



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Tecnológico
de Monterrey

DISEÑO DE UN CONTROLADOR HÍBRIDO PREDICTIVO ADAPTIVO PARA LA SUBIDA Y ESTABILIZACIÓN DE UN PÉNDULO INVERTIDO

Alumno: Javier Canales Martínez

Asesor: Dr. Antonio Favela Contreras
Coasesor: Rafael Camilo Lozoya-Gamez

Departamento de Mecatrónica y Eléctrica

Junio, 2020



Índice de contenido

1	Resumen.....	1
2	Introducción al problema y objetivos	2
3	Sistema del péndulo invertido.....	4
3.1	Descripción del sistema	4
3.2	Componentes	4
3.3	Ecuaciones del movimiento	6
4	Identificación del sistema.....	9
4.1	Modelo ARX.....	9
4.2	Diseño del ensayo.....	10
4.3	Criterio de decisión (LMS)	11
4.4	Resultados obtenidos y validación	13
5	Diseño del algoritmo de control basado en modelo.....	15
5.1	Enfoque de control	15
5.2	Cálculo de la acción de control: estructura de control propuesta	17
5.3	Bloque conductor: controlador híbrido	18
5.3.1	Estado 1: Levantamiento	20
5.3.2	Estado 2: Estabilización.....	20
5.4	Simulación y ajuste de parámetros	21
5.5	Implementación del controlador.....	24
5.6	Validación y desempeño del controlador	25
6	Ampliaciones	27
6.1	Controlador adaptivo.....	27
6.1.1	Mecanismo de adaptación: Mínimos Cuadrados Recursivos.....	27
6.1.2	Implementación y validación del mecanismo de adaptación	28
6.1.3	Control del sistema modificado	31
6.1.4	Limitaciones	33
6.2	Control embebido en un microcontrolador	34
6.2.1	Equipo utilizado	34
6.2.2	Simulación: Hardware-in-the-loop.....	36
6.2.3	Control del sistema real	37
7	Conclusiones y futuras oportunidades de investigación.....	39
8	Bibliografía.....	40

Índice de figuras

Fig 1. Sistema péndulo invertido y variables de equilibrio.	4
Fig 2. Fotografía 1 del sistema real.	5
Fig 3. Fotografía 2 del sistema real.	5
Fig 4. Parámetros del sistema.	6
Fig 5. Diagrama de fuerzas péndulo.	7
Fig 6. Diagrama de fuerzas del carrito.	7
Fig 7. Señal PRBS y sus parámetros.	10
Fig 8. Diagrama de bloques en Simulink para la validación de los modelos ARX.	13
Fig 9. Validación de los modelos ARX.	13
Fig 10. Esquema de la filosofía de control.	16
Fig 11. Esquema de control propuesto.	18
Fig 12. Estados del sistema: levantamiento y estabilización.	19
Fig 13. Esquema de control durante el levantamiento.	20
Fig 14. Esquema de control durante la estabilización.	20
Fig 15. Diagrama de bloques en Simulink para la simulación del controlador.	22
Fig 16. Simulación del control del sistema sin control cascada.	23
Fig 17. Simulación control del sistema con control cascada.	23
Fig 18. Pestaña de monitoreo y control.	24
Fig 19. Validación del controlador sobre el sistema real.	25
Fig 20. Pestaña del mecanismo de adaptación en el instrumento virtual.	29
Fig 21. Fotografía del sistema modificado.	30
Fig 22. Validación de los coeficientes obtenidos para el sistema modificado.	31
Fig 23. Control sistema modificado sin mecanismo de adaptación.	32
Fig 24. Control sistema modificado con mecanismo de adaptación.	32
Fig 25. Fotografía del sistema embebido utilizado.	35
Fig 26. Esquema Hardware-in-the-loop.	36
Fig 27. Control del sistema simulado mediante el microcontrolador (Hardware-in-the-loop).	37
Fig 28. Validación del sistema embebido.	38

1 Resumen

Este documento trata sobre uno de los problemas clásicos de la teoría de control: el levantamiento y la estabilización de un péndulo invertido sobre un carro. El enfoque propuesto para lograr este objetivo es un algoritmo de control híbrido predictivo y no lineal basado en modelos ARX. Para ello, las prestaciones del algoritmo se han testado en simulación y validado sobre una maqueta de un péndulo invertido sobre un carro del departamento de mecatrónica del TEC de Monterrey.

El documento está estructurado de la siguiente forma:

En los capítulos 1 y 2 se contextualiza el problema con: una breve descripción del sistema, el objetivo de control y el equipo que se ha utilizado. Seguidamente, se plantean las ecuaciones diferenciales que describen su comportamiento desde el punto de vista de la mecánica Newtoniana.

El capítulo 3 inicia con la discretización de dichas ecuaciones para llegar a los modelos ARX en los que se basa el algoritmo de control. Continúa con la identificación de sus coeficientes mediante el algoritmo de Mínimos Cuadrados y, finaliza, con la validación los modelos sobre el sistema real. De esta forma, se concluye el capítulo con un modelo no lineal del sistema en ecuaciones de diferencias en el que se basa el controlador predictivo propuesto en el presente trabajo.

En el capítulo 4 del documento, se explica, detalladamente, el algoritmo de control que se ha utilizado dividiéndolo en dos partes: el cálculo de la acción de control a través de un modelo ARX no lineal que conducirá a la estructura de control propuesta y, el bloque conductor que deriva en la necesidad de emplear un controlador híbrido. Luego, por medio de la simulación, se evalúa el comportamiento del controlador y, al final del capítulo, se muestra su implementación en un programa de LabView y los resultados obtenidos durante la experimentación sobre el sistema real así como su desempeño.

En el capítulo 5 se muestran las dos ampliaciones que se han realizado sobre el trabajo inicial: el diseño de un mecanismo de adaptación a los cambios en los parámetros del sistema y el control del péndulo invertido a través de un microcontrolador con un sistema operativo de tiempo real en el que se ha embebido el algoritmo de control.

Finalmente, en el capítulo 6, aparecen las conclusiones que se han extraído de todo lo anterior y las futuras áreas de investigación para trabajos posteriores.

2 Introducción al problema y objetivos

El péndulo invertido es uno de los problemas clásicos de la teoría de control. Se trata de un sistema que, a pesar de presentar una mecánica poco compleja, se caracteriza por ser: trivialmente poco activo (subactuado), inestable y no lineal [1]. Estas características dinámicas, lo han convertido en un punto de referencia para comparar las prestaciones de diversos algoritmos de control. Su estudio también ha contribuido al desarrollo de varias tecnologías como: el equilibrio de robots, máquinas de transporte que necesitan balancear objetos, vehículos aéreos no tripulados (UAVs) o vehículos comerciales como el Segway, entre otras.

Aunque este trabajo se centre en el levantamiento y la estabilización de un péndulo simple, existen varias configuraciones posibles: péndulo de Furuta (rotacional), péndulo tradicional (lineal), péndulo 3D, péndulo doble, etc. En estos sistemas subactuados se han probado diversas estrategias de control. Técnicas de control óptimo NMPC sobre péndulos triples [2] que ofrecen grandes prestaciones pero presentan una carga computacional elevada. Otros enfoques [3] [4] consiguen reducir esta carga computacional mediante la linealización del modelo por retroalimentación y cambio de controladores. También, se han propuesto otras técnicas basadas en el control tipo Fuzzy para el levantamiento y estabilización del péndulo invertido [5].

En este trabajo se propone el levantamiento y estabilización del péndulo invertido sobre un carro mediante un controlador híbrido predictivo no lineal. Por un lado, se clasifica como predictivo porque, como se verá más adelante, hace uso de modelos ARX no lineales del sistema para el cálculo de la acción de control. Y, por el otro, se habla de híbrido porque dependiendo en el estado en el que se encuentre el sistema (levantamiento o estabilización), se aplicará una estrategia de control determinada.

Por tanto, el objetivo general de este trabajo es:

- Desarrollar una estrategia de control predictivo híbrido no lineal para un péndulo invertido sobre un carro.

Y, como objetivos secundarios, que derivan del anterior, se tienen:

- Obtención y discretización de un modelo paramétrico de ecuaciones diferenciales que describa el comportamiento dinámico del sistema.
- Identificación y validación de los coeficientes del modelo discreto.
- Evaluación del comportamiento del controlador diseñado mediante simulación, su implementación en un instrumento virtual de LabView y medición de su desempeño sobre el sistema real mediante indicadores.
- Diseño de un mecanismo de adaptación ante variaciones en el comportamiento del péndulo invertido sobre un carro.



- Embeber el algoritmo de control en un microcontrolador que ejecuta un sistema operativo en tiempo real.

3 Sistema del péndulo invertido

Este capítulo comienza con una breve descripción del sistema para comprender su funcionamiento y, también se explica en qué consiste el objetivo de control que se persigue. A continuación, se detalla la función y las características del conjunto de componentes que integran el sistema real. Y, para acabar, se deducen, desde el enfoque de la mecánica Newtoniana, las ecuaciones diferenciales que rigen la dinámica del péndulo invertido sobre un carro.

3.1 Descripción del sistema

De todas las configuraciones existentes de péndulo invertido, este trabajo se centra en el control de un péndulo invertido sobre un carro Fig 1.

El sistema consta de un carrito que se desplaza linealmente sobre un riel en ambos sentidos gracias a la acción de una fuerza motriz (un motor eléctrico en este caso). A dicho carrito va unido, solidariamente, un eje de rotación sobre el que pivota un péndulo. Este péndulo cuenta con dos puntos de equilibrio: uno estable ($\theta_{eq,1} = \pi$) y otro inestable ($\theta_{eq,2} = 0$). El objetivo que se persigue es: estabilizar el péndulo en su punto de equilibrio inestable $\theta_{eq,2}$, partiendo de su punto de equilibrio estable $\theta_{eq,1}$. Para ello, la única acción de control (U) que se tiene es el motor eléctrico que hace que el carrito se desplace en un sentido u otro.

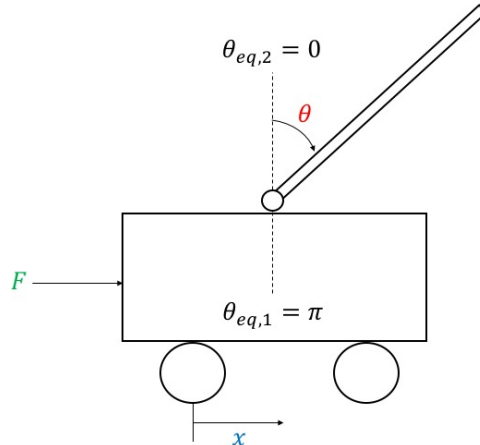


Fig 1. Sistema péndulo invertido sobre un carro y variables de equilibrio.

Donde:

x : La posición del carrito con respecto al eje de referencia fijo.

θ : Es el ángulo que forma el péndulo con respecto de la línea vertical.

F : Es la fuerza que ejerce el motor sobre el carrito.

$\theta_{eq,1}$: Es el punto de equilibrio estable del péndulo.

$\theta_{eq,2}$: Es el punto de equilibrio inestable del péndulo.

3.2 Componentes

A continuación, se describen el conjunto de elementos que componen el sistema real:

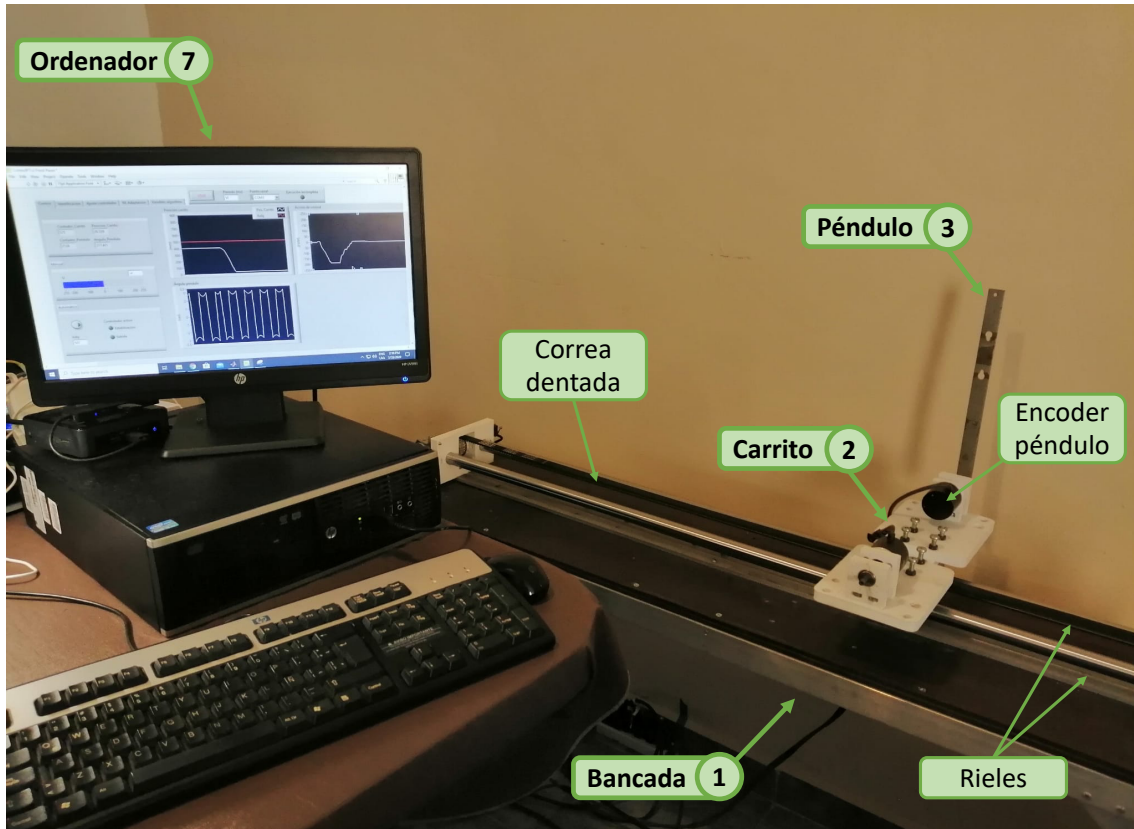


Fig 2. Fotografía 1 del sistema real.

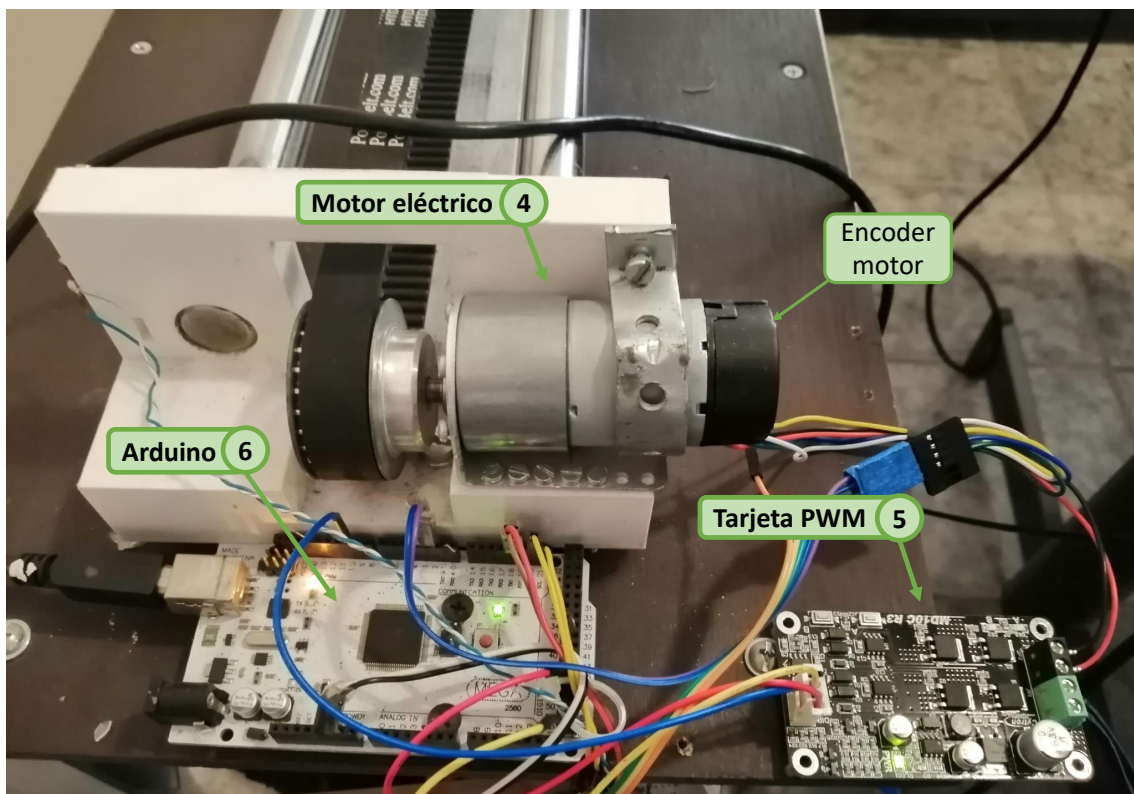


Fig 3. Fotografía 2 del sistema real.

1. Bancada: soporte que sustenta al resto de elementos y sobre el que se han colocado dos rieles horizontales por los que se desplaza el carrito linealmente.
2. Carrito: sólido rígido que se desplaza sobre los rieles en ambos sentidos gracias a la fuerza motriz que le transmite el motor eléctrico por medio de una correa dentada.
3. Péndulo: varilla metálica que pivota sobre un eje unido solidariamente al carrito. El ángulo θ se mide a través de un encoder incremental de cuadratura de efecto Hall.
4. Motor eléctrico 12(V): Es el encargado de mover el carrito. Su movimiento rotacional se transmite al carrito por medio de una correa dentada flexible. El motor es de corriente continua y se alimenta a través de un circuito electrónico que genera una señal PWM (U). Además, al igual que el péndulo, también dispone de un encoder incremental que permite conocer la posición del carrito (x).
5. Tarjeta PWM (MD10C R3): capaz de generar señales PWM bipolares. Para ello, es necesario indicarle la amplitud y la polaridad de la señal deseada en sus pines de entrada PWM y DIR, respectivamente. Esto se hace a través de el Arduino Mega 2560. La amplitud máxima de estas señales deberá ser igual a la tensión nominal del motor, por ello se conecta a una fuente de alimentación de 12(V).
6. Arduino Mega 2560: funciona como una tarjeta de adquisición de datos (TAD) que permite la comunicación entre el sistema y el ordenador. Lee los valores de los encoders tanto del péndulo como del motor y actúa sobre el motor eléctrico.
7. Ordenador: necesario para el diseño y simulación del controlador y la ejecución del instrumento virtual de LabView Fig 18 utilizado para la identificación, control y monitoreo del sistema.

3.3 Ecuaciones del movimiento

En esta sección se calcularán las ecuaciones diferenciales que describen el movimiento del péndulo invertido sobre un carro. Para ello, se aplica el enfoque basado en la dinámica Newtoniana.

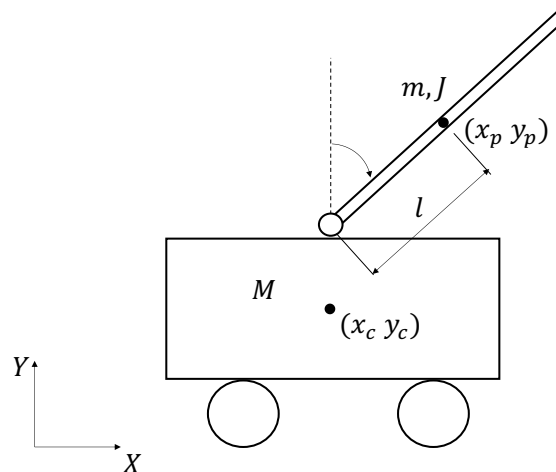


Fig 4. Parámetros del sistema.

En la Fig 4 se pueden observar los parámetros del sistema, siendo:

M : Masa del carrito.

m : Masa del péndulo

x_i : Posición en el eje X del centro de gravedad del carrito (c) y el péndulo (p).

y_i : Posición en el eje Y del centro de gravedad del carrito (c) y el péndulo (p).

J : Momento de inercia del péndulo con respecto de su centro de gravedad.

l : Distancia desde el eje de giro al centro de gravedad del péndulo.

- Para el péndulo:

<p>Fig 5. Diagrama de fuerzas péndulo.</p>	$x_{cg} = x + l \sin(\theta)$
	$y_{cg} = l \cos(\theta)$
	$\sum F_x = m\ddot{x}_{cg} \rightarrow -H = m \cdot (\ddot{x} + l \cos(\theta) \ddot{\theta} - l \sin(\theta) \dot{\theta}^2)$
	$\sum F_y = m\ddot{y}_{cg} \rightarrow -V - mg = m \cdot (l \cos(\theta) \dot{\theta}^2 + l \sin(\theta) \ddot{\theta})$
$\sum T = (J + ml^2)\ddot{\theta} \rightarrow -Vl \sin(\theta) + Hl \cos(\theta) - b_1\dot{\theta} = (J + ml^2)\ddot{\theta}$	

$(J + ml^2)\ddot{\theta} = mgl \sin(\theta) - ml \cos(\theta) \ddot{x} - b_1\dot{\theta}$	(1)
---	-----

- Para el carrito:

<p>Fig 6. Diagrama de fuerzas del carrito.</p>	$\sum F_x = M\ddot{x} \rightarrow F + H - b_0\dot{x} = M\ddot{x}$
--	---

$(M + m)\ddot{x} = ml \sin(\theta) \dot{\theta}^2 - ml \cos(\theta) \ddot{\theta} + u - b_0\dot{x}$	(2)
---	-----

Donde, en la Fig 5 y la Fig 6:

H : componente horizontal de la reacción que se genera en la unión articulada.

V : componente vertical de la reacción que se genera en la unión articulada.

b_0 : coeficiente de rozamiento de carrito.

b_1 : coeficiente de rozamiento del péndulo.

g : aceleración de la fuerza de la gravedad

N : componente normal ejercida contra el carrito.



4 Identificación del sistema

El algoritmo de control que se ha utilizado se basa en un modelo ARX del sistema que ha sido obtenido a partir de las ecuaciones diferenciales que rigen el comportamiento dinámico del mismo.

En este capítulo, se describen los pasos que se han seguido para la obtención y validación de dicho modelo. Para ello, se comienza con la discretización de las ecuaciones dinámicas (1) (2) llegando así a un modelo ARX (ecuaciones en diferencias). Seguidamente, con el objetivo de obtener los valores de los coeficientes del modelo, se lleva a cabo un ensayo sobre el sistema real que permite recoger un conjunto de valores de entrada-salida y, aplicar el algoritmo de mínimos cuadrados (LMS) a estos datos. Finalmente, se procede a la validación del modelo obtenido comparando su respuesta con la del sistema real en un segundo ensayo de validación.

4.1 Modelo ARX

Para la discretización de las ecuaciones diferenciales del sistema se ha seleccionado el método de Euler de entre los múltiples existentes en la literatura porque el periodo de muestreo (del que se hablará con más detalle en la siguiente sección) que se ha seleccionado y el ruido presente en el sistema están por debajo de los máximos admisibles.

La aproximación discreta para una variable genérica q , según este método, viene dada por:

$$\dot{q}_k = \frac{q_k - q_{k-1}}{T_s} \quad (3)$$

$$\ddot{q}_k = \frac{q_k - 2q_{k-1} + q_{k-2}}{T_s^2} \quad (4)$$

- **Para el péndulo:**

Tomando la ecuación (1) como punto de partida, y asumiendo que el término del rozamiento ($b_1 \dot{\theta}$) es despreciable. Se tiene:

$$(J + ml^2) \frac{\theta_k - 2\theta_{k-1} + \theta_{k-2}}{T_s^2} = mgl \sin(\theta_{k-1}) - ml \cos(\theta_{k-1}) \frac{x_k - 2x_{k-1} + x_{k-2}}{T_s^2} \rightarrow$$

$$\theta_k = a_{p,1} \theta_{k-1} + a_{p,2} \theta_{k-2} + a_{p,3} \sin(\theta_{k-1}) - b_{p,1} \cos(\theta_{k-1}) a_k \quad (5)$$

Siendo:

θ_i : posición angular del péndulo en el instante i .

$a_{p,j}$: coeficientes que acompañan a la variable endógena del modelo ARX.

$b_{p,j}$: coeficientes que acompañan a la variable exógena del modelo ARX.

a_k : aceleración en el instante actual definida como $a_k = x_k - 2x_{k-1} + x_{k-2}$

- **Para el carrito:**

Partiendo de la ecuación (2), y asumiendo que M es lo suficientemente más grande que m como para despreciar los términos $(ml \sin(\theta) \dot{\theta}^2 - ml \cos(\theta) \ddot{\theta})$. Queda:

$$(M + m) \frac{x_k - 2x_{k-1} + x_{k-2}}{T_s^2} = -b_0 \frac{x_k - x_{k-1}}{T_s} + u_{k-1} + u_{k-2}$$

$x_k = a_{c,1}x_{k-1} + a_{c,2}x_{k-2} + b_{c,1}u_{k-1} + b_{c,2}u_{k-2}$	(6)
---	-----

Siendo:

x_i : posición del carrito en el instante i .

$a_{p,j}$: coeficientes que acompañan a la variable endógena del modelo ARX.

$b_{p,j}$: coeficientes que acompañan a la variable exógena del modelo ARX.

4.2 Diseño del ensayo

Una vez se tienen los modelos ARX (5) (6), es necesario identificar sus coeficientes (a_i b_i). Para ello, se decide hacer un ensayo sobre el sistema real alimentando el motor eléctrico con una señal PRBS como acción de control y, con los datos obtenidos, aplicar el algoritmo de mínimos cuadrados para el cálculo de a_i y b_i (siguiente sección).

La señal PRBS posee unas características similares a las del ruido blanco: presenta amplitud casi constante en todas las frecuencias [6]. De manera que, alimentando al sistema con esta señal se puede observar su respuesta en todo el espectro frecuencial. Por ese motivo, los resultados de este ensayo contienen toda la información dinámica del sistema y se pueden considerar adecuados para el proceso de identificación. A la hora de diseñar el ensayo se deben decidir los dos parámetros que caracterizan esta señal: ancho mínimo de pulso y amplitud Fig 7.

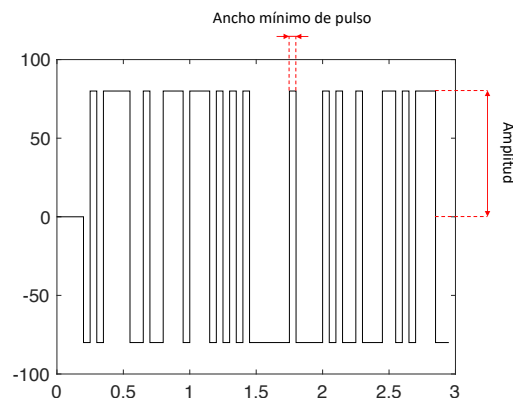


Fig 7. Señal PRBS y sus parámetros.

La forma de escoger un ancho mínimo de pulso adecuado es observando el ancho de banda de la acción de control que provoca cambios en el sistema. Es decir, existe un límite máximo en la frecuencia de cambio de las acciones de control a partir de el cual el sistema filtra esas acciones no actuando en consecuencia. Para determinar dicho límite, basta con introducir acciones de control en forma de trenes de pulsos con una amplitud moderada (lo que generará un movimiento alternativo del carrito) e ir aumentando la frecuencia de esta señal hasta que el sistema comience a filtrar la señal (deje de producirse el movimiento de vaivén). Haciendo este experimento, se determina que una frecuencia de cambio en las acciones de control adecuada para evitar el filtrado es de 20Hz ($T_s = 0.05$ s). Lo que determina el periodo de muestreo del experimento y, por consiguiente, el periodo de control (ya que los coeficientes del modelo dependen del T_s que se escoja).

Por lo que respecta a la amplitud de la PRBS, se escoge de acuerdo con el rango de actuación de la acción de control. En este caso $U \in [-255, 255]$. Por un lado, una amplitud demasiado grande podría dañar los componentes del sistema y, por otro, un valor demasiado pequeño generaría movimientos inapreciables. Así, se acaba escogiendo una amplitud de 80.

4.3 Criterio de decisión (LMS)

Una vez se tiene un conjunto de datos del experimento (y_i y u_i), y los modelos que se quieren identificar (5) (6), ya se puede aplicar el algoritmo de mínimos cuadrados para la identificación de los coeficientes. Se escoge este método por los buenos resultados que proporciona para sistemas poco ruidosos (como es el caso) y por ser válido para modelos no lineales [7].

Partiendo de los modelos ARX que tienen la forma:

$$\hat{y}_k = a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_{na} y_{k-na} + b_1 u_{k-1} + b_2 u_{k-2} + \dots + b_{nb} u_{k-nb} \quad (7)$$

Se construyen las ecuaciones normales que utiliza el método como sigue:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_{na} y_{k-na} + b_1 u_{k-1} + b_2 u_{k-2} + \dots + b_{nb} u_{k-nb} + e_k \quad (8)$$

Que expresadas en forma matricial quedan:

$$y_k = \psi_k^T \Theta + e_k \quad (9)$$

Donde:

- \hat{y}_k : salida estimada del sistema en el instante k.
- y_i : salida real del sistema en el instante i.
- u_i : entrada real del sistema en el instante i.
- a_i : coeficientes de la variable endógena en el modelo.

b_i : coeficientes de la variable exógena en el modelo.

e_k : error en el instante k. Calculado como la diferencia entre la salida real (y_k) y la estimada (\hat{y}_k).

ψ_k : vector de regresión en el instante k. $\psi_k = [y_{k-1} \ \dots \ y_{k-na} \ u_{k-1} \ \dots \ u_{k-nb}]^T$

Θ : vector de coeficientes del modelo. $\Theta = [a_1 \ a_2 \ \dots \ a_{na} \ b_1 \ b_2 \ \dots \ b_{nb}]^T$

Para el caso de los modelos ARX que se han obtenido, las ecuaciones normales se forman como sigue:

- **Para el péndulo**

$$\Psi_{p,k} = [\theta_{k-1} \ \theta_{k-2} \ \sin(\theta_{k-1}) \ \cos(\theta_{k-1}) \ a_k]^T$$

$$\Theta_p = [a_{p,1} \ a_{p,2} \ a_{p,3} \ b_{p,1}]^T$$

$\theta_k = [\theta_{k-1} \ \theta_k \ \sin(\theta_{k-1}) \ \cos(\theta_{k-1}) \ a_k][a_{p,1} \ a_{p,2} \ a_{p,3} \ b_{p,1}]^T + e_k$	(10)
---	------

- **Para el carrito**

$$\Psi_{c,k} = [x_{k-1} \ x_{k-2} \ u_{k-1} \ u_{k-2}]^T$$

$$\Theta_c = [a_{c,1} \ a_{c,2} \ b_{c,1} \ b_{c,2}]^T$$

$x_k = [x_{k-1} \ x_{k-2} \ u_{k-1} \ u_{k-2}][a_{c,1} \ a_{c,2} \ b_{c,1} \ b_{c,2}]^T + e_k$	(11)
--	------

Si se expresan los conjuntos de N ecuaciones normales de manera matricial, queda:

$Y_N = \Psi_N^T \Theta_N + e_N$	(12)
---------------------------------	------

Definiendo el índice (J) a minimizar como:

$J = \sum e_k^2 = e_N^T e_N$	(13)
------------------------------	------

Derivando J con respecto de θ_N se llega a la expresión para calcular el valor de los coeficientes de los modelos que minimiza la suma de los errores al cuadrado:

$\theta_N = (\Psi_N^T \Psi_N)^{-1} \Psi_N^T Y_N$	(14)
--	------

A partir de los datos experimentales del ensayo con la señal PRBS, aplicando el algoritmo de mínimos cuadrados, se llega a los siguientes modelos:

$\theta_k = 1,9932\theta_{k-1} - 0,9932\theta_{k-2} + 0,1368 \sin(\theta_{k-1}) - 0,0063 \cos(\theta_{k-1}) a_k$	(15)
--	------

$x_k = 1,3829x_{k-1} + 0,3829x_{k-2} + 0,0332u_{k-1} + 0,0386u_{k-2}$	(16)
---	------

4.4 Resultados obtenidos y validación

Por último, para finalizar este capítulo, se pasa a la validación de los dos modelos obtenidos (15) (16). Para ello, se vuelve a realizar otro ensayo con una señal PRBS, se guardan los datos y se comparan las respuestas del sistema real y de los modelos (simulada).

Para la comparación de las respuestas, se hace uso del siguiente diagrama de bloques de Simulink. Donde el bloque “Modelo PI” ejecuta las ecuaciones en diferencias (15) y (16) y los bloques “Posición real”, “Ángulo real” y “U” contienen, respectivamente, las posiciones del carrito, el ángulo del péndulo y las acciones de control del ensayo que se ha realizado sobre el sistema real. Por último, el bloque “Visualizador” permite comparar las respuestas real y simulada.

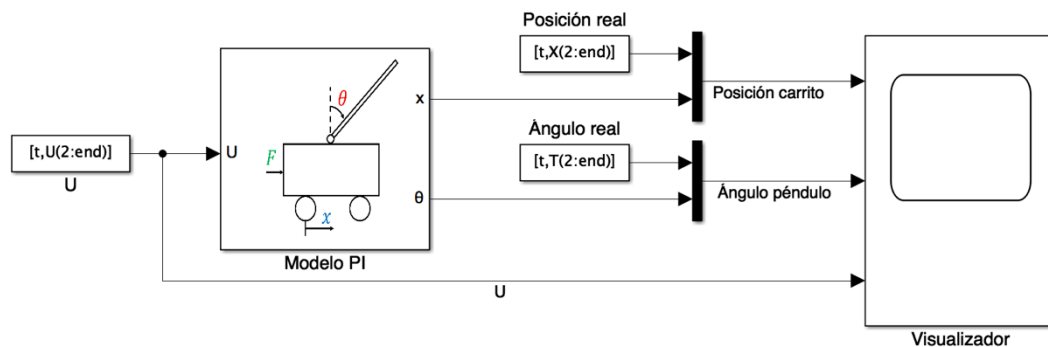


Fig 8. Diagrama de bloques en Simulink para la validación de los modelos ARX.

Los resultados obtenidos son:

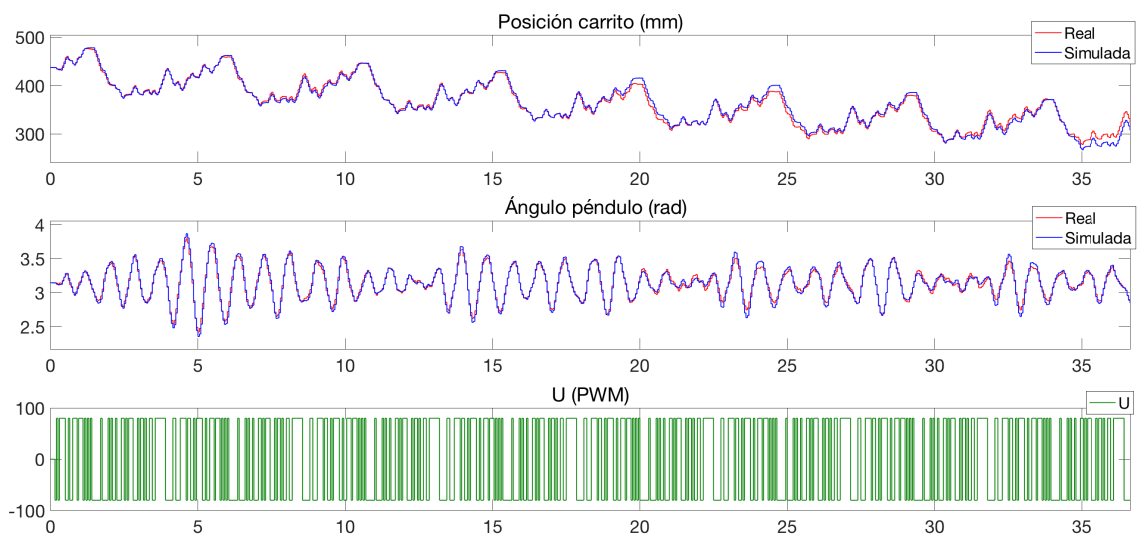


Fig 9. Validación de los modelos ARX.

Como se puede apreciar en la figura la respuesta del modelo se ajusta con bastante exactitud a la real en ambos casos. Además, a la hora de valorar estos resultados hay que tener en cuenta dos consideraciones:

1. En este caso, el bloque “Modelo PI”, a la hora de ejecutar las ecuaciones en diferencias para calcular el valor de las salidas en cada instante, utiliza los valores predichos de las salidas (\hat{y}_i) en lugar de los reales (y_i). Esto implica que los errores de predicción se vayan acumulando. Algo que no pasará durante el control del sistema real puesto que el algoritmo de control utilizará los valores reales en lugar de los predichos.
2. El carrito es un sistema integrador. Por ello, los errores se van acumulando a lo largo del tiempo. Si se presta atención a (5), se puede observar como el ángulo del péndulo depende directamente de las posiciones del carrito. Esto provoca que los errores de predicción del carrito se reflejen también en el péndulo.

A pesar de que estos dos motivos generan mayores errores de predicción, las respuestas real y simulada son muy parecidas en todo el ensayo. Por tanto, se concluye este capítulo con la validación de los modelos ARX que se han obtenido.

5 Diseño del algoritmo de control basado en modelo

Con el fin de lograr una mayor comprensión del algoritmo de control, este capítulo comienza explicando con detalle el enfoque de control utilizado. Para ello se divide la explicación en dos partes: el cálculo de las acciones de control a través de los modelos ARX obtenidos y el bloque conductor que es el encargado de generar trayectorias de referencia.

Una vez comprendida esta idea, se estudian las ecuaciones en diferencias del péndulo invertido sobre un carro para observar como se puede descomponer el sistema en dos subsistemas acoplados y, así, llegar de forma lógica a la estructura de control propuesta.

En la siguiente sección se justifica como el uso de los bloques conductores, propios del enfoque de control empleado, deriva en la necesidad de que el controlador tenga un carácter híbrido. Es decir, que sea capaz de cambiar la estrategia de control en función del estado en el que se encuentre el sistema.

Hecho esto, se evalúa el comportamiento del controlador en simulación y se hace un ajuste más fino de sus parámetros.

Para finalizar, en las dos últimas secciones, se presenta el instrumento virtual que se ha diseñado en LabView para la implementación del algoritmo de control. Y, por último, se muestran los resultados obtenidos en su aplicación sobre el sistema real junto con sus índices de desempeño.

5.1 Enfoque de control

El enfoque de control que se propone en este trabajo está inspirado en [8]. La idea en la que se basa es muy sencilla: utilizar un modelo predictivo ARX del sistema a controlar con el fin de calcular las acciones de control que son necesarias para llevarlo por una trayectoria de referencia deseada (Y_{REF}).

La forma en la que se genera esta trayectoria de referencia y en la que se calculan las acciones de control, se explica a continuación apoyándose en el esquema de la Fig 10.

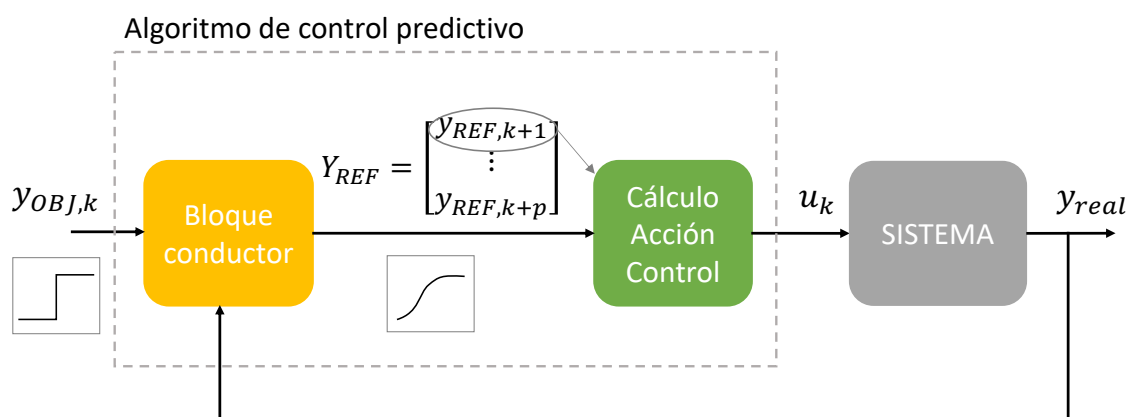


Fig 10. Esquema del enfoque de control.

En primer lugar, si se tiene un modelo ARX con un paso hacia delante de la forma:

$$y_{k+1} = a_1 y_k + a_2 y_{k-1} + \dots + a_{na} y_{k-na} + b_1 u_k + b_2 u_{k-1} + \dots + b_{nb} u_{nb} \quad (17)$$

Es posible calcular la acción de control necesaria en el instante actual (u_k) para llevar el sistema hasta y_{k+1} en el siguiente instante. Para ello, despejando de la ecuación anterior se llega a:

$$u_k = \frac{y_{k+1} - a_1 y_k - a_2 y_{k-1} - \dots - a_{na} y_{k-na} - b_2 u_{k-1} - \dots - b_{nb} u_{nb}}{b_1} \quad (18)$$

Así que, se puede calcular, de una forma sencilla, la acción de control que hay que dar en el instante actual para que la salida alcance un valor deseado en el siguiente instante ($y_{k+1} = y_{REF,k+1}$). Llegados a este punto, conviene tener en cuenta dos consideraciones:

1. La acción de control calculada hará que el sistema alcance el valor exacto de la salida deseada, si y sólo si, el modelo ARX no presenta errores de modelado y, además, no existen perturbaciones ni ruido en el sistema. Nótese la importancia de haber obtenido un modelo que represente fielmente la respuesta del sistema.
2. Es importante que la acción de control calculada esté dentro de los límites físicos del actuador, en caso contrario, no será posible aplicarla sobre el sistema y, por consiguiente, la salida no alcanzará el valor deseado. Aquí, cabe destacar la importancia de que la trayectoria propuesta (Y_{REF}) sea factible desde el punto de vista de los límites de la acción de control. Esto está directamente relacionado con un buen ajuste del bloque conductor del que se hablará más adelante.

En segundo lugar, una vez se conoce cómo calcular las acciones de control para alcanzar una trayectoria deseada, se puede pasar a describir cómo se realiza el cálculo de dicha trayectoria. El encargado de esta tarea es el bloque conductor Fig 10.

Los cambios en el valor objetivo de la salida ($y_{OBJ,k}$) durante el control del sistema suelen ser de tipo escalón, lo que generaría, normalmente, trayectorias que no serían factibles desde el punto de vista de los límites del actuador. Por ejemplo, para hacer que el péndulo pase de $\theta_{eq,1}$ a $\theta_{eq,2}$ en un periodo de muestreo, se requeriría una energía inalcanzable por el motor. Por ese motivo, el objetivo del bloque conductor es generar, a partir de estos escalones en $y_{OBJ,k}$, trayectorias progresivas (Y_{REF}) que sí sean alcanzables por el sistema. Esto se consigue implementando una función de primer o de segundo orden (cuya respuesta ante escalón es bien conocida) y, variando sus parámetros hasta conseguir una trayectoria deseada (ajuste del controlador). Es importante destacar que el bloque conductor debe ser de ganancia unitaria para que el sistema alcance $y_{OBJ,k}$ en régimen permanente.

De forma que, el bloque conductor no es más que una ecuación en diferencias de un modelo de primer (19) o segundo (20) orden con ganancia unitaria y respuesta transitoria ajustable al que es necesario indicarle: el valor objetivo de cada instante ($y_{OBJ,k}$) y la salida/s real/es del sistema (y_k, y_{k-1}). De ahí la necesidad de la realimentación Fig 10.

$$\begin{bmatrix} y_{REF,k+1} \\ \vdots \\ y_{REF,k+p} \end{bmatrix} = \begin{bmatrix} a_1 y_k + b_1 y_{OBJ,k} \\ \vdots \\ a_1 \hat{y}_{k+p-1} + b_1 y_{OBJ,p} \end{bmatrix} \quad (19)$$

$$\begin{bmatrix} y_{REF,k+1} \\ \vdots \\ y_{REF,k+p} \end{bmatrix} = \begin{bmatrix} a_1 y_k + a_2 y_{k-1} + b_1 y_{OBJ,k} + b_2 y_{OBJ,k-1} \\ \vdots \\ a_1 \hat{y}_{k+p-1} + a_2 \hat{y}_{k+p-2} + b_1 y_{OBJ,p} + b_2 y_{OBJ,p-1} \end{bmatrix} \quad (20)$$

5.2 Cálculo de la acción de control: estructura de control propuesta

Si se analizan las ecuaciones dinámicas (5) y (6), se puede ver como, a pesar de que el péndulo invertido sobre un carro es un único sistema, es posible entenderlo como dos subsistemas acoplados entre sí. Ya que, la salida de uno se convierte en la acción de control del otro. Para ello, teniendo presente el enfoque de control explicado en la sección anterior, se presta de nuevo atención a los modelos ARX desplazados un instante hacia delante:

$$\theta_{k+1} = a_{p,1} \theta_k + a_{p,2} \theta_{k-1} + a_{p,3} \sin(\theta_k) - b_{p,1} \cos(\theta_k) a_k \quad (21)$$

$$x_{k+1} = a_{c,1} x_k + a_{c,2} x_{k-1} + b_{c,1} u_k + b_{c,2} u_{k-1} \quad (22)$$

En el caso del péndulo (21), se puede observar como el ángulo en el instante $k+1$ (θ_{k+1}), depende de la posición angular actual (θ_k), de la anterior (θ_{k-1}) y, además, de la aceleración del instante actual (a_k). Es decir, se puede ver a_k como la acción de control de este subsistema. Por tanto, siguiendo el enfoque de control planteado, es posible calcular la aceleración necesaria en el instante actual (acción de control) para alcanzar un ángulo de referencia dado para el siguiente instante ($\theta_{REF,k+1}$) como:

$$a_k = \frac{-\theta_{REF,k+1} + a_{p,1} \theta_k + a_{p,2} \theta_{k-1} + a_{p,3} \sin(\theta_k)}{b_{p,1} \cos(\theta_k)} \quad (23)$$

Recordando que la aceleración en el instante actual es:

$$a_k = x_{k+1} - 2x_k + x_{k-1} \quad (24)$$

Aquí es donde se manifiesta el acoplamiento de los dos subsistemas: ya que, para lograr la aceleración deseada, es necesario que la posición en el siguiente instante (x_{k+1}) sea:

$$x_{k+1} = a_k + 2x_k - x_{k-1}$$

Si se presta atención ahora a el modelo ARX del carrito (22) y se aplica la misma idea que en el caso anterior, se llega a que se puede alcanzar una posición deseada en el siguiente instante (x_{k+1}), si se da una acción de control con el motor tal que:

$u_k = \frac{x_{k+1} - a_{c,1}x_k - a_{c,2}x_{k-1} - b_{c,2}u_{k-1}}{b_{c,1}}$	(25)
--	------

Por tanto, el estudio de los subsistemas por separado con el enfoque de control propuesto sugiere, tal y como indica la Fig 11, el siguiente esquema de control:

1. Dado un ángulo de referencia para el siguiente instante ($\theta_{REF,k+1}$), se calcula la aceleración necesaria para alcanzarlo (acción de control del subsistema péndulo), tal y como indica (23).
2. Siguiendo con la definición de a_k (24), se calcula la posición de referencia para el carrito en el siguiente instante (x_{k+1}).
3. Finalmente, siguiendo (25), se calcula la acción de control del motor (acción de control del subsistema carrito), necesaria para alcanzar la posición del carrito deseada.

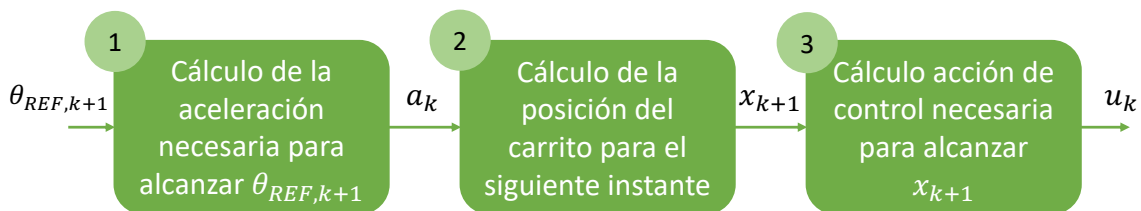


Fig 11. Esquema de control propuesto.

De esta forma se llega a un esquema de control Fig 11 que permite calcular qué acción de control se debe dar al motor eléctrico para que el péndulo alcance un ángulo deseado el siguiente instante.

Siguiendo con el enfoque de control propuesto Fig 10 , ahora quedaría definir el bloque conductor del controlador encargado de generar la trayectoria de referencia de las posiciones angulares del péndulo. Y esto es lo que se aborda en la siguiente sección.

5.3 Bloque conductor: controlador híbrido

Una vez se conoce como calcular la potencia que debe entregar el motor para alcanzar un ángulo deseado en el siguiente instante, es hora de hablar del encargado de generar esas trayectorias deseadas: el bloque conductor. Como ya se mencionó, es importante definir trayectorias que sean factibles con los límites físicos del actuador.

Prestando atención al objetivo del control: estabilizar el péndulo en su punto de equilibrio inestable ($\theta_{eq,2}$) partiendo de su punto de equilibrio estable ($\theta_{eq,1}$). Se puede intuir, conociendo mínimamente el sistema, que las trayectorias que describirá el péndulo hasta alcanzar $\theta_{eq,2}$ serán muy distintas: por un lado, durante el levantamiento del péndulo, las trayectorias serán más rápidas y se requerirán acciones de control más energéticas. En cambio, durante la estabilización, las trayectorias seguirán dinámicas más lentas y acciones de control necesarias serán más suaves. Es decir, el péndulo va a trazar trayectorias con dinámicas muy distintas y el bloque conductor es una función capaz de generar trayectorias con una dinámica determinada.

Por tanto, esto sugiere abordar el problema como un controlador híbrido capaz de cambiar la estrategia de control (trayectoria deseada), según el estado en el que se encuentre el sistema (levantamiento o estabilización). Es importante destacar que la única variación del algoritmo de control entre un estado y otro es el modelo que se utiliza como bloque conductor (que generará trayectorias distintas). La metodología seguida para calcular la acción de control de motor Fig 11 es la misma en ambos.

De esta forma, se dividen todas las posibles posiciones angulares del péndulo en dos zonas o estados Fig 12 que harán que el controlador escoja entre una estrategia u otra dependiendo de en cuál de los dos se encuentre el sistema en ese instante.

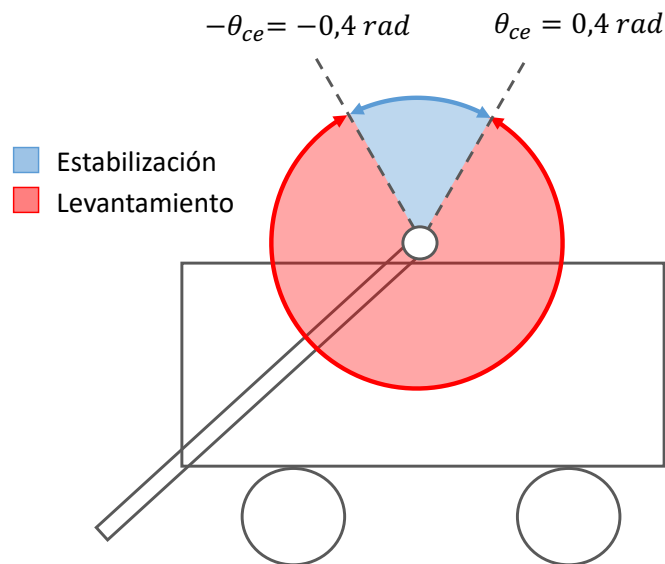


Fig 12. Estados del sistema: levantamiento y estabilización.

Con el fin de evitar una interpretación errónea de lo anterior, se quiere aclarar que el sistema no es de naturaleza híbrida. Si no que, por el enfoque de control que se ha decidido aplicar, se hace necesario, como ya se ha visto, cambiar la estrategia de control según cual sea la posición angular del péndulo. Ya que, con un mismo bloque dirver, no se conseguiría generar trayectorias factibles para todas las posiciones angulares.

5.3.1 Estado 1: Levantamiento

Se entiende que el sistema se encuentra en este estado cuando el péndulo se encuentra en el área coloreada en rojo Fig 12. En este caso, se necesitan acciones de control enérgicas que permitan la subida del péndulo hasta la zona de estabilización. Por ese motivo, el bloque conductor será una función de primer orden (19) con ganancia unitaria y una constante de tiempo pequeña que le permita generar trayectorias rápidas.

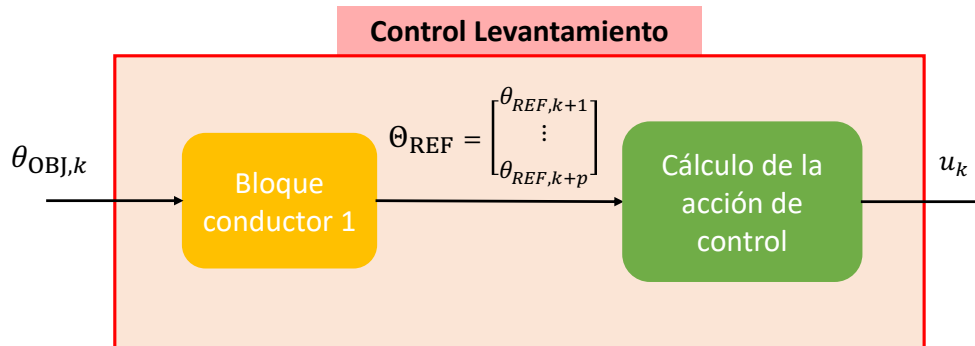


Fig 13. Esquema de control durante el levantamiento.

Además, otro aspecto a tener en cuenta es qué valor objetivo ($\theta_{OBJ,k}$) se le debería dar como entrada a este bloque conductor Fig 13. En este caso, dicho valor objetivo dependerá de la velocidad angular del péndulo en ese instante: si el péndulo lleva una velocidad angular positiva (sentido horario), el valor objetivo será $\theta_{OBJ,k} = 2\pi$. Mientras que, si es negativa (sentido antihorario), deberá ser $\theta_{OBJ,k} = 0$. Se ha de tener en cuenta que el péndulo puede dar vueltas completas y $\theta_{OBJ,k}$ pasará a ser múltiplos de 2π .

5.3.2 Estado 2: Estabilización

En este segundo estado en el que el péndulo se encuentra más cerca de su punto de equilibrio inestable, las trayectorias deben ser más suaves que en el caso anterior y pueden presentar pequeñas sobreoscilaciones. Por ese motivo, se opta por un sistema de segundo orden subamortiguado (20) de ganancia unitaria como bloque conductor.

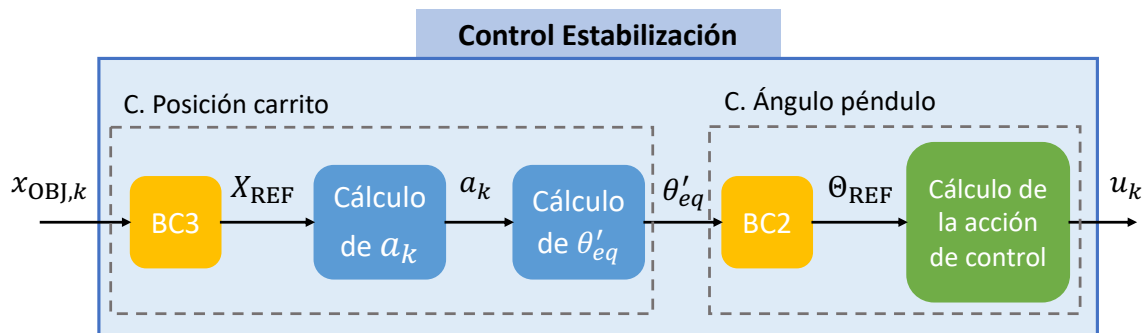


Fig 14. Esquema de control durante la estabilización.

Como en el caso anterior, se debe decidir qué valor objetivo se ha de pasar como entrada al bloque conductor. Lo lógico parecería utilizar los mismos que en el caso anterior puesto que son los puntos en los que se quiere estabilizar el sistema. Pero, en este caso, surge un problema: si se presta atención a la ecuación (23) se puede comprobar que cuando el sistema se encuentra estabilizado en $\theta_{OBJ,k} = 0$ la aceleración necesaria será nula. Pero, una aceleración nula, no implica una velocidad nula. Por tanto, el carrito seguiría con una velocidad constante y acabaría chocando con uno de los extremos del riel Fig 16.

Para evitar este problema, se propone un control en cascada Fig 14, en el que el valor objetivo del ángulo ($\theta_{OBJ,k}$) que se le pasa al controlador de la posición angular del péndulo viene dado por el controlador que se encarga de la posición del carrito.

El funcionamiento de este controlador externo es el siguiente:

1. En primer lugar, recibe el valor objetivo de la posición del carrito ($X_{OBJ,k}$) y calcula una trayectoria de referencia (X_{REF}) mediante un bloque conductor de segundo orden y ganancia unitaria (20).
2. Con el primer valor de dicha trayectoria de referencia ($x_{REF,k+1}$), se calcula la aceleración en el instante actual (a_k) según (24).
3. Por último, se calcula en qué ángulo de equilibrio (θ'_{eq}) puede permanecer el péndulo si el carrito llevase, constantemente, la aceleración a_k calculada. Para ello, partiendo de la ecuación del péndulo (5):

$$\theta'_{eq} = a_{p,1}\theta'_{eq} + a_{p,2}\theta'_{eq} + a_{p,3} \sin(\theta'_{eq}) - b_{p,1} \cos(\theta'_{eq}) a_k \rightarrow$$
$$\theta'_{eq} = \text{actg} \left(\frac{b_{p,1} a_k}{a_{p,3}} \right)$$

4. Finalmente, se escoge este ángulo como valor objetivo para el siguiente controlador ($\theta_{OBJ,k} = \theta'_{eq}$).

De esta forma se consigue estabilizar el péndulo y que el carrito permanezca en la posición deseada ($X_{OBJ,k}$) como se puede ver en la Fig 17. A diferencia de lo que pasaba cuando no se utiliza este controlador en cascada Fig 16.

Los coeficientes de los bloques conductores se escogen analizando las trayectorias que describe el péndulo durante el levantamiento y la estabilización. Y, posteriormente, se hace un ajuste más fino durante la simulación.

5.4 Simulación y ajuste de parámetros

Con el fin de evitar daños sobre el sistema, verificar el correcto funcionamiento del controlador y realizar un ajuste más fino de los parámetros del mismo, se lleva a cabo el control del modelo (obtenido en el capítulo 4), haciendo uso de un diagrama de bloques que se ha creado en Simulink Fig 15.

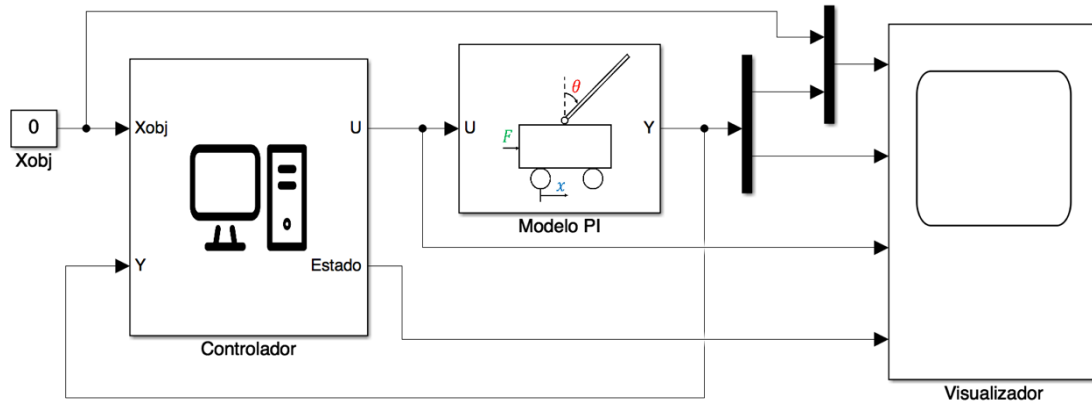


Fig 15. Diagrama de bloques en Simulink para la simulación del controlador.

Los elementos que componen el diagrama son:

1. Modelo del sistema: que contiene las ecuaciones en diferencias del sistema con los coeficientes que se identificaron (15)(16).
2. Bloque controlador: ejecuta el algoritmo de control descrito en las secciones anteriores.
3. Valor objetivo del carrito: bloque encargado de pasarle al controlador la referencia en la que se quiere estabilizar el carrito ($X_{OBJ,k}$).
4. Visualizador: bloque que permite visualizar los resultados de la simulación.

El reajuste de los parámetros de los bloques conductores se lleva a cabo de forma heurística tras varias simulaciones llegando a los siguientes sistemas:

Bloque conductor 1: para el levantamiento del péndulo

$$\tau = 0,15 \text{ (s)}$$

$\theta_{REF,k+1} = 0,7165\theta_k + 0,2835\theta_{OBJ,k}$	(26)
--	------

Bloque conductor 2: para la estabilización del péndulo

$$t_p = 0,3 \text{ (s)} \quad \delta = 2 \text{ (\%)}$$

$\theta_{REF,k+1} = 0,9024\theta_k - 0,2714\theta_{k-1} + 0,2244\theta_{OBJ,k} + 1446\theta_{OBJ,k-1}$	(27)
--	------

Bloque conductor 3: para el control de posición del carrito

$$t_p = 2 \text{ (s)} \quad \delta = 6 \text{ (\%)}$$

$X_{REF,k+1} = 1,8584x_k - 0,8688x_{k-1} + 0,0053x_{OBJ,k} + 0,0051x_{OBJ,k-1}$	(28)
---	------

Tras la simulación del diagrama de bloques se obtienen los siguientes resultados:

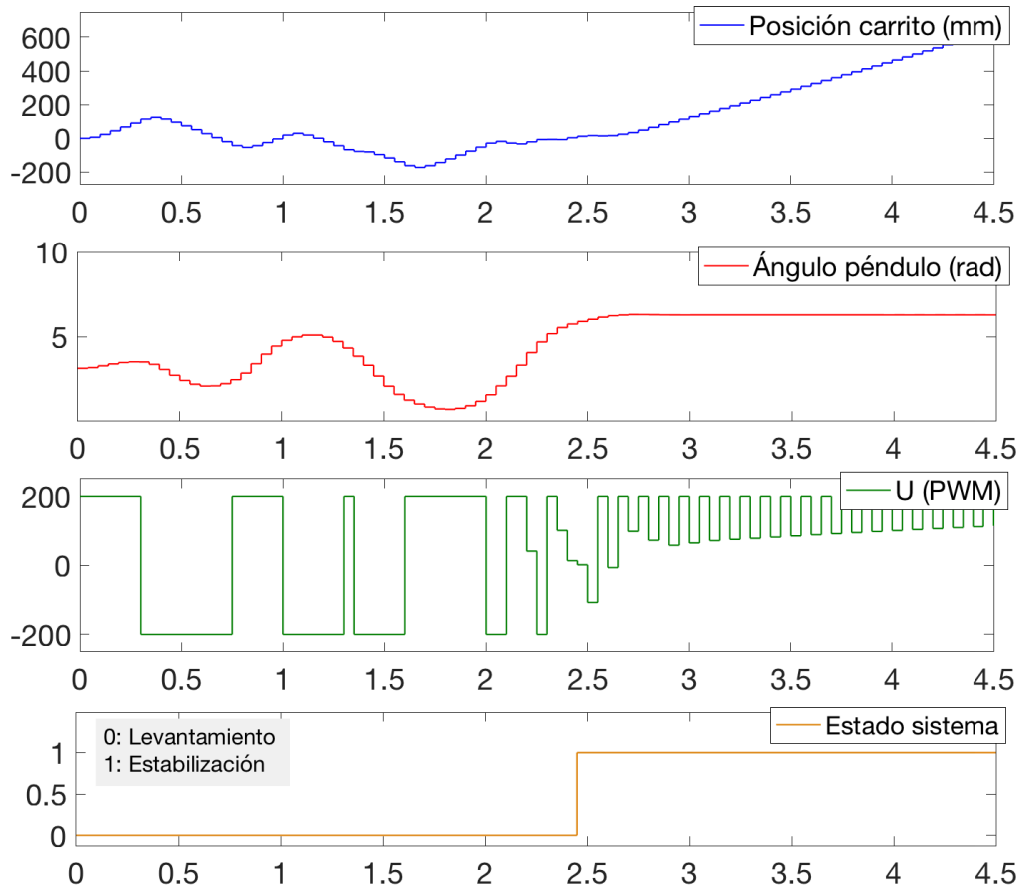


Fig 16. Simulación del control del sistema sin control cascada.

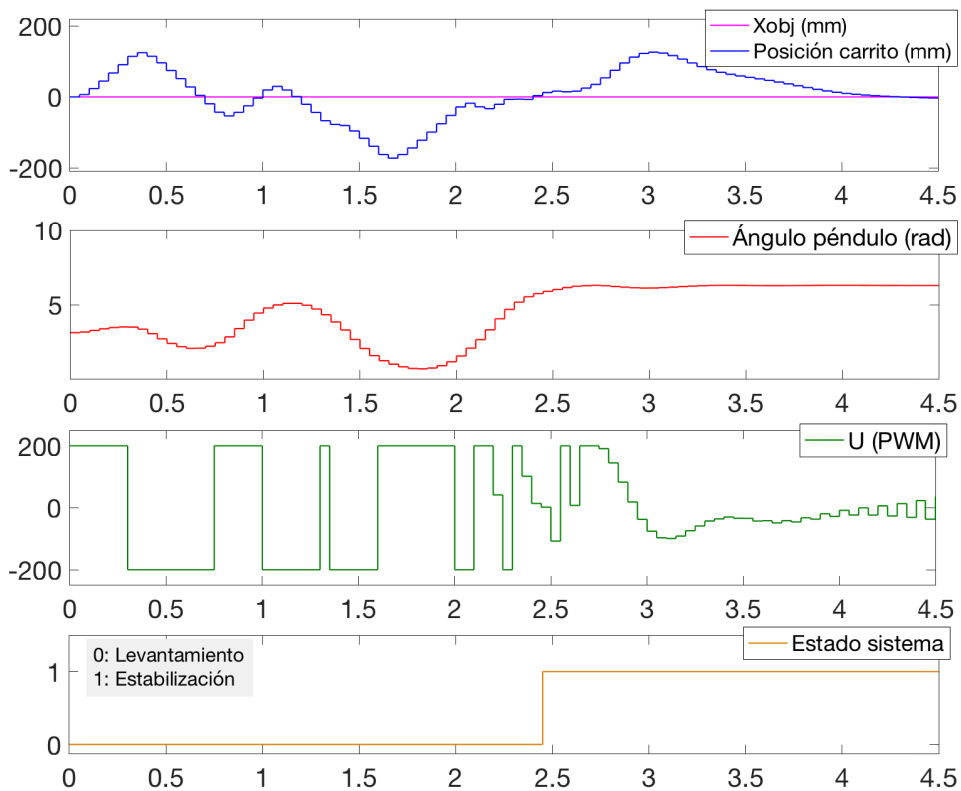


Fig 17. Simulación control del sistema con control cascada.

Como se puede ver, en la Fig 16, no se ha implementado el control en cascada para que se pueda observar como, aunque el péndulo se haya estabilizado y, por tanto, la aceleración sea nula, el carrito se sigue moviendo con velocidad constante. Lo que haría que chocase con los extremos del riel en el sistema real como ya se comentó. Sin embargo, cuando se incorpora el control cascada Fig 17, este problema queda resuelto quedando el carrito en la posición deseada.

A la vista de la Fig 17, se puede concluir que los resultados del controlador son satisfactorios y se logra el objetivo de control. Por tanto, se puede pasar a su implementación sobre el sistema real.

5.5 Implementación del controlador

Se ha diseñado un instrumento virtual en LabView que permita tanto el control como el monitoreo de las variables del sistema entre otras funciones. La Fig 18 muestra el aspecto de la interfaz del programa.

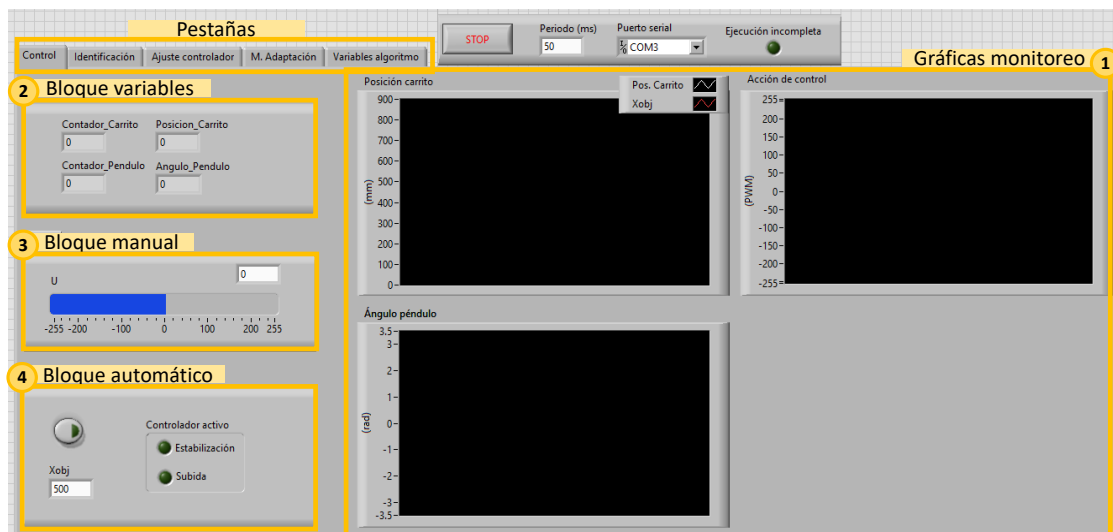


Fig 18. Pestaña de monitoreo y control.

Como se puede ver en la figura, la aplicación se está organizada en pestañas para dotarla de mayor claridad y facilitar su uso. Estas son:

1. Control Fig 18: Dedicada al monitoreo y control manual o automático del sistema. Los elementos que la componen se detallan más adelante.
2. Identificación: En ella se pueden generar señales PRBS destinadas a los ensayos de identificación del sistema que se llevaron a cabo en la sección 4.4.
3. Ajuste del controlador: en esta ventana se pueden configurar los parámetros del controlador que se han visto. Es decir, el ángulo de cambio de estado entre levantamiento y estabilización (θ_{ce}) y los parámetros de los bloques conductores (τ, t_p, δ).
4. Mecanismo adaptación Fig 20: contiene los parámetros de ajuste y los resultados del mecanismo de adaptación que se ha diseñado. Sus elementos se detallarán en la sección 6.1.

5. Variables algoritmo: recoge una serie de variables propias del funcionamiento del algoritmo de control como las trayectorias de referencia o vectores en los que se almacenan los valores anteriores de las salidas.

Se detallan, a continuación, los elementos que componen la pestaña de control y su funcionalidad:

1. Graficas de monitoreo: muestran las variables del sistema: posición del carrito (x) y valor objetivo (X_{OBJ}), ángulo del péndulo (θ) y acción de control (U).
2. Bloque variables: muestra el valor de los contadores de los encoders y de las variables del carrito y del péndulo.
3. Bloque manual: dispone de una barra que se puede mover para dar acciones de control de forma manual.
4. Bloque automático: cuenta con un botón para activar y desactivar el controlador, el valor objetivo del carrito y dos indicadores luminosos que indican en cuál de los dos estados se encuentra el sistema en ese instante.

5.6 Validación y desempeño del controlador

Una vez se ha presentado la aplicación que se utiliza, se finaliza este capítulo con la validación del algoritmo de control sobre el sistema real.

En primer lugar, se posiciona el carrito de forma manual en el centro del riel. Y, seguidamente, se activa el modo de control automático. Además, durante en el ensayo se ha ido cambiando el valor de X_{OBJ} para observar si el controlador es capaz de estabilizar al péndulo en su punto de equilibrio inestable y, a su vez, que el carrito permanezca en la posición deseada. Los resultados obtenidos se muestran en la siguiente figura:

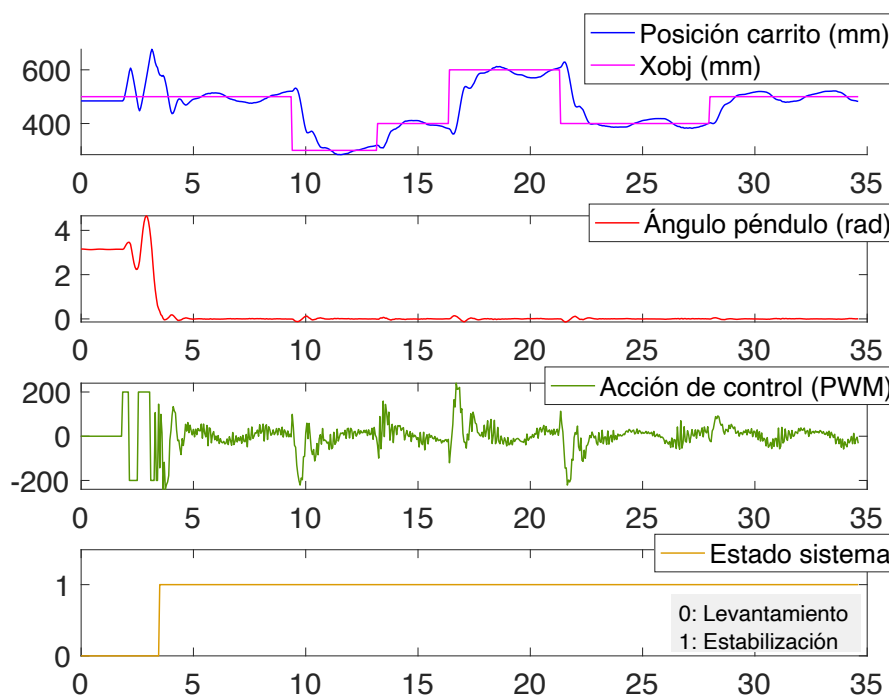


Fig 19. Validación del controlador sobre el sistema real.

A la vista del ensayo, se puede afirmar que el controlador es capaz de cumplir con el objetivo de control: el levantamiento del péndulo partiendo desde $\theta_{eq,1}$ y su estabilización en $\theta_{eq,2}$. Además, gracias al control en cascada que se usa durante el estado de estabilización, el carrito se posiciona en el valor deseado (X_{OBJ}). Aunque, es cierto que siempre oscila al rededor de este valor sin llegar a quedarse completamente estable en él. Esto se debe, principalmente, a la fricción del carrito sobre el riel.

Además, la gráfica del “Estado del sistema” indica qué estrategia de control (levantamiento o estabilización) está activa en cada instante del ensayo: un cero corresponde al estado de Levantamiento. Mientras que, el uno, indica que el sistema se encuentra en el estado de Estabilización.

Se escogen como indicadores para medir el desempeño del controlador el tiempo de establecimiento ($t_{e,98\%}$) y la sobreoscilación (δ) de la posición angular del péndulo y la desviación angular entendida como la desviación angular máxima del péndulo cuando se hace un cambio en el valor objetivo del carrito (X_{OBJ}). Además, se distingue entre dos fases a la hora de calcular estos indicadores: el levantamiento del péndulo que va desde que el péndulo se encuentra en $\theta_{eq,1}$ hasta que se consigue estabilizarlo en $\theta_{eq,2}$ por primera vez. Y la estabilización donde se van dando cambios en X_{OBJ} cuando el péndulo ya se encuentra estable en $\theta_{eq,2}$.

Los valores para este ensayo son los siguientes:

LEVANTAMIENTO		
$t_{e,98\%}$	2,8	(s)
δ	2,21	(%)

Tabla 1. Indicadores de desempeño durante el levantamiento para el control con el ordenador.

ESTABILIZACIÓN		
$t_{e,98\%}$	1,1	(s)
desviación angular	0,1307	(rad)

Tabla 2. Indicadores de desempeño durante la estabilización para el control con el ordenador.

6 Ampliaciones

En este capítulo se van a comentar las dos ampliaciones que se han realizado tras conseguir el objetivo principal de este trabajo. La primera de ellas ha sido dotar al algoritmo de control de la capacidad para detectar cambios en los parámetros del sistema, pudiendo hablar así de un mecanismo de adaptación.

Tras esto, motivado por las tendencias que siguen muchos de los sistemas de control actuales, se decidió, además, embeber el algoritmo de control en un microcontrolador. Para ello, se utilizó un DSPIC33FJ256GP710A de Microchip al que se le instalaron las librerías de un sistema operativo de tiempo real (FreeRTOS).

6.1 Controlador adaptivo

El algoritmo de control diseñado utiliza modelos ARX para el cálculo de la acción de control en cada instante. Estos modelos, se obtienen de una identificación previa del sistema y, su capacidad para representar fielmente el comportamiento del sistema, como ya se ha argumentado, es crucial para un buen funcionamiento del algoritmo de control. De ahí, la importancia que tiene una correcta identificación de los coeficientes del modelo.

Sin embargo, los sistemas reales están sujetos a variaciones en su comportamiento, o lo que es lo mismo: variaciones en los parámetros del sistema Fig 4. Estos cambios pueden estar provocados por: el desgaste de ciertos componentes, modificaciones de los elementos que componen el sistema u otros fenómenos. Esto, a su vez, implica un cambio en los coeficientes ($a_{p,i}$ $b_{p,i}$ $a_{m,i}$ $b_{m,i}$) del modelo y, por consiguiente, deriva en un mal comportamiento del controlador.

Para el péndulo invertido sobre un carro, las variaciones de sus parámetros por desgaste podrían ser mínimas. Sin embargo, modificaciones en el sistema como: el acople de una masa puntual al péndulo, el uso de varillas distintas, cambios en la masa del carrito... provocan variaciones significativas en los coeficientes de los modelos obtenidos.

Así que, en esta sección se explica la técnica de adaptación que se ha utilizado para detectar estos cambios en el sistema con el fin de poder actualizar los coeficientes de los modelos ARX que utiliza el controlador.

6.1.1 Mecanismo de adaptación: Mínimos Cuadrados Recursivos

Como se ha comentado, los cambios en el comportamiento del sistema (que pueden estar provocados por múltiples causas), se traducen en un cambio de los coeficientes del modelo ARX. La identificación de estos coeficientes se llevó a cabo por medio del algoritmo de Mínimos Cuadrados en el capítulo 4.

Como ya se explicó, este algoritmo utiliza un conjunto de datos entrada-salida para calcular el valor que minimiza la suma de los errores al cuadrado (J). De forma que no es posible ejecutar el algoritmo en línea. Sin embargo, en este caso, se va a emplear la técnica de mínimos cuadrados recursivos cuya diferencia con respecto a la anterior es

que el algoritmo se puede ejecutar con cada paquete nuevo de información que se reciba de las variables del sistema. Por tanto, se puede ejecutar en línea para hacer el cálculo de los coeficientes del modelo con toda la información disponible hasta ese momento.

Cabe destacar que el algoritmo que se utiliza en ambos casos es el mismo, pero, con ayuda de herramientas matemáticas y la reformulación del problema de Mínimos Cuadrados, se consigue llegar al algoritmo recursivo. Esto se traduce en que la identificación se hace de la misma forma: minimizando el error cuadrático en todos los instantes. Pero, en este caso, no es necesario disponer de un conjunto de datos de un ensayo, sino que se puede hacer el cálculo online.

Los pasos que sigue el algoritmo de detallan a continuación:

Con los nuevos valores de y_k y u_k se genera el vector ψ_k de la forma:

$$\psi_k = [y_{k-1} \quad y_{k-2} \quad \dots \quad y_{k-na} \quad u_{k-1} \quad u_{k-2} \quad \dots \quad u_{k-nb}]$$

Se calcula c como:

$$c = \frac{1}{a} + \psi_k^T \frac{Q_{k-1}}{\gamma} \psi_k$$

Se calcula el factor de ponderación del error σ_k :

$$\sigma_k = \frac{1}{\gamma c} Q_{k-1} \psi_k$$

Los nuevos coeficientes del modelo θ_k son:

$$\theta_k = \theta_{k-1} + \sigma_k (y_k - \psi_k^T \theta_{k-1})$$

Y finalmente, se calcula la matriz de covarianzas para la siguiente iteración:

$$Q_{k-1} = (1 - \sigma_k \psi_k^T) \frac{Q_{k-1}}{\gamma}$$

Donde:

ψ_k : vector de regresión en el instante k .

a : peso máximo de ponderación del error. En este caso $a = 1$.

γ : factor de olvido. En este caso $\gamma = 0,99$.

Q_i : matriz de covarianzas en el instante i .

θ_i : coeficientes del modelo en el instante i .

6.1.2 Implementación y validación del mecanismo de adaptación

Una vez conocido el algoritmo, basta con implementar los pasos que se deben seguir en el instrumento virtual que se ha diseñado para el control del sistema (sección 5.5).

La pestaña del programa que se en la que se pueden configurar los parámetros del mecanismo de identificación y observar sus resultados es la siguiente:

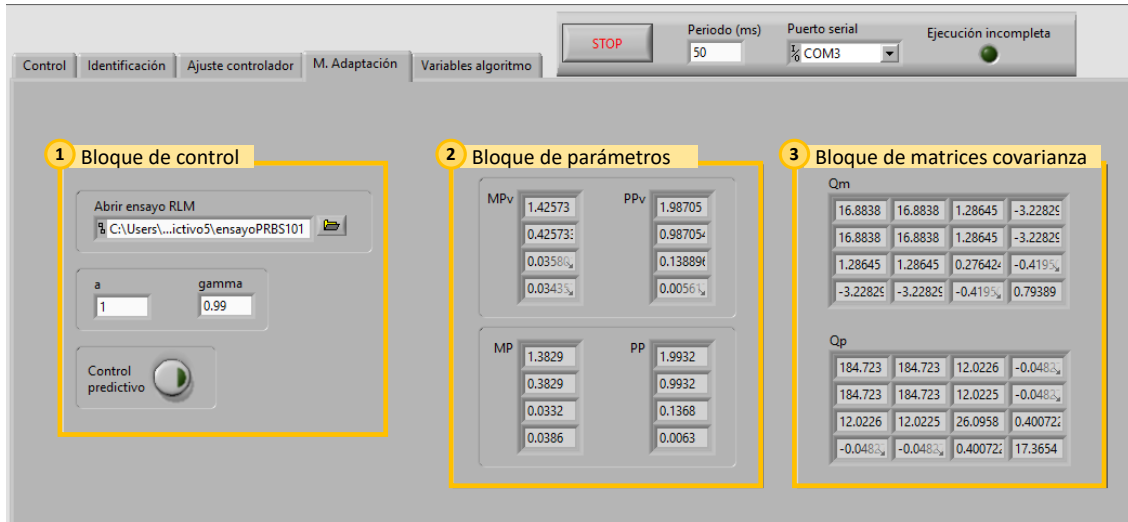


Fig 20. Pestaña del mecanismo de adaptación en el instrumento virtual.

En ella aparecen:

1. Bloque de control, que contiene: la dirección del archivo con los valores de entrada-salida para calcular los coeficientes iniciales de los modelos (15)(16) por el método de mínimos cuadrados tal y como se hizo en el capítulo 4; el peso máximo de ponderación de los errores (α) y el factor de olvido (γ); y, un pulsador que actualiza los coeficientes de los modelos que utiliza el controlador.
2. Bloque de coeficientes: en el aparecen los valores iniciales de los modelos (MP y PP) calculados por mínimos cuadrados a partir del archivo mencionado. Y también los coeficientes que se obtienen en cada interacción al aplicar la técnica de RLMS (MPv PPv).
3. Bloque de matrices de covarianza: contiene las matrices de covarianza propias del método utilizado.

Finalmente, se realiza un experimento para comprobar la capacidad de adaptación del mecanismo implementado. Los pasos seguidos se enumeran a continuación:

1. Se añade una masa puntual al péndulo: de ahora en adelante “Sistema real modificado” Fig 21. Así, se están cambiando parámetros del sistema como: la masa del péndulo y su momento de inercia. Lo que, a su vez, implica una variación de los coeficientes del modelo inicial (15) y (16).
2. Se comienza a mover el carrito mientras se va ejecutando el algoritmo de mínimos cuadrados recursivos. De esta forma, se añade información de entrada-salida para la identificación de los nuevos coeficientes del modelo. El ensayo finaliza una vez el mecanismo de identificación se estabiliza en los nuevos valores de PPv y MPv. Así, se llega al nuevo modelo del sistema modificado (29) y (30).

3. Tras esto, con el fin de validar el modelo obtenido en el punto anterior, se realiza otro ensayo sobre el sistema real modificado recogiendo todos los valores de las salidas y las acciones de control en un fichero.
4. Finalmente, se comparan las salidas reales del sistema (experimento anterior), con las del nuevo modelo obtenido en el punto 2. Esta comparación se lleva a cabo con el diagrama de bloques que se utilizó durante la validación de los modelos Fig 8.

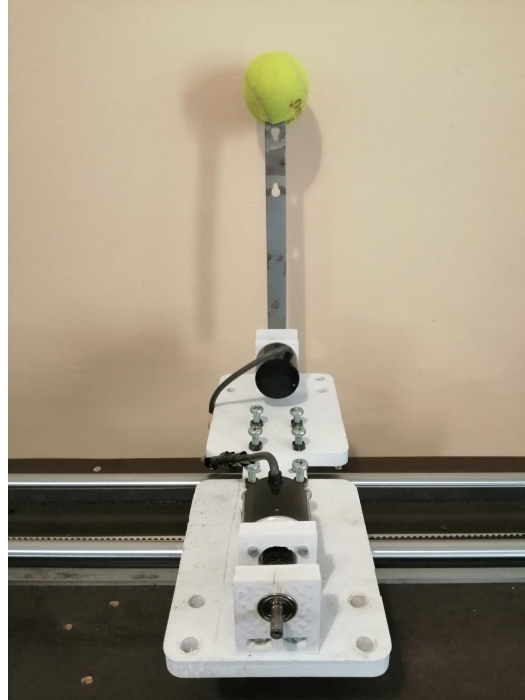


Fig 21. Fotografía del sistema modificado.

El modelo obtenido mediante el mecanismo de adaptación para el sistema modificado es:

$$\theta_k = 1,997\theta_{k-1} - 0,997\theta_{k-2} + 0,111 \sin(\theta_{k-1}) - 0,048 \cos(\theta_{k-1}) a_k \quad (29)$$

$$x_k = 1,397x_{k-1} + 0,397x_{k-2} + 0,025u_{k-1} + 0,045u_{k-2} \quad (30)$$

Los resultados obtenidos durante la validación del nuevo modelo son los siguientes:

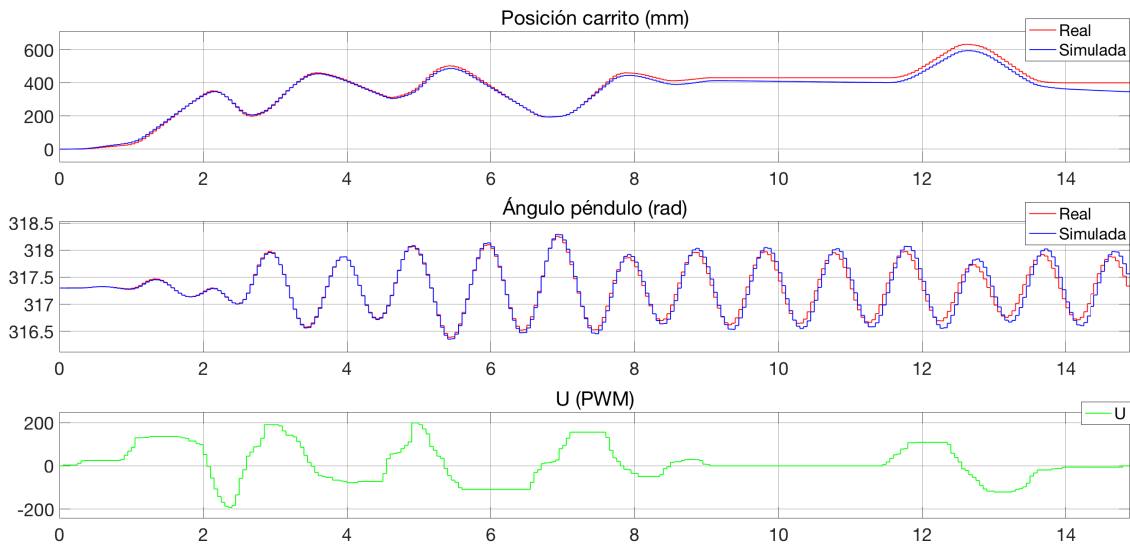


Fig 22. Validación de los coeficientes obtenidos para el sistema modificado.

Se puede apreciar como la respuesta del modelo se ajusta con exactitud a la del sistema real modificado. Como ocurría en la sección 4.4, las diferencias entre ambas se van acentuando conforme avanza el experimento por los motivos que se comentaron: el carrito es un sistema integrador que acumula errores y el modelo utiliza los valores predichos anteriores y no los del sistema real.

6.1.3 Control del sistema modificado

Una vez se tiene la certeza de que el mecanismo de adaptación es capaz de detectar los cambios en el comportamiento del sistema y calcular los nuevos coeficientes del modelo, se decide llevar a cabo el control del sistema modificado para observar si esta capacidad de adaptación incrementa la robustez del algoritmo de control frente a los cambios en los parámetros del sistema.

Para ello, se realizarán dos experimentos:

1. Control del sistema modificado sin el uso del mecanismo de adaptación: los coeficientes de los modelos que utiliza el controlador son los iniciales sin actualizarlos (15) y (16).
2. Control del sistema modificado con el uso del mecanismo de adaptación: en este caso si que se actualizan los coeficientes de los modelos utilizando (29) y (30).

Los resultados de los experimentos se pueden observar en las siguientes figuras:

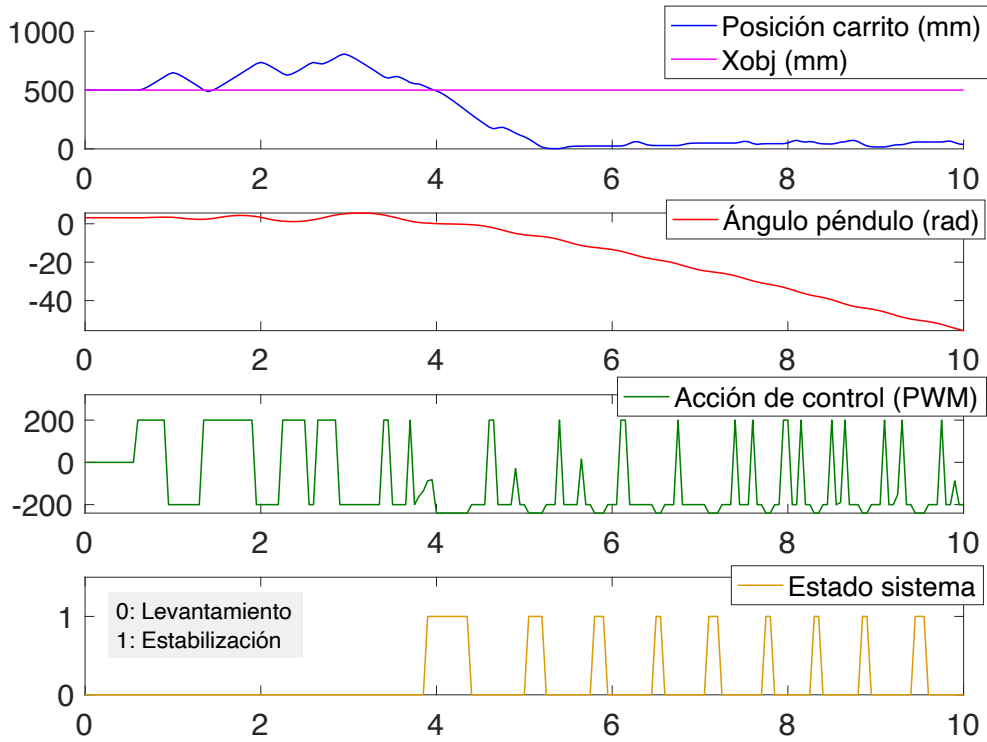


Fig 23. Control sistema modificado sin mecanismo de adaptación.

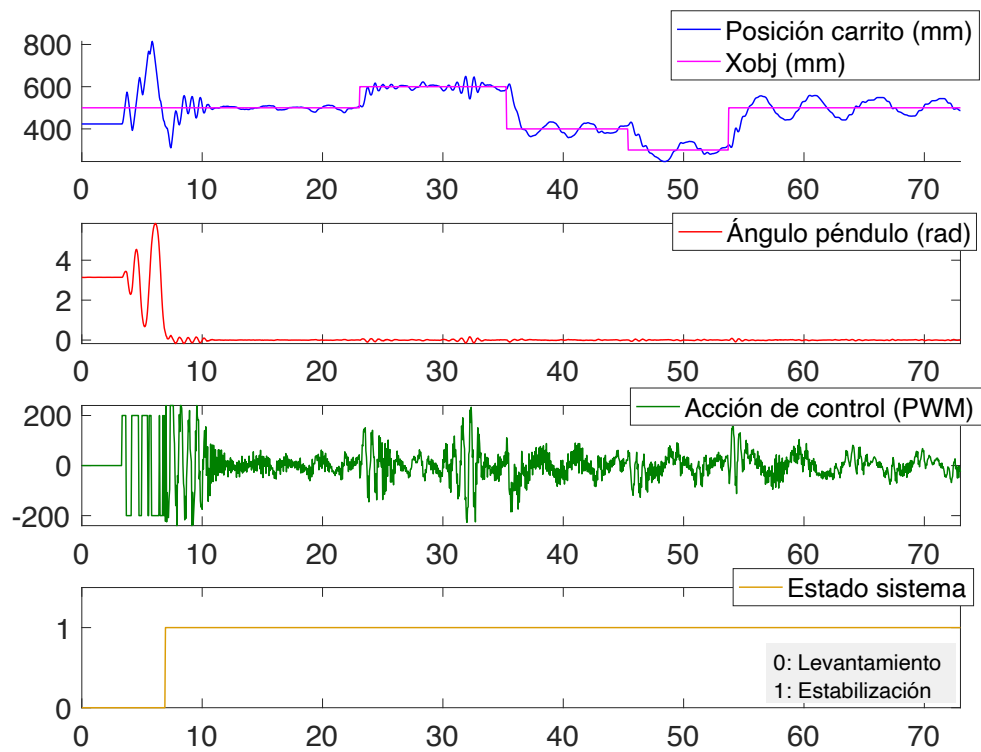


Fig 24. Control sistema modificado con mecanismo de adaptación.

Se puede observar como el controlador no es capaz de estabilizar el sistema modificado cuando no se utiliza el mecanismo de adaptación Fig 23. Sin embargo, cuando se hace uso de este mecanismo Fig 24, se puede comprobar como se logra el objetivo de control. Así, se pone de manifiesto la importancia que tiene una correcta identificación del sistema en el correcto funcionamiento del controlador, como ya se había mencionado.

Por tanto, a la vista de los resultados, se puede afirmar que el mecanismo de adaptación incrementa la robustez del controlador ante los cambios en el comportamiento del sistema.

Los valores de los indicadores de desempeño del controlador para este caso son los siguientes:

LEVANTAMIENTO		
$t_{e,98\%}$	7,5	(s)
δ	5,61	(%)

Tabla 3. Indicadores de desempeño durante el levantamiento para el control con el ordenador del sistema modificado.

ESTABILIZACIÓN		
$t_{e,98\%}$	1,95	(s)
<i>desviación ang</i>	0,1044	(rad)

Tabla 4. Indicadores de desempeño durante la estabilización para el control con el ordenador del sistema modificado.

Se debe prestar atención a como empeora el comportamiento del controlador sobre el sistema modificado con respecto al no modificado a pesar de que se consigan actualizar los coeficientes del modelo. La razón de esto, como se explicará en el siguiente punto, se debe a que al no reajustar los bloques conductores para el nuevo sistema puede que las trayectorias propuestas no sean factibles.

6.1.4 Limitaciones

A pesar de que los coeficientes del modelo se actualicen correctamente aparece un nuevo problema que deriva de la filosofía de control propuesta: si la variación de los parámetros es muy brusca puede que las trayectorias que generan los bloques conductores ajustados para el sistema inicial (26) (27) (28) no sean factibles para el nuevo comportamiento del sistema.

Por ejemplo, si se aumenta demasiado el peso del péndulo, el bloque conductor de levantamiento puede hacer que suba con demasiada inercia y que la trayectoria que propone el bloque conductor de estabilización no sea factible porque las acciones de control necesarias no son alcanzables por el motor. Por tanto, no se puede controlar el sistema.

Así que, sería necesario que los bloques conductores también tuvieran esa capacidad de adaptación y propusieran trayectorias factibles para el sistema en función de sus

parámetros. De esta forma, se lograría una mejora sustancial puesto que el ajuste de los bloques conductores sería automático logrando una mayor versatilidad del controlador e incrementando aún más su robustez frente a estos cambios. Este tema no se ha podido abordar durante el trabajo, pero se propone como una futura ampliación.

6.2 Control embebido en un microcontrolador

La presencia de los sistemas embebidos en el entorno industrial es cada vez mayor por razones como: su bajo coste en comparación con otros dispositivos como los PLC, capacidad para funcionar como sistemas de tiempo real, facilidad en la conectividad con otros equipos o su diseño modular, entre otras [9]. Por ese motivo, embeber el algoritmo de control en un microcontrolador con un sistema operativo en tiempo real, supone poder controlar el sistema con un equipo más económico, menos voluminoso y que garantiza la periodicidad de la tarea de control.

Para ello, se ha utilizado un microcontrolador DSPIC33FJ256GP710A de Microchip junto a la plataforma de desarrollo Explorer 16/32 que contiene elementos hardware necesarios para el desarrollo de las aplicaciones Fig 25. Además, se ha hecho uso de las librerías de FreeRTOS: un sistema operativo en tiempo real que permite ejecutar tareas en el microcontrolador garantizando una periodicidad deseada (sistema de control en tiempo real). Con este equipo se ha logrado sustituir al ordenador que ejecuta la aplicación de LabVIEW diseñada para el control del péndulo invertido sobre un carro y al Arduino que funciona como TAD.

En esta sección se comienza explicando brevemente el equipo que se ha empleado y las funciones de los elementos hardware de la plataforma Explorer 16/32 que se utilizan. A continuación, se detalla como se ha hecho uso de la técnica de hardware-in-the-loop para el desarrollo y la testado del algoritmo de control sobre sistema embebido. Y, para finalizar, se pasa al control del sistema y se muestran los resultados obtenidos.

6.2.1 Equipo utilizado

Como ya se ha dicho, el sistema embebido que se ha usado para sustituir al equipo encargado del control y a la TAD está compuesto por un microcontrolador DSPIC33FJ256GP710A y una plataforma de desarrollo Explorer 16/32. A continuación, se muestra una imagen del equipo y se detallan las funciones de los componentes utilizados:

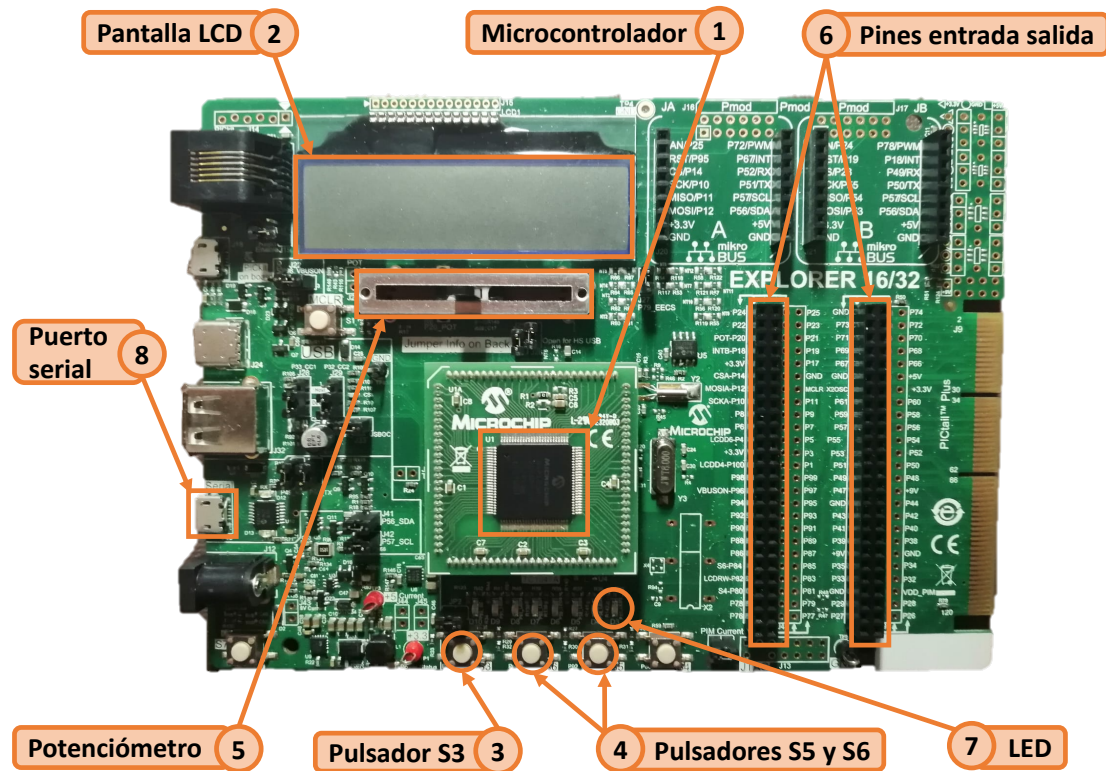


Fig 25. Fotografía del sistema embebido utilizado.

1. Microcontrolador: ejecuta las tareas de control (manual o automático) que se han programado, lee la información de los encoders del péndulo y el carrito y escribe sobre la tarjeta PWM para dar las acciones de control al motor. Además, ejecuta un sistema operativo (FreeRTOS) que lo convierte en un sistema de control en tiempo real.
2. Pantalla LCD: se utiliza para mostrar el valor de las variables de salida (x y θ) en cada instante.
3. Pulsador S3: permite el cambio entre modo manual y automático. Cada vez que se presiona cambia el modo en el que se encuentra en ese instante.
4. Pulsadores S5 y S6: sirven para, en modo automático, aumentar o disminuir el valor de X_{OBJ} , respectivamente.
5. Potenciómetro: se utiliza en modo manual para dar las acciones de control. Tiene la misma función que la barra de control manual de la Fig 18.
6. Pines entrada y salida: permiten la conexión de los encoders para leer sus pulsos mediante interrupciones y la escritura de las acciones de control sobre la tarjeta PWM.
7. LED: indicador luminoso que se activa cuando el modo automático está activo.

8. Puerto serie: necesario para la programación del microcontrolador y la comunicación con el ordenador en el control del Hardware-in-the-loop.

6.2.2 Simulación: Hardware-in-the-loop

El algoritmo de control es el mismo que se simuló en el diagrama de bloques en Simulink y que ejecuta la aplicación de LabView para el control del sistema real. Pero, en este caso, para asegurar un buen funcionamiento del microcontrolador antes de ponerlo en marcha sobre el sistema real, se controla un sistema simulado que ejecuta las ecuaciones en diferencias del sistema real (15) (16). Esta técnica se conoce como hardware-in-the-loop.

Para ello, se crea en Matlab un programa encargado de comunicarse con el microcontrolador a través del puerto serial y simular el comportamiento del sistema real. De forma que, el programa recibe la acción de control del microcontrolador, ejecuta las ecuaciones en diferencias y le devuelve al micro las salidas del sistema como si fueran las leídas por los sensores reales del sistema. El siguiente esquema muestra los pasos que se siguen en la técnica de Hardware-in-the-loop:

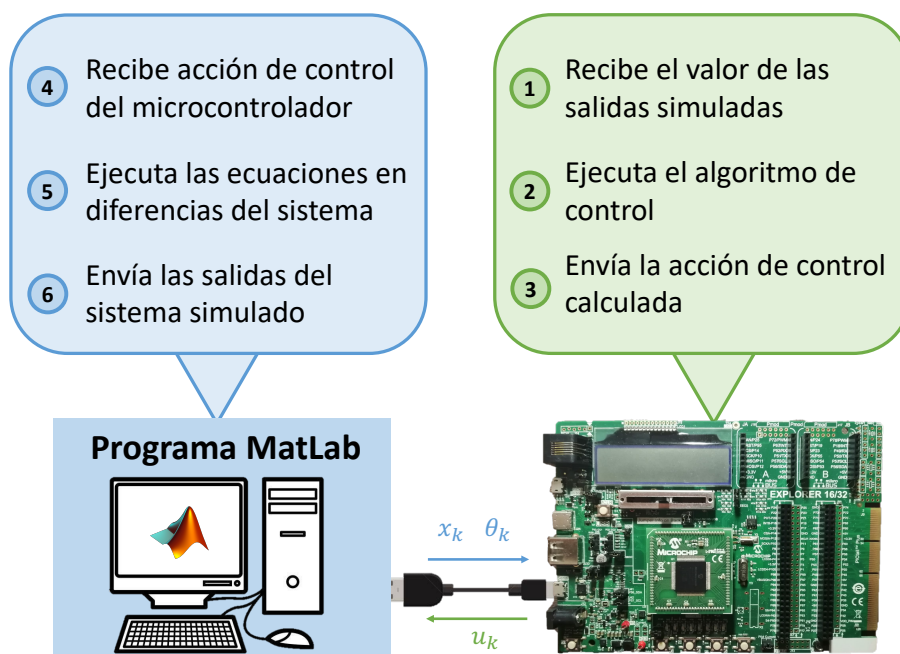


Fig 26. Esquema Hardware-in-the-loop.

La respuesta que se obtiene del control del modelo virtual mediante el microcontrolador es la siguiente:

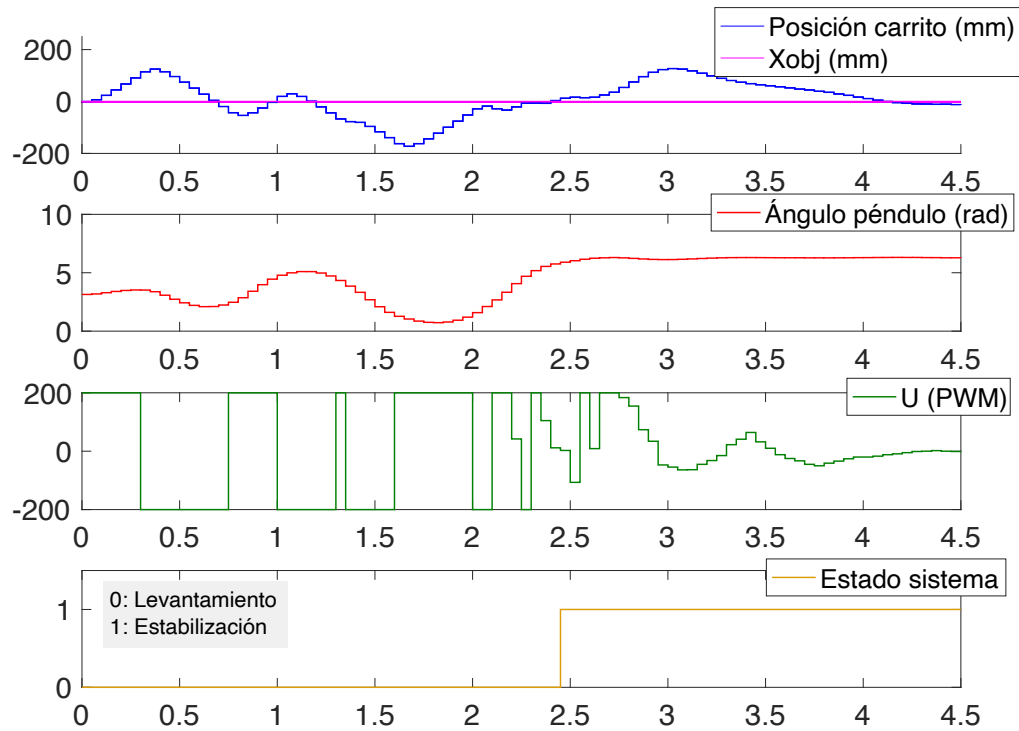


Fig 27. Control del sistema simulado mediante el microcontrolador (Hardware-in-the-loop).

Como se puede observar, la respuesta es muy similar a la que se obtuvo durante la simulación del algoritmo en la Fig 17. Por tanto, se puede validar el correcto funcionamiento del micro y pasar así al control del sistema real.

6.2.3 Control del sistema real

Una vez se tiene la certeza de que el algoritmo de control funciona correctamente se puede pasar al control del sistema real. Ahora, será el microcontrolador el que desempeñe: por un lado, el papel del ordenador, decidiendo en cada instante el valor de la acción de control según se encuentre en modo manual o automático. Y, por otro, la función del Arduino como TAD leyendo los sensores y actuando sobre el motor.

El algoritmo de control ya se ha testado en el punto anterior. Así que, para actuar sobre el sistema real, el primer paso es conseguir que el microcontrolador también pueda realizar la función de TAD. Para ello, se establecen las conexiones entre los sensores y el motor con el microcontrolador en sus pines de entrada y salida. Y, hecho el conexionado, se programa el DSPIC con las funciones de lectura de los sensores y escritura sobre la tarjeta PWM.

Una vez se tiene todo listo, se pasa al control del sistema real para validar el control. La única diferencia con respecto al punto anterior es que ahora el microcontrolador lee directamente los valores de los sensores.

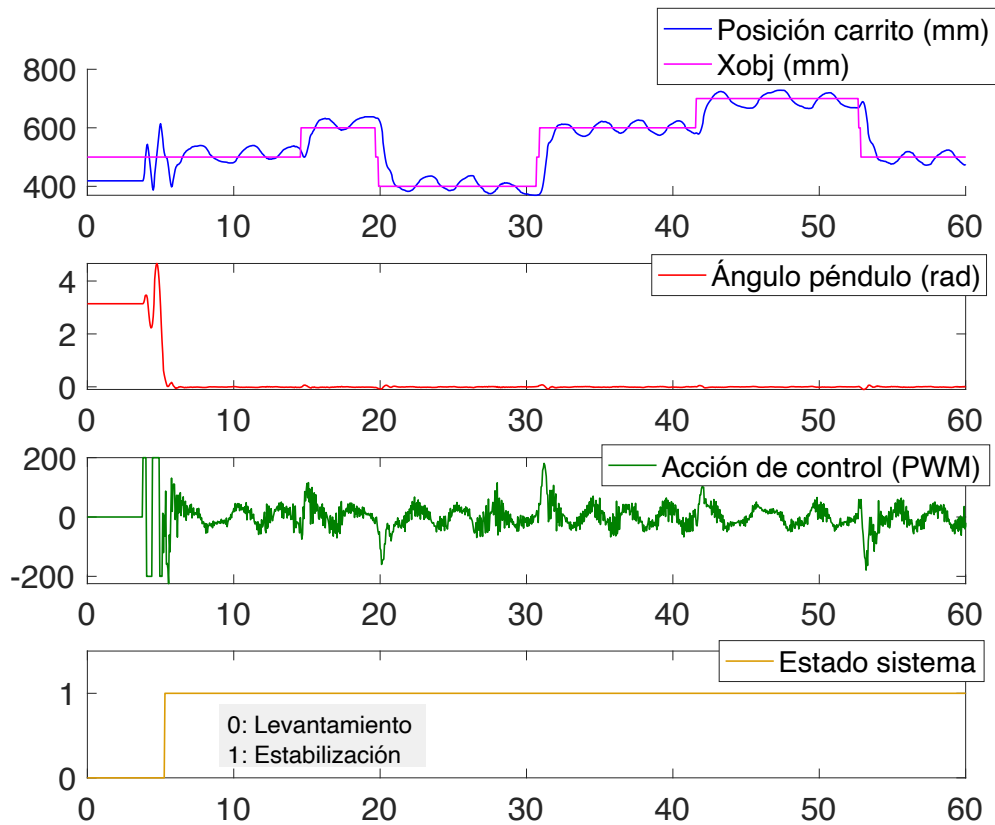


Fig 28. Validación del sistema embebido.

LEVANTAMIENTO		
$t_{e,98\%}$	2,4	(s)
δ	2,02	(%)

Tabla 5. Indicadores de desempeño durante el levantamiento para el control con el sistema embebido.

ESTABILIZACIÓN		
$t_{e,98\%}$	1	(s)
desviación ang	0,0824	(rad)

Tabla 6. Indicadores de desempeño durante la estabilización para el control con el sistema embebido.

A la vista de las figuras se concluye que se alcanza el objetivo de control satisfactoriamente. Además, si se comparan estos resultados con los del control del sistema a través del ordenador Fig 19, se comprueba como, a pesar de estar ejecutando el mismo algoritmo de control, el desempeño que se obtiene con el sistema embebido es mayor. De esta forma, se pone de manifiesto como los sistemas en tiempo real, al ser capaces de cumplir de manera estricta con la periodicidad con la que se controla el sistema (T_s), logran un mayor desempeño que un ordenador de uso convencional.

7 Conclusiones y futuras oportunidades de investigación

Para poner punto final a este documento, se enumeran, a continuación, las conclusiones que se extraen del trabajo realizado:

En primer lugar, se ha puesto de manifiesto como con un enfoque de control tan sencillo, que calcula las acciones de control basándose en modelos ARX, se es capaz de lidiar con la dinámica compleja de un péndulo invertido sobre un carro.

También, cabe destacar que el correcto sintonizado de los bloques conductores de esta filosofía de control es crucial para lograr el objetivo que se persigue. Aunque, es cierto que esta tarea se vuelve más crítica en este caso por tratarse de un sistema inestable.

Además, el mecanismo de adaptación propuesto es un complemento que dota de mayor robustez al algoritmo de control por el hecho de poder detectar y calcular, de manera eficaz, a las variaciones en el comportamiento de los sistemas. Lo que permite actualizar los coeficientes de los modelos que utiliza el controlador.

De los dos puntos anteriores surge la necesidad de que los bloques conductores que utiliza este algoritmo también tengan esa capacidad de adaptación para garantizar la factibilidad de las trayectorias que generan desde el punto de vista del actuador. Esto, dotaría al algoritmo de una mayor robustez y versatilidad. Por ello, se propone como una futura ampliación de este proyecto.

Por último, se ha observado como, a pesar de ejecutar el mismo algoritmo de control, el desempeño que se obtiene con el sistema embebido es mayor que el de un ordenador de uso convencional. Esto, sumado a su diferencia de precios, justifica el uso de estos dispositivos a nivel industrial.

8 Bibliografía

- [1] O. Boubaker, "The inverted Pendulum: A fundamental Benchmark in Control Theory and Robotics", 2012 International Conference on Education and e-Learning Innovations, 2012.
- [2] T. Glü'ck, A. Eder, and A. Kugi, "Swing-up control of a triple pendulum on a cart with experimental validation", *Automatica*, vol. 49, no. 3, pp. 801–808, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.automata.2012.12.006>
- [3] M. R. Nalavade, M. J. Bhagat, and V. V. Patil, "Balancing Double Inverted Pendulum on A cart by Linearization Technique", *Journal of Recent Technology and Engineering*, vol. 3, no. 1, pp. 153–157, 2014.
- [4] M. A. Sahnehsaraei, M. Mahmoodabadi, and A. Bagheri, "Pareto optimum control of a 2-DOF inverted pendulum using approximate feedback linearization and sliding mode control", *Transactions of the Institute of Measurement and Control*, vol. 36, no. 4, pp. 496–505, 2013.
- [5] C. W. Tao, J. Taur, J. H. Chang, and S. F. Su, "Adaptive fuzzy switched swing-up and sliding control for the double-pendulum-and-cart system", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 1, pp. 241–252, 2010.
- [6] G. S. Miljković, I. S. Stojković, D. B. Denić, "Generation and application of pseudorandom binary sequences using virtual instrumentation", *Automatic Control and Robotics* Vol. 10, No 1, pp. 51 - 58, 2011.
- [7] P. Diniz, "The Least-Mean-Square (LMS) Algorithm", 2020.
- [8] **JUAN MANUEL**
- [9] J. F. Ready, "VRTX: A Real-Time Operating System for embedded Microprocessor Applications," *IEEE Micro*. 6(4), 1986.