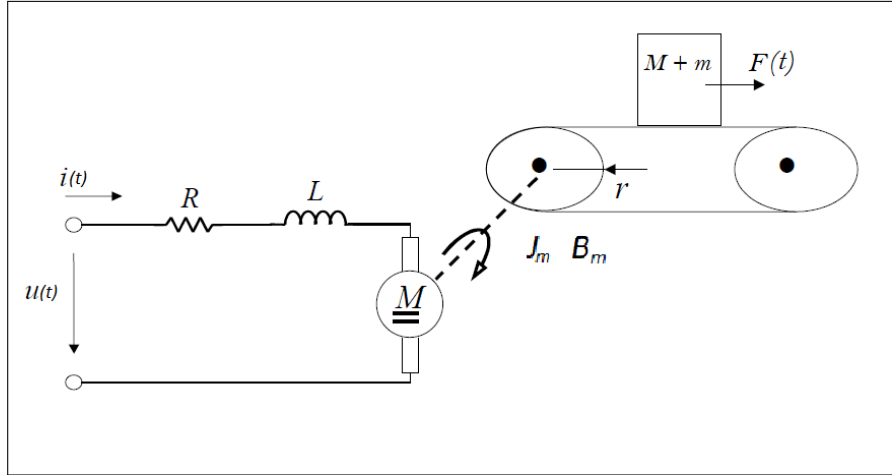


Figura 2: Diagrama del motor de corriente continua y sistema de poleas para el desplazamiento del carro.[2]

El sistema electro-mecánico representado en la Figura 2 está gobernado por las siguientes ecuaciones:

$$u(t) = R \cdot i(t) + L \frac{d}{dt} i(t) + K_b \cdot \omega(t) \quad (16)$$

$$T_m(t) = K_p \cdot i(t) = B \cdot \omega(t) + J \frac{d}{dt} \omega(t) + r F(t) \quad (17)$$

$$F(t) = (M + m) \cdot r \cdot \frac{d}{dt} \omega(t) \quad (18)$$

Siendo  $J$  el momento de inercia total del rotor y de la carga con respecto al eje del rotor;  $K_p$  y  $K_b$  las constantes de par y fuerza electro-motriz del motor, respectivamente;  $T_m$  el par entregado por el motor y  $B$  la constante de rozamiento viscoso del rotor, el sistema de poleas y el carro. Aplicando Laplace en las tres expresiones y eliminando las variables  $\omega$  e  $i$  se obtiene, tras despejar y simplificar, la siguiente función de transferencia, que relaciona la fuerza aplicada sobre el carro y la tensión de entrada del motor:

$$\frac{F(s)}{U(s)} = \frac{s \cdot K_p \cdot r \cdot (M + m)}{s^2 \cdot (LJ + r^2 L(M + m)) + s \cdot (RJ + LB + r^2 L(M + m)) + K_p K_b + RB} \quad (19)$$

A continuación se muestran los valores numéricos de los diferentes parámetros del modelo:

<b>Parámetros del Motor DC</b>		
<i>Parámetro</i>	<i>Valor</i>	<i>Unidad</i>
$K_p$	0.0494	$N \cdot m/A$
$K_b$	0.0308	$V/rad/s$
$R$	6.84	$\Omega$
$L$	0.0105	$H$
$r$	0.005	$m$
$M+m$	0.6	$Kg$
$J$	$1,5 \cdot 10^{-5}$	$Kg \cdot m^2$
$B$	$3 \cdot 10^{-4}$	$N \cdot m \cdot s/rad$

Tabla 4: Valores numéricos de los parámetros del modelo, correspondientes al motor y la carga del mismo.

Para calcular el valor del momento de inercia total  $J$  se ha empleado la siguiente expresión:

$$J = (M + m) \cdot r^2 \quad (20)$$

Donde  $M$ ,  $m$  y  $r$  son la masa del carro, la masa del péndulo y el radio de las poleas que transportan el carro, respectivamente.

Para estimar el valor del coeficiente de rozamiento viscoso  $B$ , se le ha asignado inicialmente un valor arbitrario y se ha implementado el modelo en *Simulink*. Comparando las respuestas en cadena abierta del sistema real y del modelo simulado, se ha podido ajustado el valor de  $B$ .



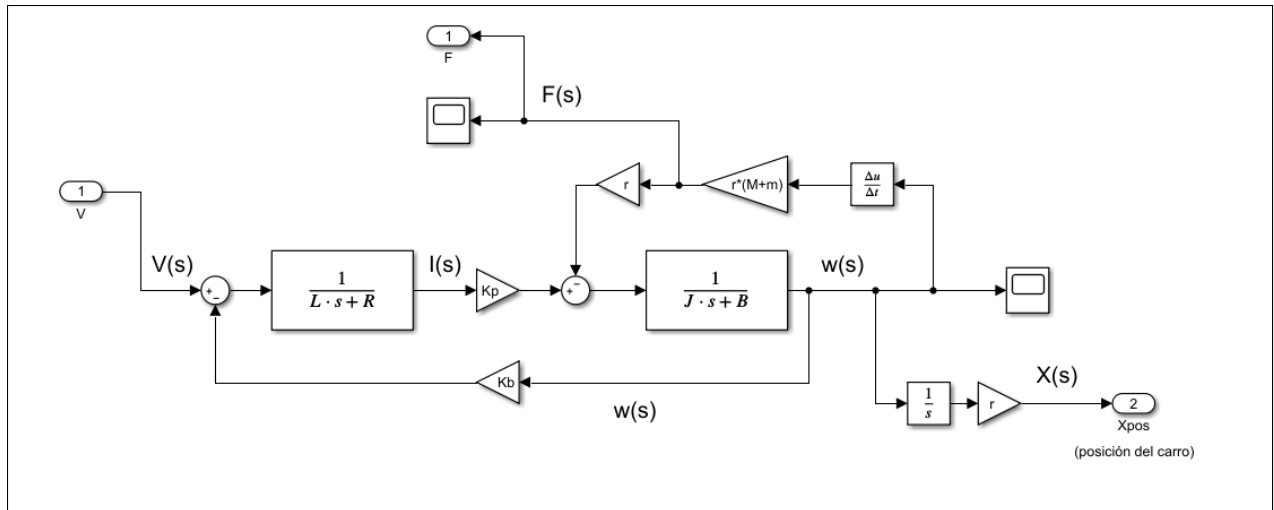


Figura 3: Modelo realizado en Simulink del conjunto motor-carro.

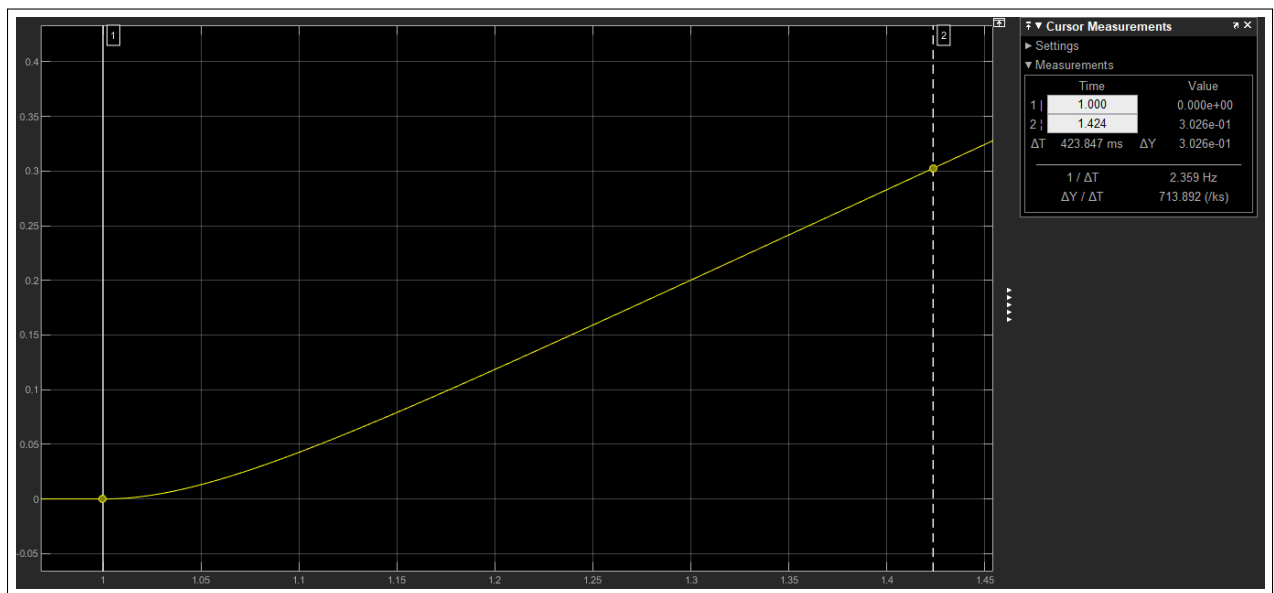


Figura 4: Respuesta del modelo en cadena abierta ante un escalón de amplitud 12 V. La respuesta mostrada se corresponde con el desplazamiento del carro en metros (tras ajustar el valor del coeficiente  $B$ ).

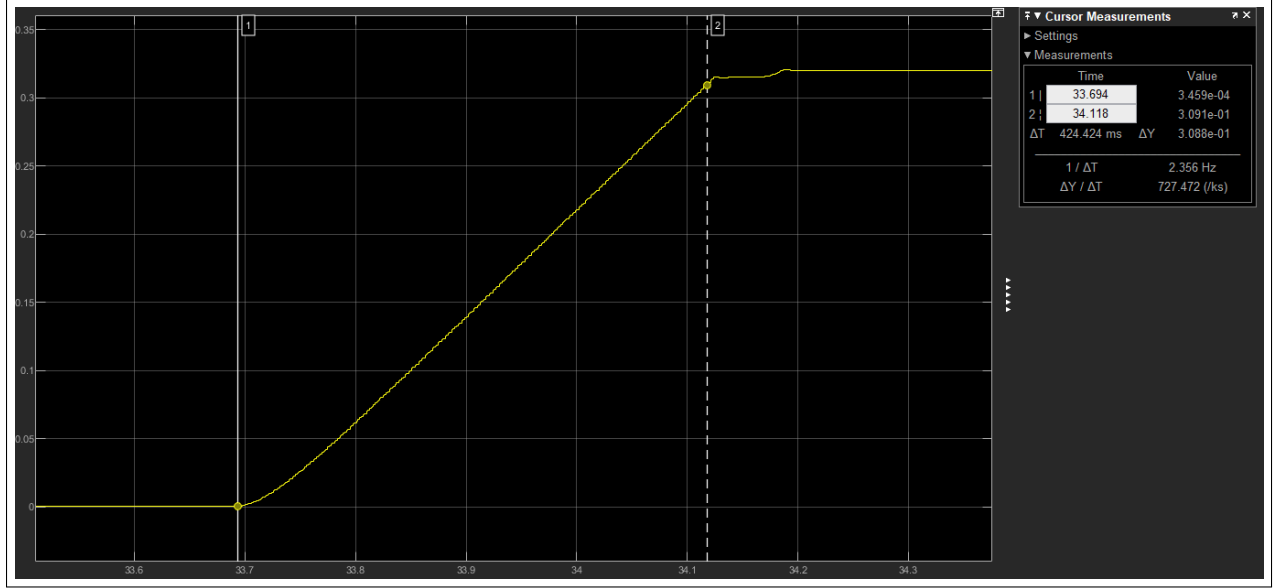


Figura 5: Respuesta del sistema real en cadena abierta ante un escalón de amplitud 12 V.

Sustituyendo los valores de la Tabla 4 en (19) se obtiene:

$$\begin{aligned} \frac{F(s)}{U(s)} &= \frac{1,482 \cdot 10^{-4} \cdot s}{3,15 \cdot 10^{-7} \cdot s^2 + 2,083 \cdot 10^{-4} \cdot s + 0,003574} = \\ &= \frac{470,48 \cdot s}{(s + 643,8)(s + 17,62)} \approx \frac{0,7713 \cdot s}{(s + 18,13)} \end{aligned} \quad (21)$$

Siendo la entrada  $U(s)$  la tensión de alimentación del motor. Se ha realizado una aproximación para reducir en un grado el orden del sistema.

Para el modelo en variables de estado será de utilidad emplear una relación simplificada entre la fuerza entregada por el motor y la tensión de alimentación del mismo. Despreciando la influencia de la inductancia  $L$  y eliminando la variable  $i$  de las expresiones (16), (17) y (18) se obtiene la siguiente relación:

$$\begin{aligned} F &= \frac{K_p(M + m)r}{(J + r^2)} \cdot V_m - \frac{(RB + K_p K_b)(M + m)}{(J + r^2)} \cdot \dot{x} = \\ &= -\frac{(RB + K_p K_b)(M + m)}{(J + r^2)} \cdot x_2 + \frac{K_p(M + m)r}{(J + r^2)} \cdot V_m \end{aligned} \quad (22)$$

Siendo  $V_m$  la tensión aplicada al motor y  $r$  el radio de la polea encargada de transmitir el movimiento al carro.

## 2.3. Modelo del Conjunto

Multiplicando las funciones de transferencia (21) y (14) se obtiene la función de transferencia global (ángulo en función de voltaje de entrada):

$$\frac{\Theta(s)}{V_m(s)} = \frac{-17,783s}{(s - 12,13)(s + 12,13)(s + 18,13)} \quad (23)$$

Sustituyendo la expresión (22) en (6) se obtiene el modelo en variables de estado del conjunto:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(RB+K_pK_b)(M+m)}{M(J+r^2)} & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{(RB+K_pK_b)(M+m)}{lM(J+r^2)} & \frac{g(M+m)}{lM} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{(M+m)rK_p}{M(J+r^2)} \\ 0 \\ -\frac{(M+m)rK_p}{lM(J+r^2)} \end{bmatrix} V_m \quad (24)$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (25)$$

Sustituyendo los valores numéricos se llega a:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -536 & -49,05 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1340 & 147,1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 37,05 \\ 0 \\ -92,62 \end{bmatrix} V_m \quad (26)$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (27)$$

## 3. Diseño del Regulador

### 3.1. PID Clásico

Para ajustar los parámetros  $K_p$ ,  $K_d$  y  $K_i$  del regulador, una posibilidad es emplear el método de Ziegler-Nichols para respuesta en frecuencia. Éste consiste en aumentar la ganancia en cadena cerrada hasta que el sistema oscile y calcular los parámetros del regulador a partir de la ganancia alcanzada y el periodo de oscilación. A través de *MATLAB* podemos obtener los parámetros del PID directamente empleando la herramienta *pidTuner*. Aplicando la aproximación de *Tustin* en (23) para un tiempo de muestreo de 1 ms se obtiene:

$$G_{total}(z) = \frac{-4,4059 \cdot 10^{-6} \cdot (z - 1)(z + 1)^2}{(z - 1,012)(z - 0,9879)(z - 0,982)} \quad (1)$$

A continuación se muestra la expresión del PID obtenido:

$$G_{PID}(z) = K_p + K_i \cdot \frac{T_s}{(z-1)} + K_d \cdot \frac{1}{T_f + \frac{T_s}{(z-1)}} \quad (2)$$

Donde  $T_s = 0,001$  s;  $K_p = -571,1$ ;  $K_i = -9031$ ;  $K_d = -8,698$  y para el filtro de la acción derivativa:  $T_f = 0,001082$  s.

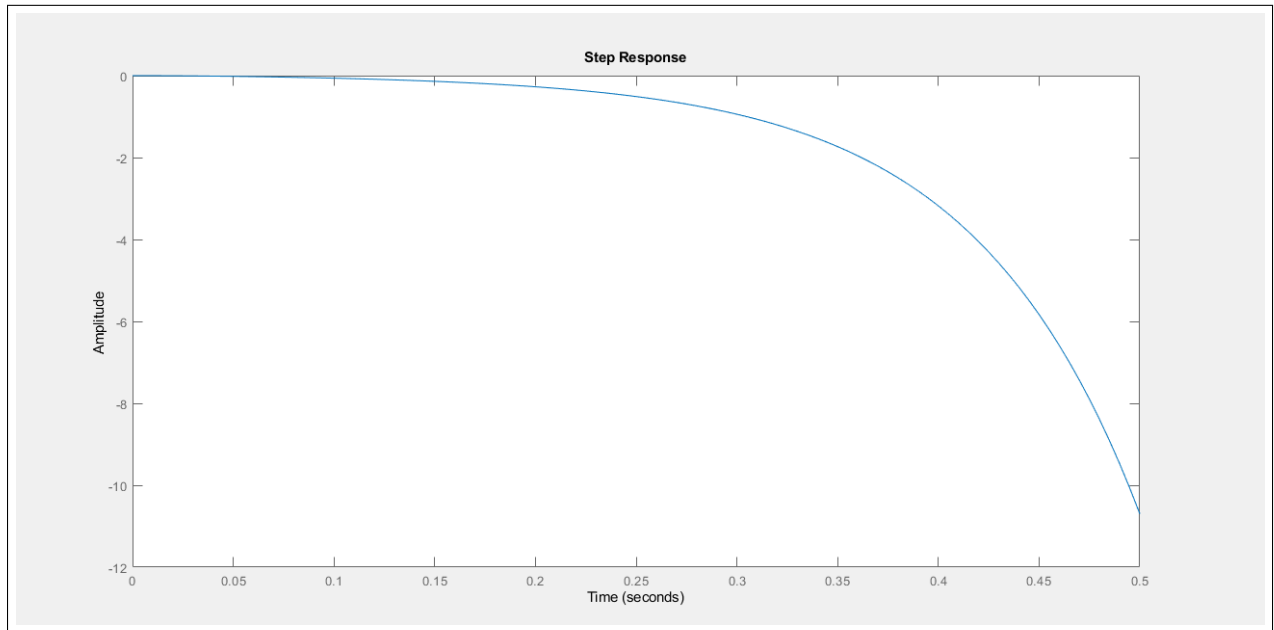


Figura 6: Respuesta al escalón del modelo completo en cadena abierta.

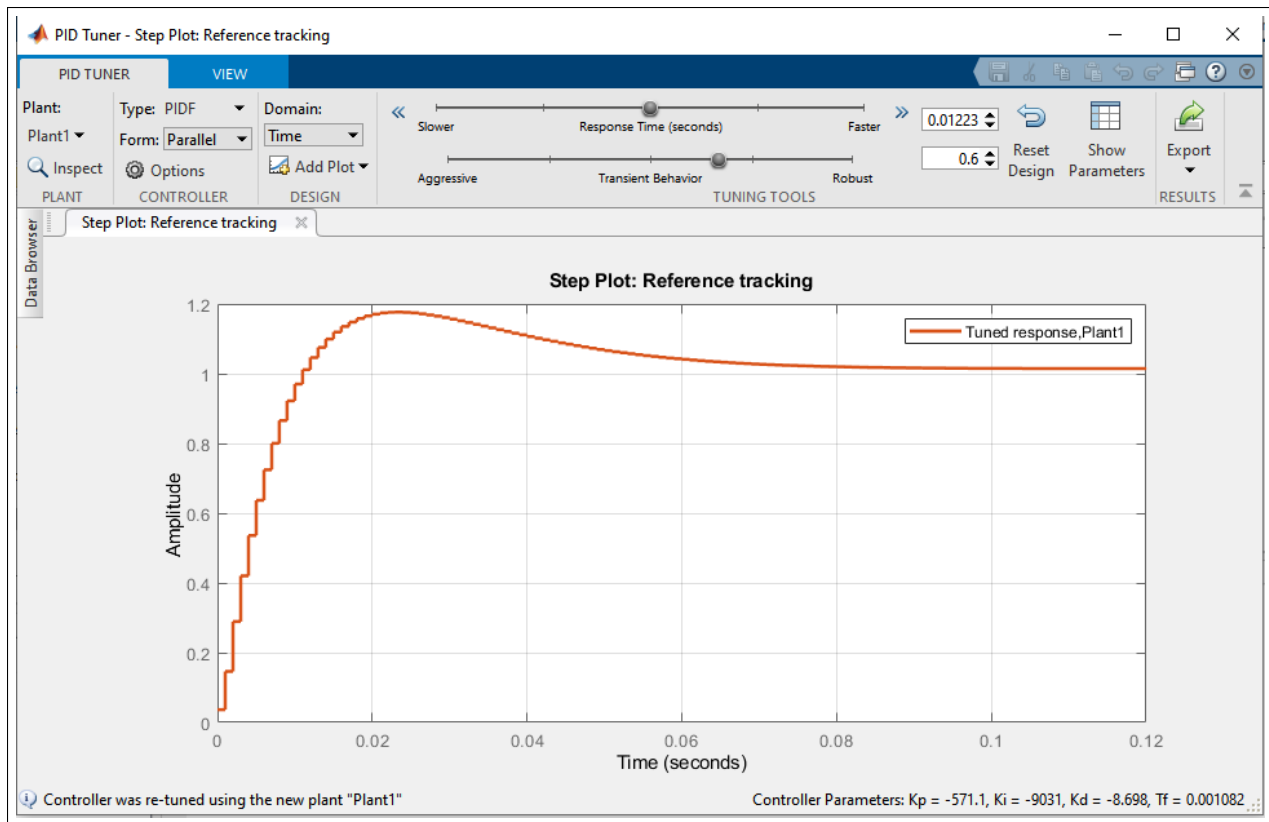


Figura 7: Respuesta al escalón del modelo completo en cadena cerrada con el regulador PID obtenido mediante la herramienta *pidTuner*.

Para comprobar el comportamiento ante perturbaciones del regulador y el orden de magnitud de la acción de control, se ha simulado el conjunto modelo-regulador.

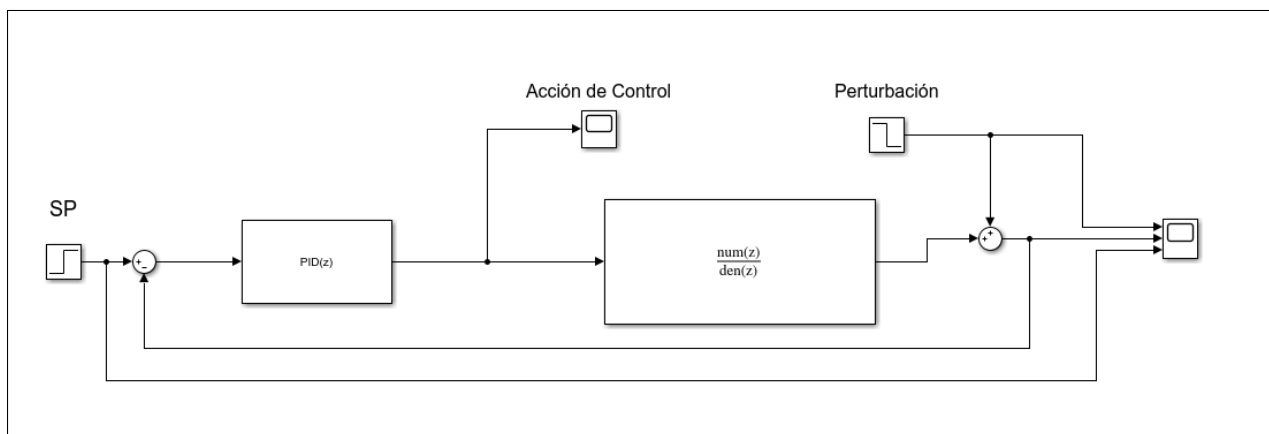


Figura 8: Implementación del PID en *Simulink* para simulación.

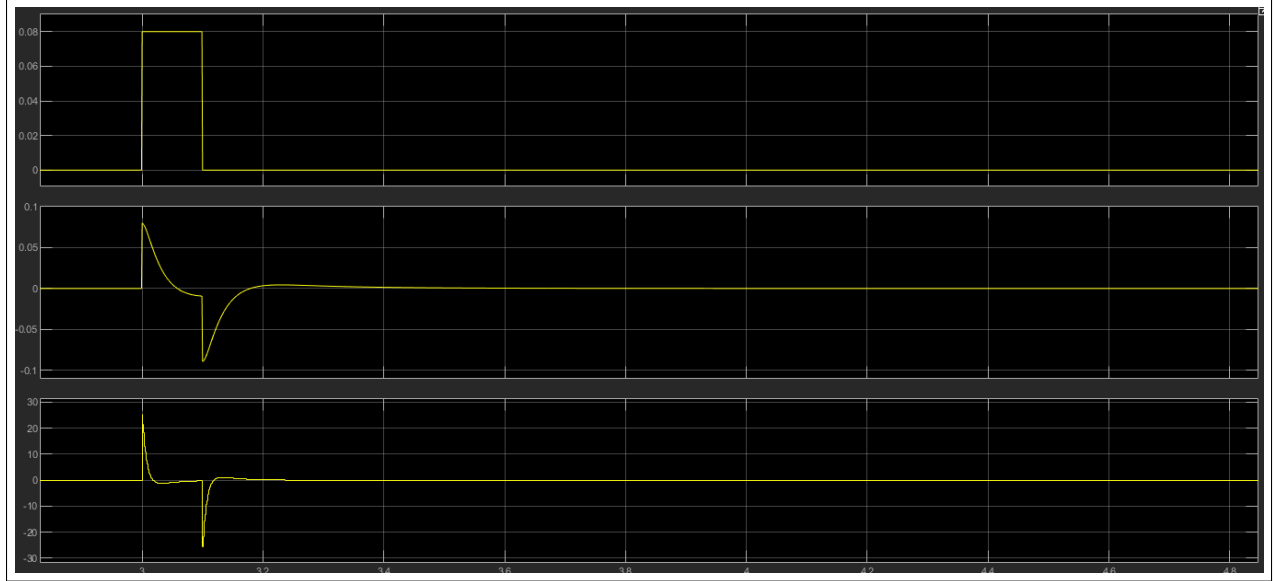


Figura 9: Respuesta del sistema con regulador PID ante una perturbación. De arriba a abajo se muestra: la señal de perturbación, la respuesta del sistema, y la acción de control.

### 3.2. Asignación de Polos por Realimentación del Estado

Para sistemas MIMO con dinámicas más complejas, la teoría moderna de control puede resultar más potente que la clásica. El modelado en el espacio de estados nos permite tener en cuenta la evolución de las variables internas que definen la dinámica del sistema, haciendo posible una regulación más robusta. Además, en el caso del péndulo invertido, será posible estabilizar no sólo el ángulo del péndulo, sino también la posición del carro.

Antes de proceder con el diseño del regulador, se ha comenzado estudiando la controlabilidad y observabilidad del sistema. Las matrices  $P$  y  $Q$  de observabilidad y controlabilidad del sistema completo son:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -536 & -49,05 & 0 \\ 0 & 1340 & 147,15 & 0 \\ 0 & 287330 & 26292 & -49,05 \\ 0 & -718320 & -65730 & 147,15 \end{bmatrix} \quad (3)$$

$$Q = \begin{bmatrix} 0 & 37,05 & -19860 & 10650000 \\ 37,05 & -19860 & 10650000 & -5711100000 \\ 0 & -92,625 & 49650 & -26627000 \\ -92,625 & 49650 & -26627000 & 14279000000 \end{bmatrix}$$

Dado que el rango de ambas matrices es máximo, se puede concluir que el sistema es tanto observable como controlable.

A continuación se han calculado los autovalores de la matriz A para obtener los polos de la cadena abierta. Se ha obtenido:  $p_1 = -536,26$ ;  $p_2 = 5,0669$ ;  $p_3 = -4,8382$  y  $p_4 = 0$ . Dado que uno de los polos se encuentra en el semiplano  $s$  positivo, podemos confirmar que el sistema es inestable en cadena abierta.

### 3.2.1. Asignación Directa

En un primer momento se optó por tratar de estabilizar el sistema mediante asignación directa de polos. Para sistemas de orden superior puede resultar complicado correlacionar las características dinámicas de la respuesta deseada con la posición de los polos de la cadena cerrada. Probando con distintos polos y valores de K se alcanzó una respuesta satisfactoria al posicionar los polos en  $p_1 = -2$ ;  $p_2 = -3$ ;  $p_3 = -4$  y  $p_4 = -1000$ . La matriz K obtenida fue:

$$K = [-21,08 \quad -37,33 \quad -87,97 \quad -17,88]$$

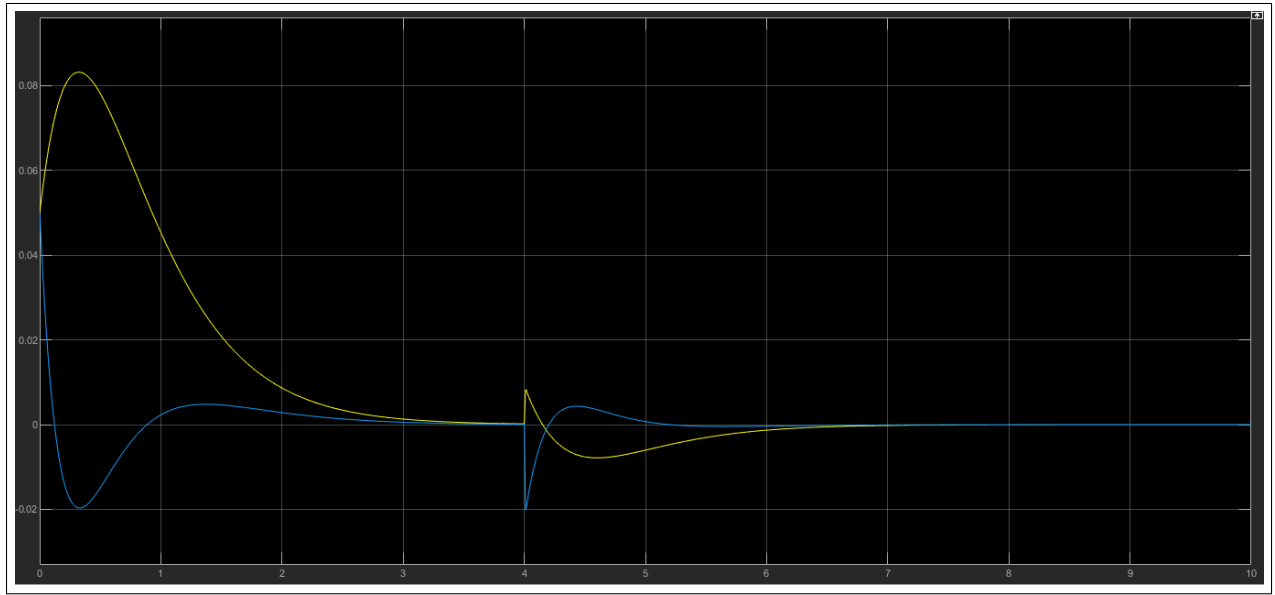


Figura 10: Respuesta del modelo con regulador por asignación directa depolos, ante condiciones iniciales  $[0,05 \ 0 \ 0,05 \ 0]$  y una perturbación en el segundo 4. Se muestran la posición del carro (amarillo) y el ángulo del péndulo (azul).

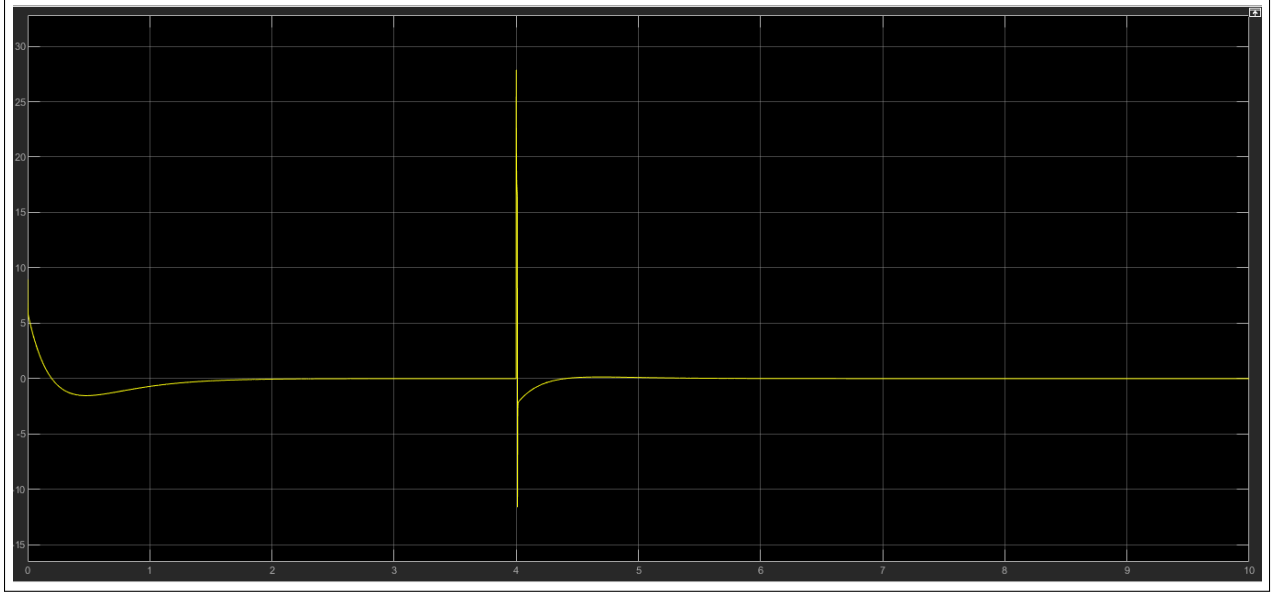


Figura 11: Señal de control para el caso mostrado en la figura anterior.

### 3.2.2. Regulador Óptimo Cuadrático (LQR)

Frente al método de asignación directa, el control LQR ofrece un procedimiento sistemático y eficiente para la obtención de la matriz de ganancia de realimentación del estado. *MATLAB* ofrece numerosas herramientas para implementar estos reguladores de manera cómoda y rápida. En aspectos generales, el problema de control óptimo consiste en determinar la matriz de ganancia  $K$  que permita minimizar la función de coste  $\int_0^\infty (x'Qx + u'Ru) dt$ . Esto se logra obteniendo la solución  $P$  de la ecuación reducida de *Riccati*:  $A'P + PA - PBR^{-1}B'P + Q = 0$ . Finalmente, la matriz  $K$  se obtiene como  $R^{-1}B'P$ .

En *MATLAB* el cálculo de la matriz  $K$  puede realizarse a través de las funciones "*lqr*" y "*dlqr*" (para sistemas discretos) que recibirán las matrices  $A$  y  $B$  del sistema a controlar, así como las matrices de pesos  $Q$  y  $R$  de la función de coste, que definirán el rendimiento del regulador y el esfuerzo energético del control, respectivamente.

Antes de proceder con la obtención de  $K$  se ha obtenido el equivalente discreto del sistema. Mediante la función "*c2d*" se ha realizado una aproximación por *Tustin* para  $T_s = 0,001s$ . No se ha considerado el efecto de la matriz  $D$ .

$$\begin{aligned}
 x(k+1) &= \begin{bmatrix} 1 & 7,886 \cdot 10^{-4} & -1,934 \cdot 10^{-5} & -9,671 \cdot 10^{-9} \\ 0 & 0,5773 & -0,03868 & -1,934 \cdot 10^{-5} \\ 0 & 5,284 \cdot 10^{-4} & 1 & 0,001 \\ 0 & 1,057 & 0,121 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1,416 \cdot 10^{-5} \\ 0,02922 \\ -3,652 \cdot 10^{-5} \\ -0,07305 \end{bmatrix} V_m(k) \\
 y(k) &= \begin{bmatrix} 1 & 3,943 \cdot 10^{-4} & -9,671 \cdot 10^{-6} & -4,835 \cdot 10^{-9} \\ 0 & 2,642 \cdot 10^{-4} & 1 & 0,0005 \end{bmatrix} x(k)
 \end{aligned} \tag{4}$$



A continuación se ha obtenido, mediante la función "dlqr", la matriz de ganancias que minimiza la función de coste  $J = \sum_{n=0}^{\infty} x'Qx + u'Ru$ . Tras varias iteraciones se han logrado resultados satisfactorios con los siguientes valores de  $Q$  y  $R$ :

$$Q = \begin{bmatrix} 500 & 0 & 0 & 0 \\ 0 & 0 & & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad R = 0,005;$$

El vector de ganancias  $K_{LQR}$  obtenido es el siguiente:

$$K_{LQR} = [-303,88 \quad -207,28 \quad -598,58 \quad -83,753]$$

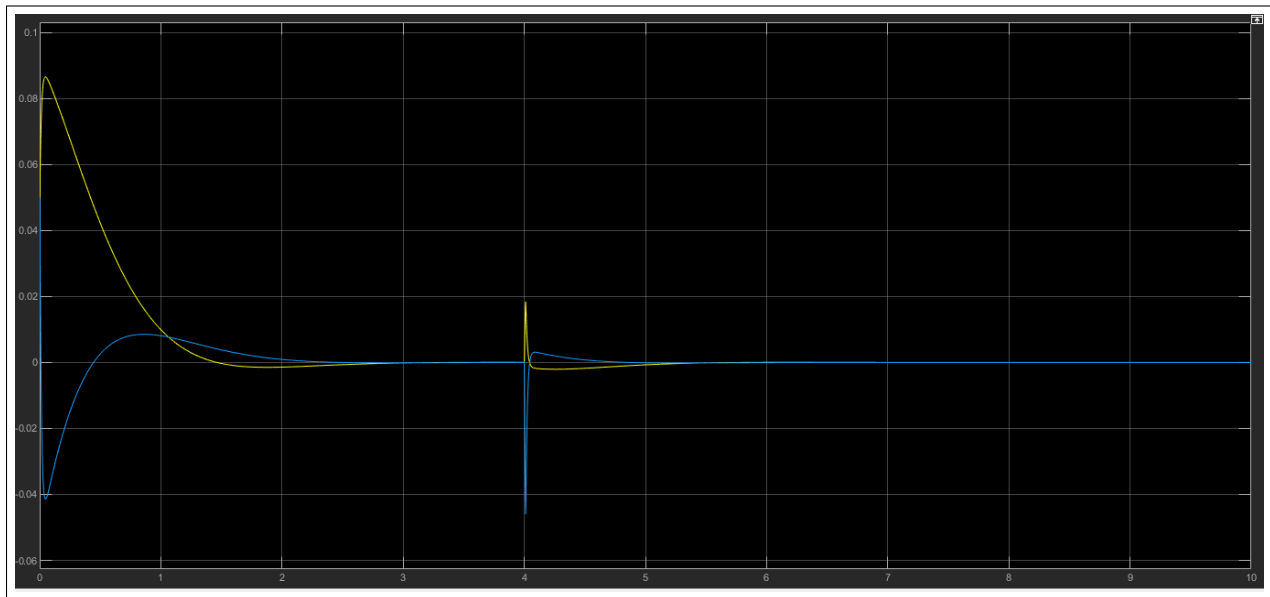


Figura 12: Respuesta del modelo con regulador LQR ante condiciones iniciales  $[0,05 \ 0 \ 0,05 \ 0]$  y una perturbación en el segundo 4. Se muestran la posición del carro (amarillo) y el ángulo del péndulo (azul).

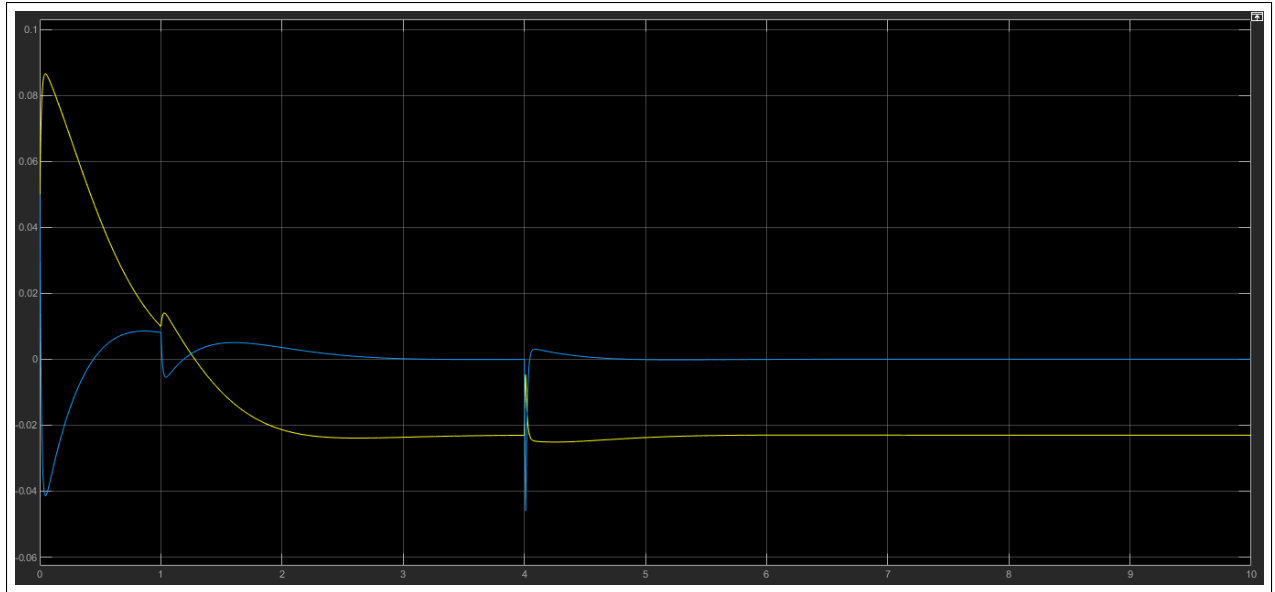


Figura 13: Respuesta del modelo con regulador LQR ante condiciones iniciales  $[0,05 \ 0 \ 0,05 \ 0]$ , entrada 7 V en el segundo 1 y una perturbación en el segundo 4. Se muestran la posición del carro (amarillo) y el ángulo del péndulo (azul).

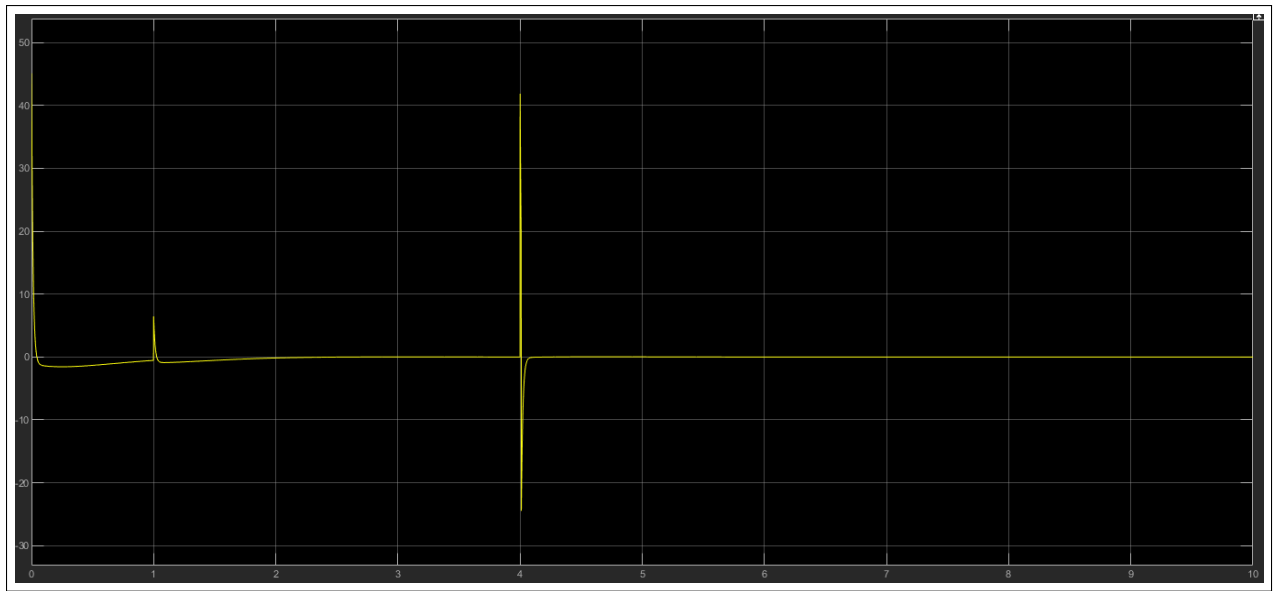


Figura 14: Señal de control para el caso mostrado en la figura anterior.

## Observaciones

En la figura 13 se puede observar que para señales de referencia no nulas, la variable  $x$  (posición del carro) presenta error en régimen permanente. Sin embargo, ante entrada nula el sistema se mantiene centrado, con error en régimen permanente nulo para ambas salidas. En la figura 14 se observa que ante las condiciones iniciales y la perturbación, la señal de control presenta picos considerables. Ajustando el valor de la matriz  $R$  podrían reducirse estos valores; sin embargo, más tarde se comprobó que al implementar el regulador en el sistema físico, se obtenían mejores resultados con ganancias mayores. En el sistema real, dichos picos saturarían a un valor de 12V, que es la alimentación máxima del motor.

En comparación con el regulador obtenido por asignación directa, el LQR presenta una respuesta más rápida pero unos picos mayores en la señal de control. Esto puede revertirse reajustando los valores de  $Q$  y  $R$  si así se deseara. La obtención del regulador LQR ha sido más intuitiva y eficiente, mientras que la obtención por asignación directa se ha basado más en el ensayo y error, y en la elección algo arbitraria del posicionamiento deseado de los polos.

### 3.2.3. Servocontrolador Óptimo

Si además de mantener el péndulo en vertical queremos controlar la posición del carro, no nos bastará con una simple realimentación del estado, también habrá que realimentar la salida. Para ello se implementará un servocontrolador.

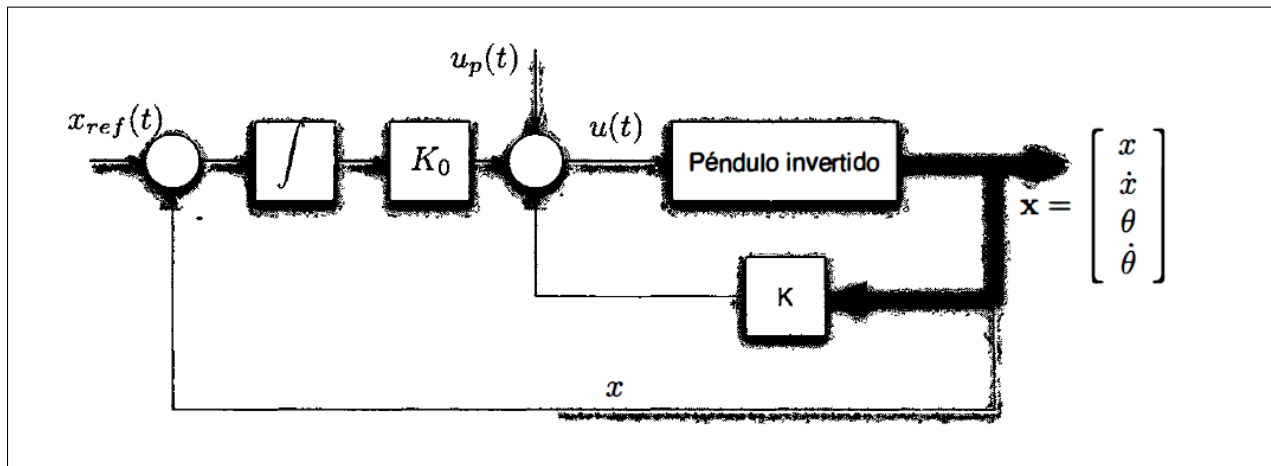


Figura 15: Diagrama del Servocontrolador a implementar. Incluye realimentación del estado e integración del error de la variable  $x$  (posición del carro).[1]

Análogamente al caso del LQR, buscaremos el vector de ganancia  $K_{servo}$  que minimice la

función de coste. En este caso el vector de ganancia tendrá dimensión cinco y será tal que:

$$u = -K_{servo} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ x_i \end{bmatrix}$$

En *MATLAB* puede resolverse el problema de optimización con la función *"lqi"*. Los pesos asignados a las matrices  $P$  y  $Q$  tras varias iteraciones han sido:

$$Q = \begin{bmatrix} 2000 & 0 & 0 & 0 & 0 \\ 0 & 0 & & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5000 \end{bmatrix} \quad R = 0,015;$$

El vector de ganancia obtenido ha sido:

$$K_{servo} = [-643,1 \quad -270,77 \quad -596,22 \quad -108,92 \quad 561,13]$$

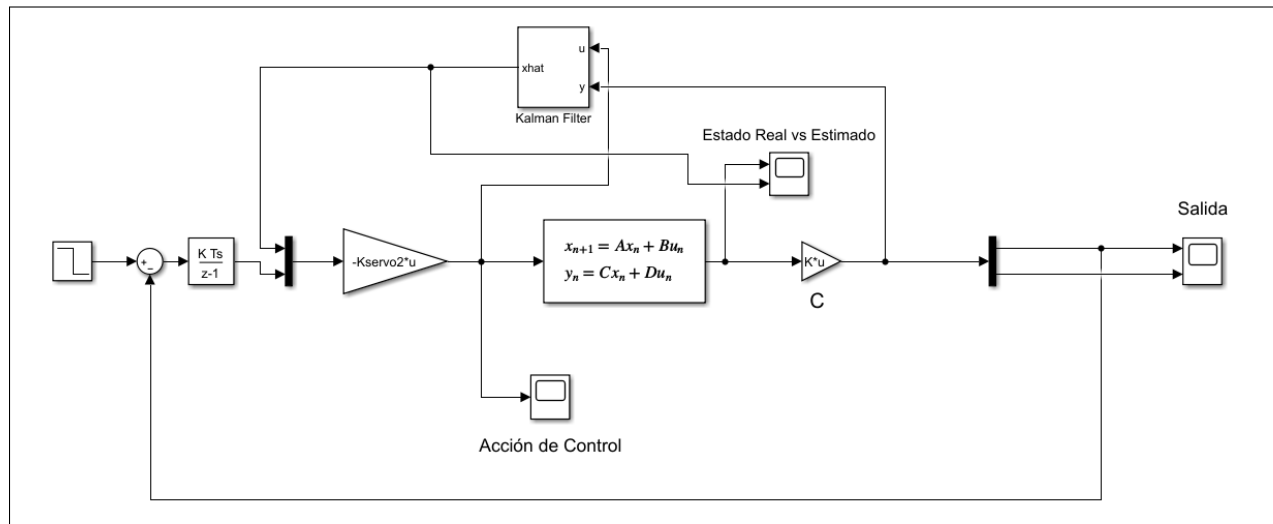


Figura 16: Implementación en *Simulink* del servocontrolador para regular la posición del carro.

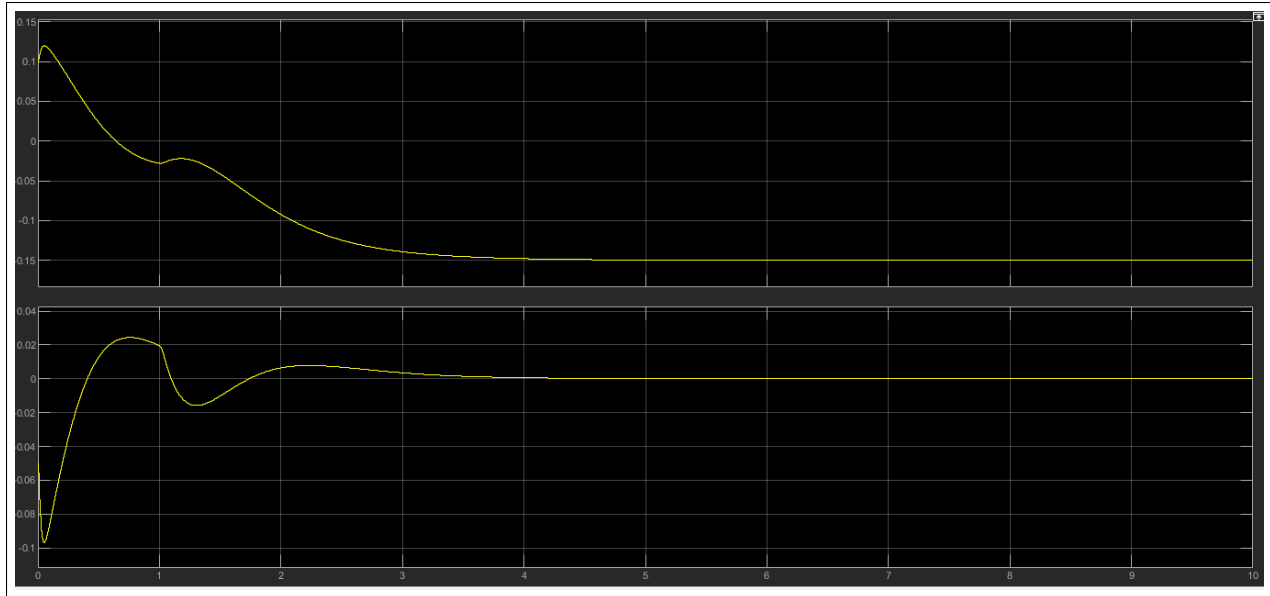


Figura 17: Respuesta del servosistema ante entrada escalón de valor -0.15 y condiciones iniciales  $[0.1 \ 0 \ -0.05 \ 0.1]$ . Posición del carro (superior) frente al ángulo del péndulo (inferior).

## Observaciones

Al añadir la realimentación de la medida de posición del carro, se ha conseguido eliminar el error en régimen permanente de dicha variable. Además, se ha hecho posible el control de la posición del carro con la señal de referencia.

## 4. Diseño del Observador

Al diseñar el regulador por asignación de polos, se ha supuesto que todas las variables de estado están accesibles para su realimentación. En el sistema físico a controlar, sin embargo, solo tenemos acceso a dos de las cuatro variables de estado (ángulo del péndulo y posición del carro). Una posible solución sería medir también las velocidades del carro y el péndulo; mientras que otra posibilidad sería derivar las dos variables de estado disponibles para obtener las restantes. Para la primera solución no disponemos de la sensórica necesaria, y la segunda opción tendría como resultado medidas considerablemente ruidosas. Otra posibilidad, que se abordará a continuación, es el diseño de un observador del estado.

### 4.1. Diseño de Observador por Asignación Directa de Polos

El diseño de un observador por asignación directa de polos es análogo al diseño de un regulador por el mismo método. Deberá calcularse la matriz de ganancia  $K_e$  tal que la

matriz  $A - K_e C$  sea asintóticamente estable y el error entre el estado de la planta y el estado estimado converja a cero lo suficientemente rápido.

En el apartado anterior se obtuvo un vector de ganancia  $K_{LQR}$  que posicionó los polos de la cadena cerrada en:  $p_{1,2} = -2,613 \pm 1,5499i$ ;  $p_3 = -81,474$  y  $p_4 = -543,51$ . Es recomendable que los polos dominantes del observador sean de 2 a 5 veces más rápidos que los del sistema observado para que el error de estimación converja a cero rápidamente. Tras varias iteraciones, finalmente se ha optado por la siguiente asignación:  $p_1 = p_2 = -6,6$ ;  $p_3 = -275$  y  $p_4 = -500$ . El vector  $K_e$  obtenido es:

$$K_e = [-1846,9 \quad -418,72 \quad -1020,9 \quad -168,35]$$

Es importante señalar que la asignación de polos se ha realizado en el plano  $z$ , ya que se ha trabajado con el modelo discreto del sistema.

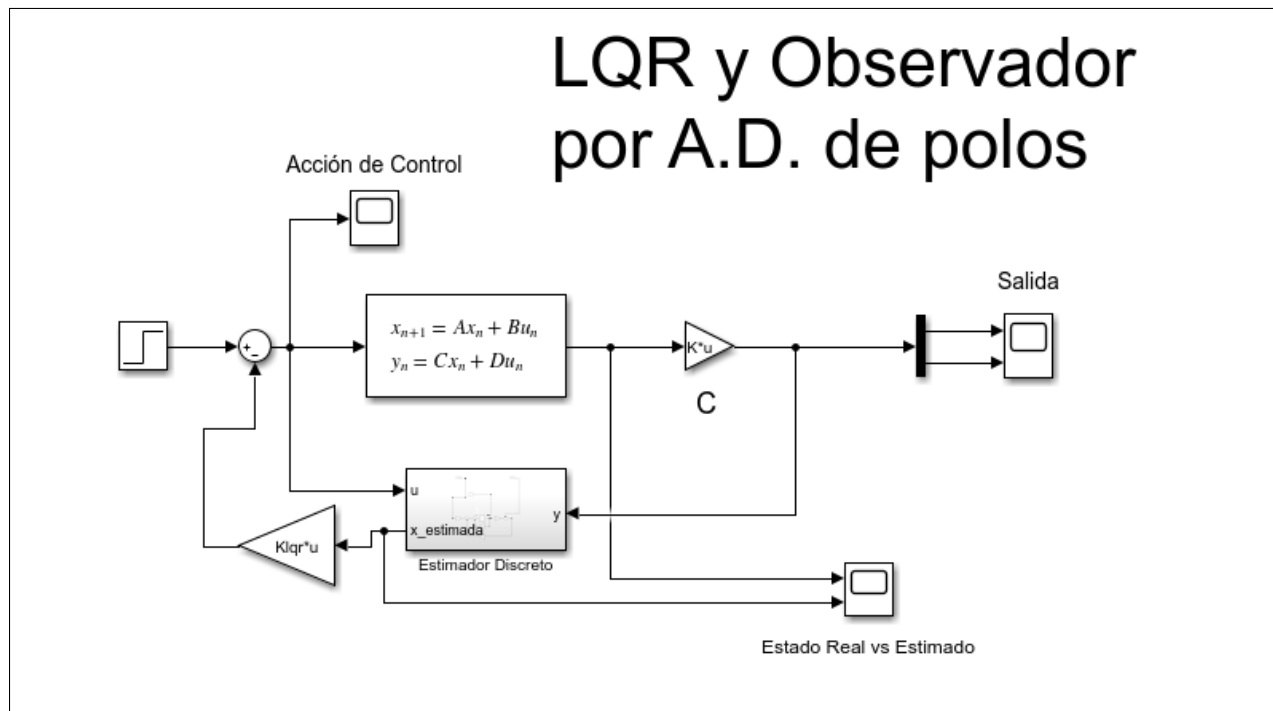


Figura 18: Implementación en *Simulink* del modelo con Regulador Óptimo Cuadrático y Observador obtenido por asignación directa de polos.

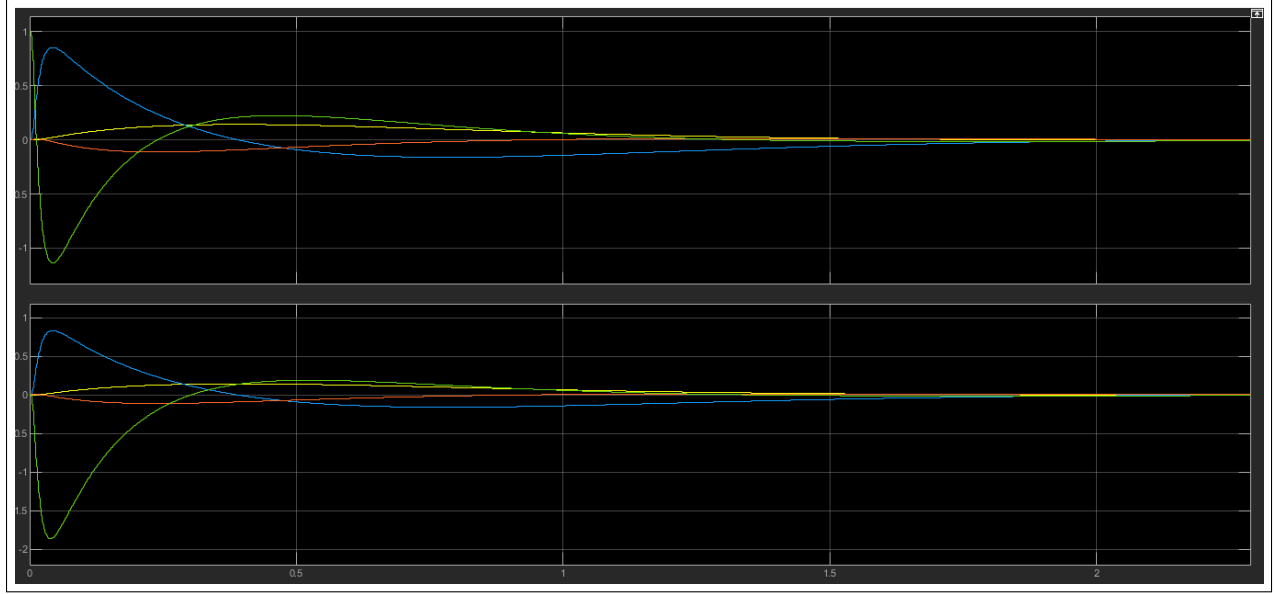


Figura 19: Evolución del estado (superior) frente a la evolución del estado estimado por el observador (inferior) ante entrada nula y condiciones iniciales  $[0 \ 0 \ 0 \ 1]$ .

En la figura 19 se observa que la estimación es bastante buena, aunque existen discrepancias; especialmente en los primeros instantes.

## 4.2. Diseño de Observador Óptimo - Filtro de Kalman

El diseño del estimador óptimo es similar al de un regulador óptimo. El objetivo será obtener la matriz de ganancia  $H_{Kalman}$  que minimice el error de la estimación, así como la influencia del ruido en las medidas y la incertidumbre del modelo. Para resolver el problema de estimación óptima deberá hallarse la solución de la ecuación de *Riccati* correspondiente. En *MATLAB* se ha obtenido dicha solución empleando la función "*kalman*", que recibe las matrices de covarianza  $Q$  y  $R$ . En este caso, la matriz  $Q$  está asociada al ruido en los estados, o incertidumbre del modelado; mientras que  $R$  se asocia al ruido medido en la salida. Se ha optado por los siguientes valores de  $Q$  y  $R$ :

$$Q = 10 \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

La matriz de ganancia obtenida es:

$$H_{Kalman} = \begin{bmatrix} 2,2112 \cdot 10^{-4} & -1,742 \cdot 10^{-4} \\ 3,4603 \cdot 10^{-5} & -9,6336 \cdot 10^{-4} \\ -1,7412 \cdot 10^{-4} & 0,01012 \\ -8,17 \cdot 10^{-4} & 0,051323 \end{bmatrix}$$

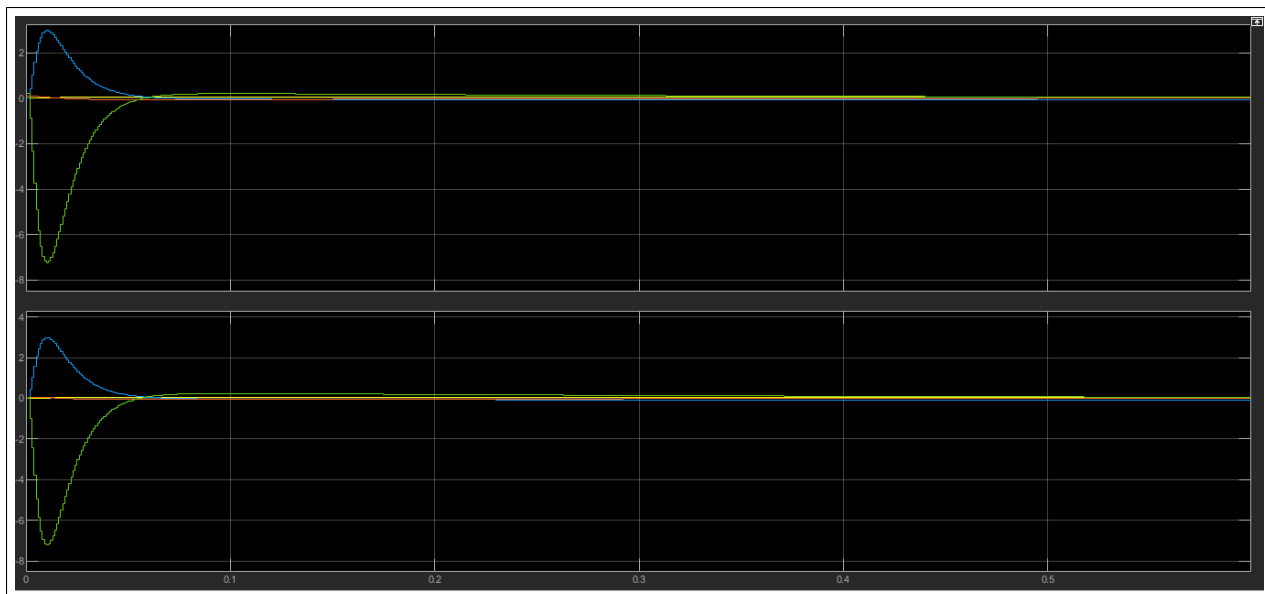


Figura 20: Evolución del estado del modelo (superior) frente a la evolución del estado estimado por el filtro de Kalman (inferior) ante entrada nula y condiciones iniciales  $[0 \ 0 \ 0.09 \ 0.2]$ .

### Observaciones

En el sistema a controlar, el ruido en las medidas no supone una gran preocupación, ya que la lectura de los encoders es bastante precisa y no presenta un ruido apreciable. Para el diseño del filtro de Kalman podremos, por lo tanto, priorizar la minimización de la incertidumbre del modelado. La estimación del estado del filtro de Kalman es apreciablemente mejor que la del observador obtenido por asignación directa de polos.

## 5. Controlador para el Balanceo del Péndulo (Swing-Up)

El objetivo es implementar un controlador que lleve al péndulo desde el reposo hasta la posición vertical de inestabilidad, donde debe quedar estabilizado. En este apartado sólo se abordará el balanceo, ya que la estabilización se ha tratado en apartados anteriores.



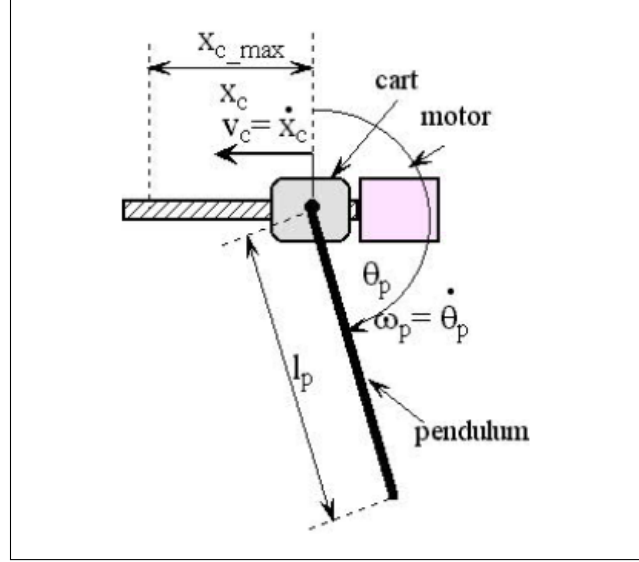


Figura 21: Diagrama del Péndulo Invertido.[4]

El enfoque clásico al problema del balanceo del péndulo invertido se basa en la acumulación de energía. Las ecuaciones de movimiento y equilibrio de energía del péndulo son:

$$2 \cdot J \cdot \frac{d\omega_p}{dt} - mgl_p \cdot \text{sen}(\theta_p) + l_p m a_c \cdot \text{cos}(\theta_p) = 0 \quad (1)$$

$$E = \frac{1}{2} J \omega_p^2 + \frac{1}{2} mgl_p (\text{cos}(\theta_p) - 1) \quad (2)$$

Siendo  $J$  el momento de inercia del péndulo. Derivando (2) y sustituyendo en (1) se obtiene:

$$\frac{dE}{dt} = -\frac{1}{2} m a_c l_p \omega_p \cdot \text{cos}(\theta_p) \quad (3)$$

De donde se puede deducir que la energía almacenada en el péndulo incrementará siempre que se cumpla:

$$a_c = -a_{cmax} \cdot \text{signo}(\omega_p) \cdot \text{signo}(\text{cos}(\theta_p)) \quad (4)$$

El péndulo será menos controlable cuando la velocidad del mismo sea pequeña o cuando se encuentre en horizontal. Por el contrario, la acción de control será más significativa cuando se encuentre en vertical. Considerando las particularidades físicas del péndulo y el carro, se considerará que el mejor ángulo  $\alpha$  desde el que empezar aplicar la acción de control antes de llegar a la posición vertical será:

$$\alpha = \theta_{max} \cdot \text{sen}\left(\frac{2\pi}{T} t_{ac}\right) \quad (5)$$

Siendo  $\theta_{max}$  la máxima amplitud de oscilación actual,  $T$  el periodo de oscilación del péndulo y  $t_{ac}$  el tiempo de aceleración del carro. El periodo  $T$  del péndulo se puede calcular fácilmente

como  $T = 2\pi\sqrt{g/l_p}$ , mientras que el tiempo  $t_{ac}$  se puede obtener a partir del ensayo mostrado en la Figura (5). Los valores obtenidos son:

Cálculo de alfa		
Parámetro	Valor	Unidad
$l_p$	0.4	m
T	0.41	s
$t_{ac}$	0.02	s
$\alpha$	0.30172	rad

Finalmente se debe tener en cuenta la limitación en la longitud de la carrera del carro. Para el balanceo se ha decidido limitar el desplazamiento del carro en  $\pm 0,045m$ : Teniendo en cuenta todo lo dicho, se ha implementado en *Simulink* el controlador.

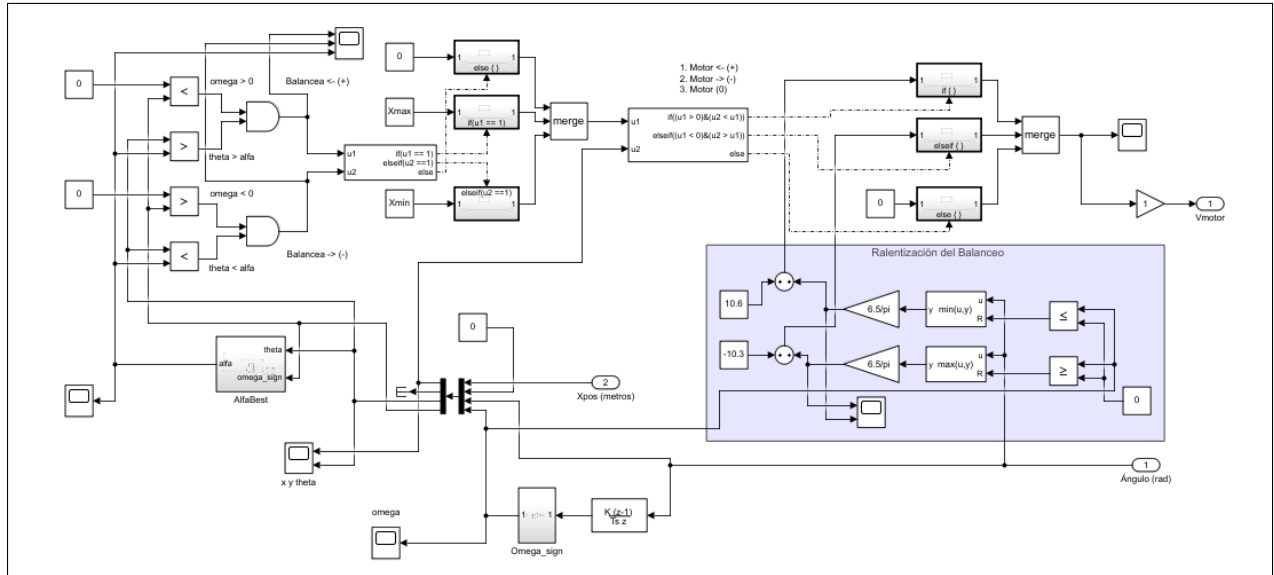


Figura 22: Controlador implementado en *Simulink* para posicionar el péndulo invertido en vertical.

## 6. Demo - Paso a Paso

Para facilitar la verificación de los resultados de simulación mostrados en los apartados 3 y 4, se ha realizado una breve guía paso a paso:

1. Abrir y ejecutar el archivo *demo.m* para cargar las variables necesarias en el espacio de trabajo.
2. Ejecutar el archivo de proyecto *Control\_Pendulo\_invertido.prj* para mostrar los modelos disponibles. Los modelos terminados en "\_test" son para simulación, mientras que los terminados en "\_final" son para ejecución en *Arduino*.

3. Abrir el modelo *pid.test.slx* y ejecutar la simulación. Se obtendrán los resultados mostrados en el [Apartado 3.1](#). (respuesta ante una perturbación del sistema con regulador PID).
4. Abrir el modelo *asig\_directa.test.slx* y ejecutar la simulación. El modelo contiene distintas combinaciones de reguladores y observadores obtenidos por asignación directa de polos y por optimización de la función de coste. Se podrá observar la respuesta ante perturbaciones de los distintos reguladores abriendo el "display" a la salida de cada uno.
5. Abrir el modelo *servo.test.slx* y ejecutar la simulación. Se mostrarán los resultados correspondientes al [Apartado 3.2.3](#). (Respuesta del servoregulador ante señal de referencia y condiciones iniciales no nulas).

## 7. Implementación de los Reguladores

### 7.1. PID Clásico

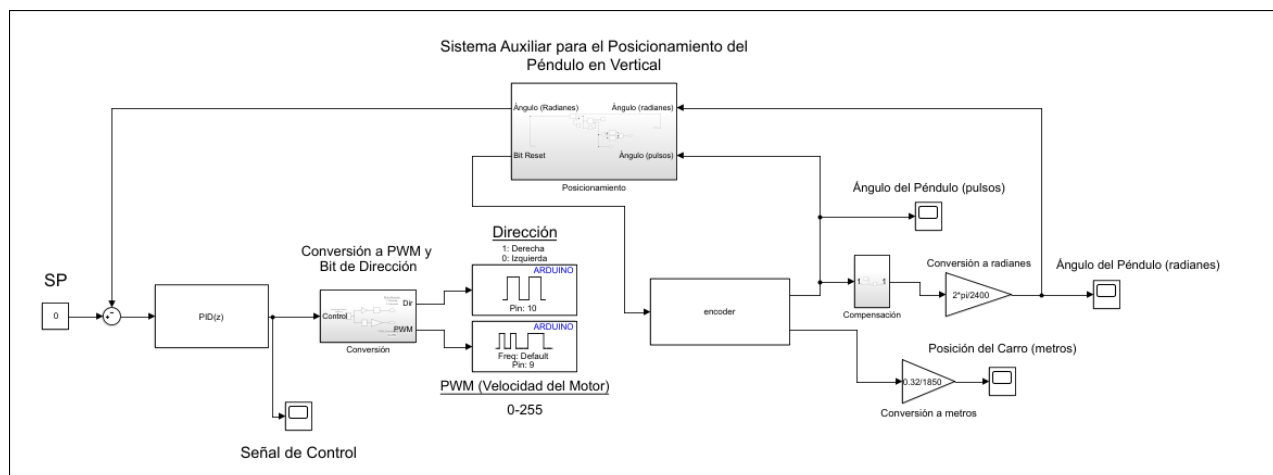


Figura 23: Implementación del PID en *Simulink* para su ejecución en *Arduino*.

En la Figura 23 se puede observar que se ha hecho uso de un sistema auxiliar para mantener el sistema en cadena abierta hasta que el péndulo sea posicionado en vertical. Dado que se ha hecho uso de encoders incrementales, es importante tener una medida precisa y bien referenciada del ángulo del péndulo respecto a la posición de reposo.

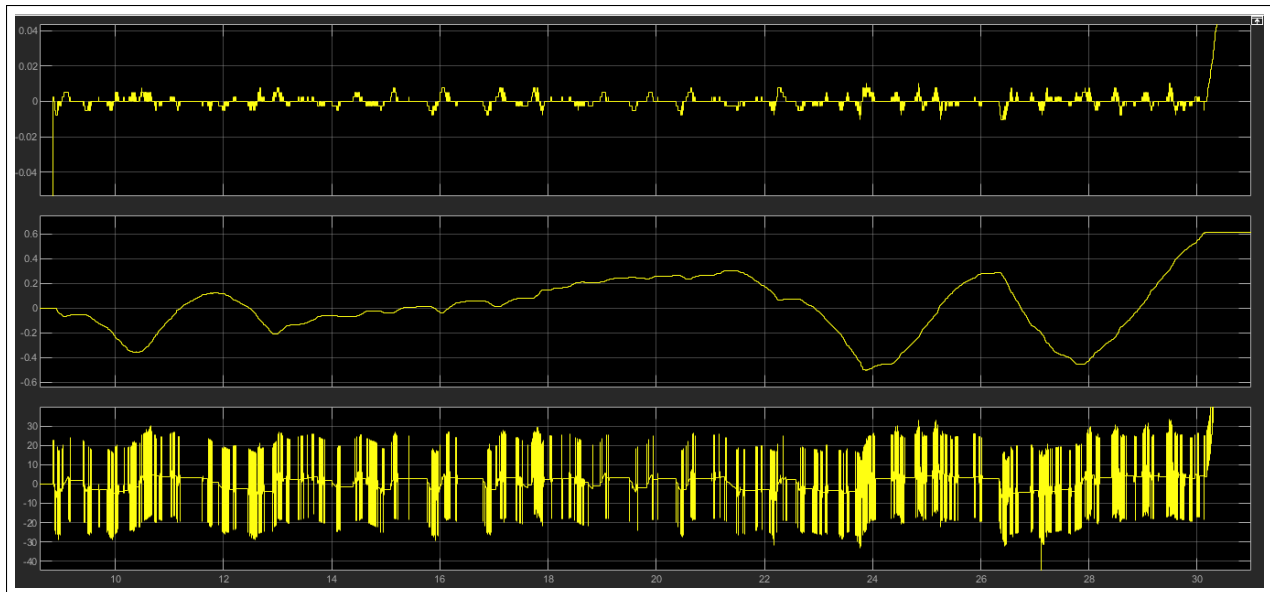


Figura 24: Resultados experimentales - Implementación PID. En orden descendente: ángulo del péndulo, posición de el carro y señal de control.

## Observaciones

Los resultados obtenidos con el PID han sido muy satisfactorios. La respuesta del regulador es muy rápida y ante perturbaciones el ángulo del péndulo apenas parece variar. Como era de esperar, con un solo regulador PID estamos limitados al control de una única variable de salida, por lo que la posición del carro ha sido ignorada. En aplicaciones en las que lo relevante es estabilizar una única variable de salida, un regulador PID cumple el objetivo con creces y además exige un diseño rápido y sencillo. Sin embargo, en casos en los que la carrera del carro es limitada; es decir, es relevante controlar una segunda variable de salida; un simple regulador PID puede no ser suficiente.

## 7.2. Regulador por Asignación Directa de Polos y Estimador de Kalman

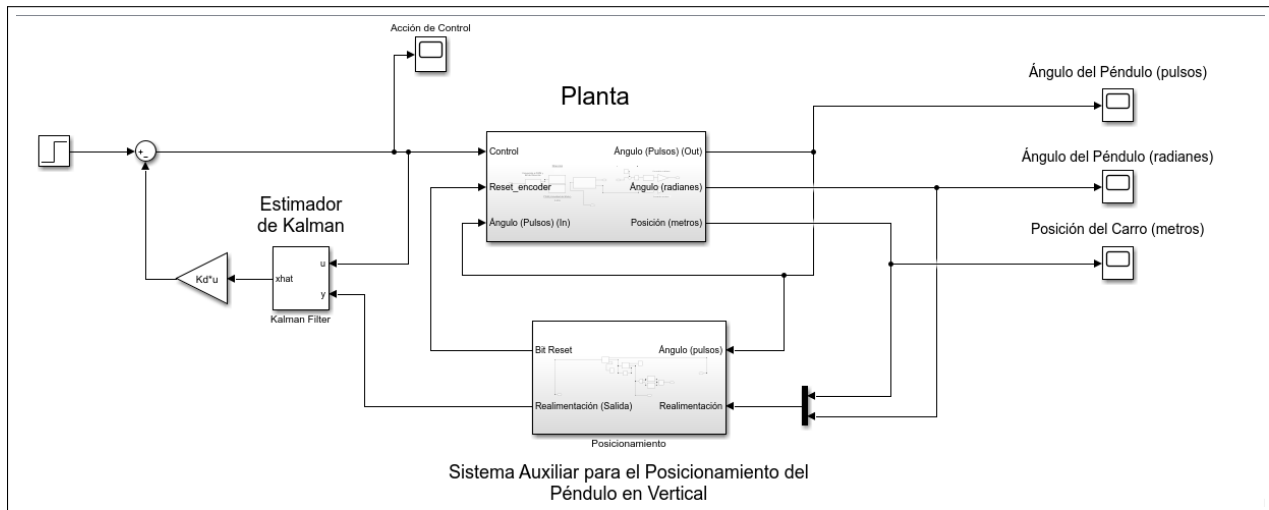


Figura 25: Implementación en *Simulink* del regulador y observador para su ejecución en *Arduino*. El regulador se ha obtenido por asignación directa de polos, mientras que el observador se ha implementado mediante un filtro de Kalman.

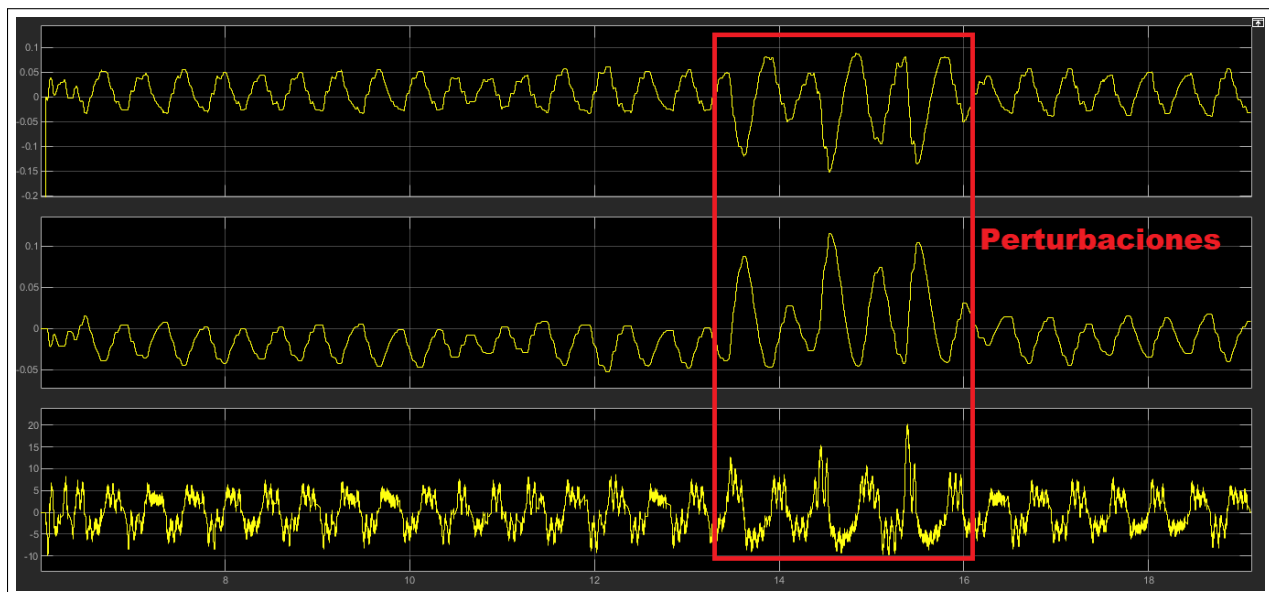


Figura 26: Resultados experimentales - Regulador por Asignación Directa y Estimador de Kalman. En orden descendente: ángulo del péndulo, posición del carro y acción de control.

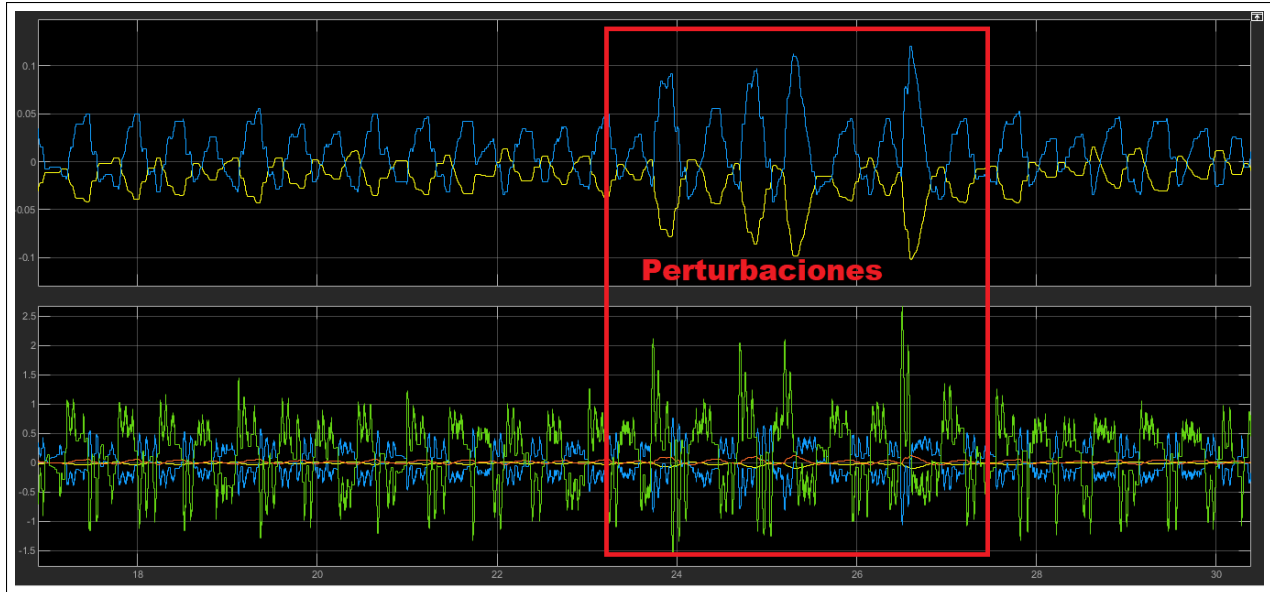


Figura 27: Resultados experimentales - Regulador por Asignación Directa y Estimador de Kalman. Salida del sistema (superior) frente al estado completo estimado mediante el fitro de Kalman (inferior).

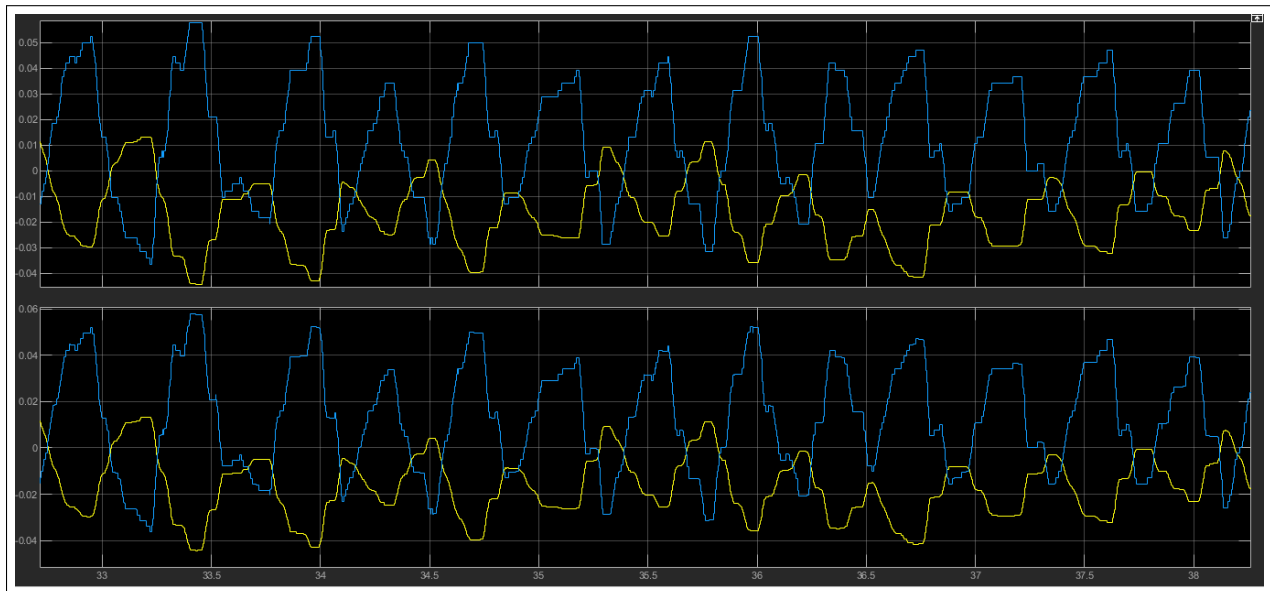


Figura 28: Resultados experimentales - Regulador por Asignación Directa y Estimador de Kalman. Salida del sistema (superior) frente a las variables correspondientes estimadas por el filtro de Kalman (inferior).

### 7.3. Servocontrolador Lineal Quadrático Gaussiano con Acción Integral (LQG)

Un controlador LQG está formado por un regulador LQ y un observador LQ, ambos óptimos. Para conformar el regulador LQG, se combinarán el servocontrolador óptimo y el filtro de Kalman diseñados en los apartados anteriores.

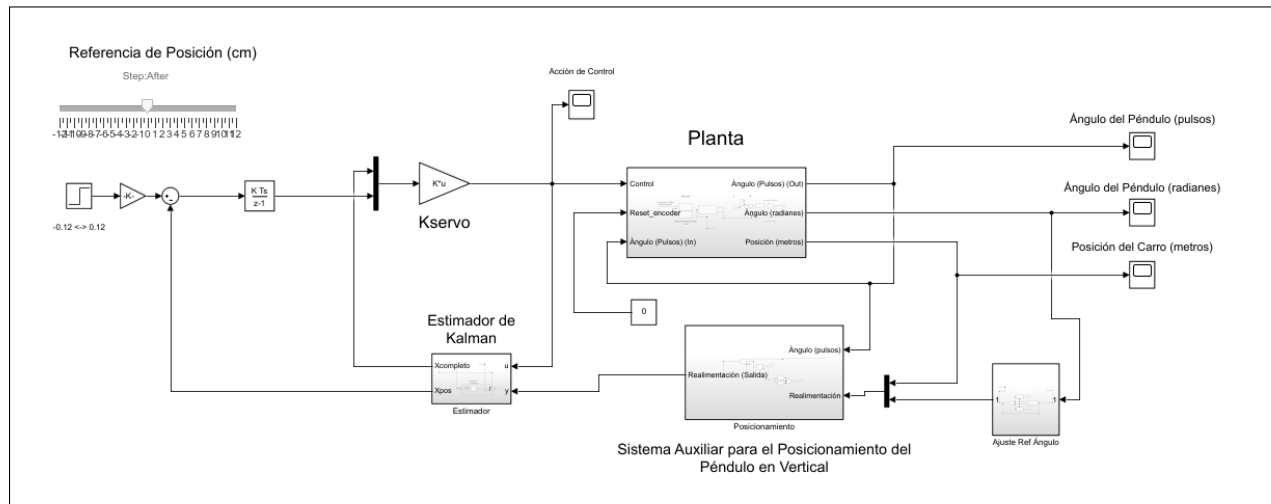


Figura 29: Servoregulador LQG implementado en *Simulink* y *Arduino* para controlar la planta física.

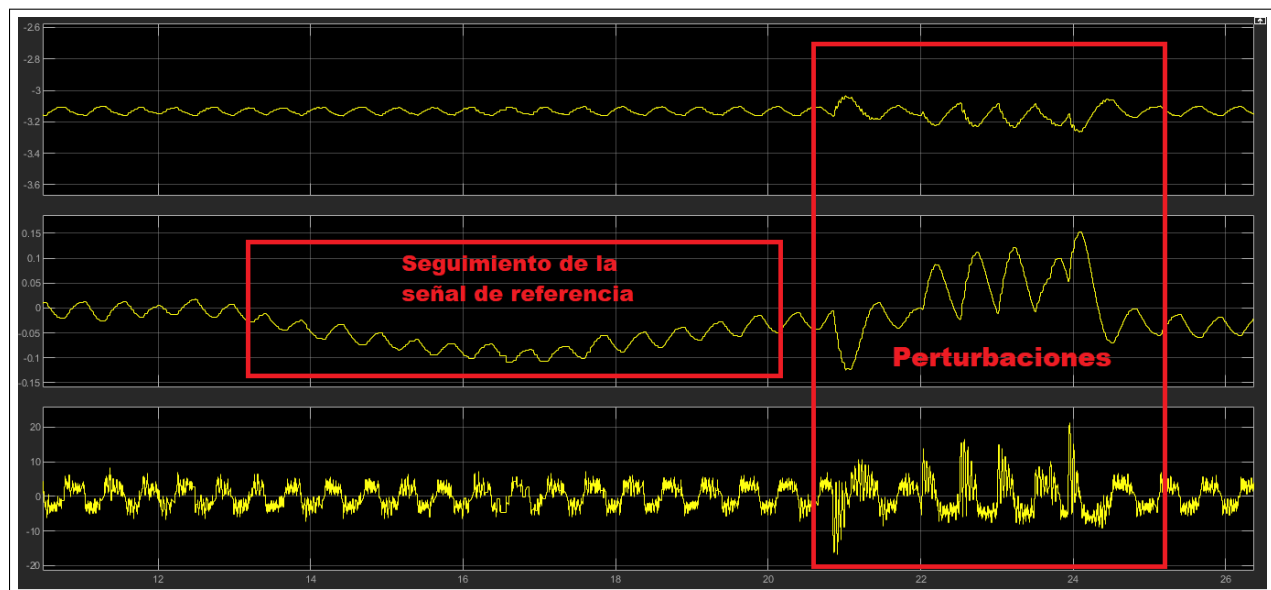


Figura 30: Resultados Experimentales - Servoregulador LQG. En orden descendente: ángulo del péndulo, posición del carro y señal de control.

## 7.4. Swing-Up y Servocontrolador LQG

Se coordinan ambos sistemas de control para que uno eleve el péndulo hasta la posición vertical y el otro lo estabilice cuando llegue.

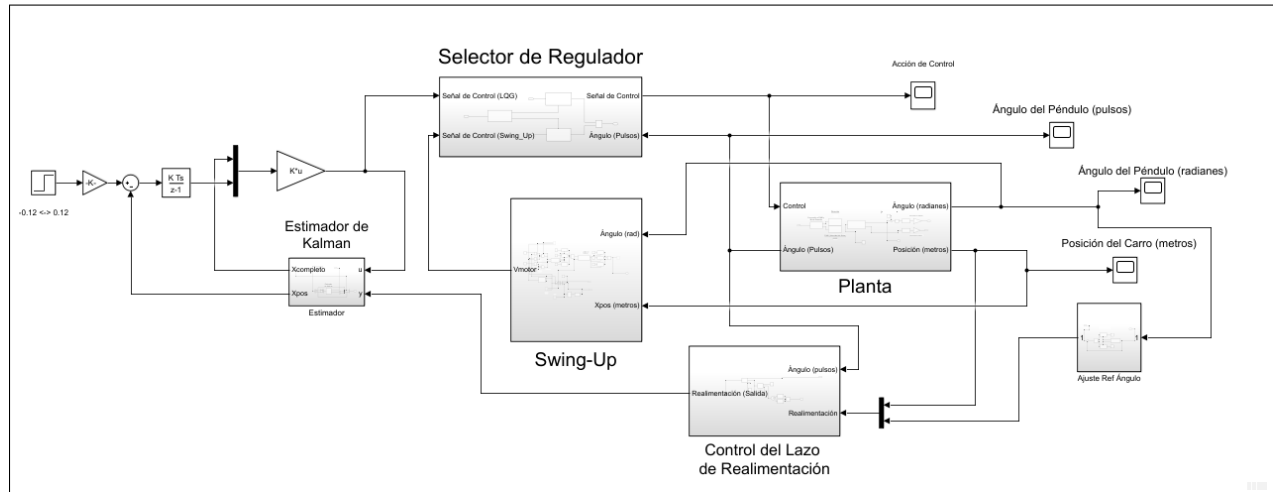


Figura 31: Sistema completo con controlador para el balanceo y Servoregulator LQG para la estabilización, implementados en *Simulink* para su ejecución en *Arduino*.

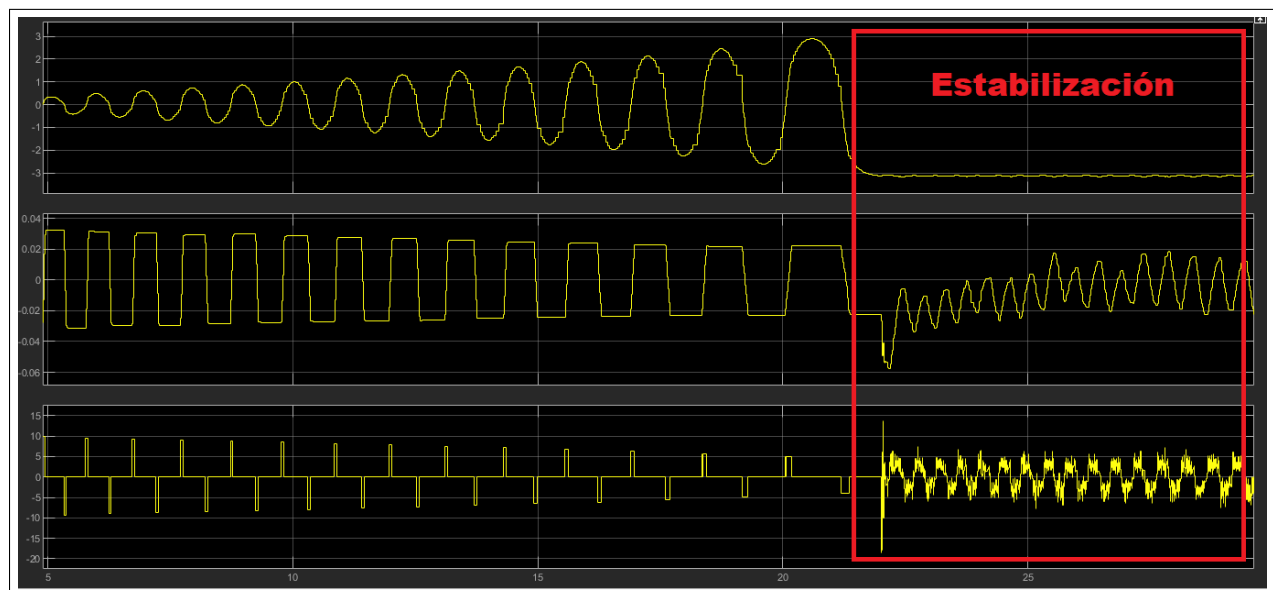


Figura 32: Resultados experimentales - Balanceo y Estabilización con LQG. En orden descendente: ángulo del péndulo, desplazamiento del carro y señal de control.



## 8. Conclusiones

Se ha podido comprobar la versatilidad y robustez de los reguladores diseñados con técnicas del espacio de estados. A pesar de que el modelado del sistema en ecuaciones de estado y el diseño de los reguladores es más trabajoso, como ventaja se obtiene la posibilidad de trabajar con sistemas MIMO e implementar una lógica de control más potente para sistemas de orden superior y con dinámicas más complejas. En contrapartida, el regulador PID clásico exige menos trabajo en el diseño y ofrece buenos resultados a pesar de sus limitaciones. Cuando las exigencias del diseño lo permitan, se optará por un regulador clásico debido a su sencillez.

Aunque operacionalmente las técnicas de diseño óptimo puedan parecer, en un primer momento, más trabajosas que el diseño por asignación directa de polos; la verdad es que al emplear programas de cálculo numérico pasa a suceder justo lo contrario. La elección de las matrices  $Q$  y  $R$  resulta realmente muy intuitivo, y la solución de la ecuación de *Riccati* puede ser fácilmente obtenida computacionalmente. Especialmente para sistemas de orden superior, la asignación directa de polos pasa a ser un ejercicio de ensayo y error, y la elección de los polos puede parecer, por momentos, algo arbitraria. Sin embargo, está claro que para sistemas de orden inferior, el diseño por asignación directa de polos suele ofrecer resultados más que aceptables.

Durante el desarrollo del proyecto ha resultado aparente la gran importancia de realizar un buen modelado del sistema a controlar. En un primer momento, se optó por hacer un modelado muy simplificado del motor de CC. Como consecuencia, el comportamiento del modelo no era lo suficientemente fiel al del sistema físico, y los reguladores que funcionaban correctamente en simulaciones no lo hacían tan bien con el sistema real. Un modelado más minucioso del motor mejoró significativamente los resultados.

A la hora de coordinar el algoritmo de balanceo con el controlador, se descubrió un nuevo inconveniente causado por la carrera reducida del carro. El péndulo alcanzaba la posición vertical con demasiada inercia y al carro le faltaba recorrido para poder compensar dicho impulso. Como solución, se mejoró el algoritmo de balanceo, haciendo que a medida que el péndulo se aproximase a la posición vertical, el empuje del carro se fuese haciendo más débil. De esta forma, se logró que el péndulo alcanzase la posición deseada con mucha menos inercia y el regulador pudiese estabilizar el mismo con la carrera disponible.

## Referencias

- [1] Sergio Domínguez, Pascual Campoy, José María Sebastián, y Agustín Jiménez. *Control en el Espacio de Estado*. Pearson Educación S.A., Madrid, 2006.
- [2] Carlos Platero Dueñas. *Apuntes de Regulación Automática*. Dpto. Electrónica, Automática e Informática Industrial. Escuela Técnica Superior de Ingeniería y Diseño Industrial, Universidad Politécnica de Madrid, 2014. Disponible en: <http://www.ieef.upm.es/moodle/course/view.php?id=521>

- [3] Jitendra Singh. *Model Based Control Design for Swing-up & Balance the Inverted Pendulum*, SardarVallabhbhai National Institute of Tecnology (SVNIT), SURAT, 2018.
- [4] Stefan Brock. *Swing-Up Methods For Inverted Pendulum*, Conference Paper, Poznan University of Technology, Poland, 2003.
- [5] Giampero Campa. *Writing a Simulink Device Driver block: a step by step guide*. The MathWorks, Inc, 2015.
- [6] Steven L. Brunton. *Control Bootcamp*, University of Washington, 2017-.  
Serie de videos disponibles en:  
<https://www.youtube.com/playlist?list=PLMrJAKhIeNNR20Mz-VpzgfQs5zrYi085m>