



UCA

Universidad
de Cádiz

Escuela Superior
de Ingeniería

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

**REALIZACIÓN, ANÁLISIS Y DISEÑO DE UN
SISTEMA DE CONTROL PARA UN PÉNDULO
INVERTIDO EXPERIMENTAL**

AUTOR: JOSÉ BARROSO VARO

Cádiz, noviembre de 2022



Universidad
de Cádiz

Escuela Superior
de Ingeniería

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

**REALIZACIÓN, ANÁLISIS Y DISEÑO DE UN
SISTEMA DE CONTROL PARA UN PÉNDULO
INVERTIDO EXPERIMENTAL**

DIRECTOR: MANUEL JESÚS SÁNCHEZ LÓPEZ

AUTOR: JOSÉ BARROSO VARO

Cádiz, noviembre de 2022

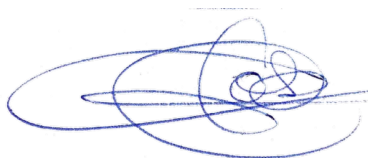
DECLARACIÓN PERSONAL DEL AUTOR

José Barroso Varo con DNI 44051808T, estudiante del Título de Grado en Ingeniería Electrónica Industrial para el que se presenta el TFG en la Escuela Superior de Ingeniería de la Universidad de Cádiz, como autor de este documento académico titulado “Realización, análisis y diseño de un sistema de control para un péndulo invertido experimental” y presentado como Trabajo Final de Grado en Ingeniería Electrónica Industrial.

DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara y precisa su origen tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse.

En Puerto Real, a 20/11/2022

A handwritten signature in blue ink, consisting of several overlapping loops and curves, positioned centrally below the date.

Fdo: José Barroso Varo

Agradecimientos

Me gustaría transmitir mis más sinceros agradecimientos a todas aquellas personas que me han ayudado a lo largo de esta etapa y han colaborado en el desarrollo de este proyecto. En primer lugar, a mi tutor, Manuel Jesús López Sánchez, no solo de TFG sino también de alumno colaborador, por la paciencia depositada en mí y haberme ayudado siempre que lo necesité. Pero principalmente me gustaría darle mis agradecimientos por todos los conocimientos que compartió conmigo más allá de las clases oficiales.

Mis agradecimientos también a cada uno de los componentes del grupo de investigación GAPSIS, ya que desde un principio me hicieron sentirme como uno más de la familia. Pero especialmente a Agustín Consegiere Castilla por la confianza que depositó en mí y porque cada vez que necesité un consejo o una ayuda siempre pude contar con él en el momento que fuese.

A mi familia... Mi hermana, ya que fue desde un principio mi motivación para cursar unos estudios universitarios. A mis padres, por el esfuerzo realizado para que yo pudiese finalizar este viaje y, por supuesto, por los valores y educación que me han dado y de la que me siento muy orgulloso. A mi pareja, por las veces que se ha tenido que leer el presente trabajo, el apoyo y ánimos mostrados en los últimos años de carrera y la paciencia que ha tenido en las últimas semanas de corrección. A Mika, mi perrita, porque de alguna forma u otra su amor incondicional se ve reflejado en este trabajo.

Finalmente, me gustaría darle mis agradecimientos a todas aquellas personas, profesores y maestros que han aportado su granito de arena en mi enseñanza y que gracias también a ellos hoy me encuentro a punto de cruzar la línea de meta de esta preciosa carrera.

Resumen

En el presente proyecto se va a realizar el análisis y estudio del modelo matemático de un péndulo invertido con un motor de corriente continua como actuador. Posteriormente, se diseñará un sistema de control mediante un método analítico para la planta en cuestión. Finalmente, se dimensionará y construirá la planta y se implementará un control experimental que consiga estabilizar al sistema en lazo cerrado y tenga unas prestaciones satisfactorias desde el punto de vista de la respuesta temporal. Los resultados serán comparados con los obtenidos en la simulación con el control analítico.

Tras un análisis del diagrama de cuerpo libre del sistema péndulo invertido se obtendrán las ecuaciones que definen la dinámica del conjunto. Estas ecuaciones serán linealizadas para el punto de operación en el cual se desea estabilizar la planta. Por simplicidad, el análisis del actuador se realizará por separado. Ambos sistemas estarán relacionados por el torque, o el par aplicado por el motor, y el radio de la rueda dentada que acople el motor con la correa del carro del péndulo invertido simple.

Para poder realizar las simulaciones del modelo obtenido y analizar los resultados, será necesario obtener los parámetros característicos de la planta. Para ello, se va a diseñar la planta en un software de diseño 3D y posteriormente se imprimirán las piezas estructurales mediante tecnología FDM¹. Una vez se tenga la planta construida se podrá caracterizar y obtener los parámetros para realizar las simulaciones y su posterior control.

Para estabilizar la planta en simulación se diseñará un sistema de control mediante una técnica analítica. Esta técnica consistirá en realizar el control a partir de la realimentación del vector de estados, en concreto un control tipo LQR². En el diseño de dicho controlador se supondrá que se tiene un modelo de la planta preciso y que todas las variables de estados son medibles por los correspondientes sensores.

Aunque se haya realizado un análisis del sistema y se hayan obtenido las ecuaciones de la dinámica del sistema con los parámetros de la planta real, esto solo es una aproximación para el diseño de un controlador y que dará buenos resultados en simulación. La planta tendrá incertidumbres no modeladas y perturbaciones no consideradas en el diseño, las cuales podrán llevar al sistema a un comportamiento no deseado o incluso a la inestabilidad. Teniendo en cuenta las perturbaciones e incertidumbres de modelado del sistema, es posible que el control LQR no sea válido en la planta real. Es por ello que el diseño del sistema de control de la planta real se realizará de forma empírica o experimentalmente con un proceso iterativo variando las constantes de un control tipo PID aplicando los conocimientos adquiridos para un ajuste manual.

El objetivo final del trabajo será obtener una planta péndulo invertido apoyado sobre un carro que se desplace sobre unos rieles y actuado por un motor de corriente continua. El algoritmo de control será desarrollado en el software de procesamiento

¹FDM - Fused Deposition Modeling

²LQR - Linear-Quadratic Regulator

matemático Matlab/Simulink. Dicho algoritmo de control será implementado posteriormente en una Raspberry Pi 4B [15]. Este dispositivo cuenta con una serie de entradas y salidas digitales que junto a su procesador *ARM Cortex-A72* hacen posible procesar las medidas de los sensores, ejecutar de forma autónoma el algoritmo de control en tiempo real y generar la señal de control necesaria para estabilizar el sistema.

Abstract

This project will analyse and study the mathematical model of an inverted pendulum with a DC motor as an actuator. Subsequently, a control system will be designed using an analytical method for the plant in question. Finally, the plant will be dimensioned and constructed and an experimental control system will be implemented to stabilise the system in a closed loop and to achieve satisfactory performance from the point of view of the time response. The results will be compared with those obtained in the simulation with the analytical control.

After an analysis of the free body diagram of the inverted pendulum system, the equations defining the dynamics of the assembly will be obtained. These equations will be linearised for the operating point at which the plant is to be stabilised. For simplicity, the actuator analysis will be carried out separately. Both systems will be related by the torque, or the torque applied by the motor, and the radius of the sprocket that couples the motor to the belt of the simple inverted pendulum trolley.

In order to carry out the simulations of the model obtained and analyse the results, it will be necessary to obtain the characteristic parameters of the plant. To do this, the plant will be designed in 3D design software and then the structural parts will be printed using FDM³ technology. Once the plant has been built, it will be possible to characterise it and obtain the parameters to carry out the simulations and subsequent control.

To stabilise the plant in simulation, a control system will be designed using an analytical technique. This technique will consist of carrying out the control from the feedback of the state vector, specifically an LQR⁴ type control. In the design of such a controller, it will be assumed that there is an accurate model of the plant and that all the state variables are measurable by the corresponding sensors.

Although an analysis of the system has been carried out and the equations of the system dynamics have been obtained with the parameters of the real plant, this is only an approximation for the design of a controller and will give good results in simulation. The plant will have unmodelled uncertainties and disturbances not considered in the design, which may lead the system to undesired behaviour or even instability. Considering the system modelling disturbances and uncertainties, the LQR control may not be valid in the real plant. That is why the design of the control system of the real plant will be carried out empirically or experimentally with an iterative process by varying the constants of a PID type control applying the knowledge acquired for a manual tuning.

The final objective of the work will be to obtain an inverted pendulum plant supported on a trolley moving on rails and driven by a DC motor. The control algorithm will be developed in the mathematical processing software Matlab/Simulink. This control algorithm will then be implemented on a Raspberry Pi 4B [15]. This device has

³FDM - Fused Deposition Modeling

⁴LQR - Linear-Quadratic Regulator

a series of digital inputs and outputs which, together with its processor *ARM Cortex-A72*, make it possible to process the sensor measurements, autonomously execute the control algorithm in real time and generate the control signal necessary to stabilise the system.

Índice general

1. Introducción general	1
1.1. Introducción	1
1.2. Motivación	1
1.3. Objetivos	2
1.4. Alcance	2
2. Estudio previo	3
2.1. El péndulo invertido	3
2.2. Variantes del péndulo invertido	4
2.2.1. Péndulo invertido simple	4
2.2.2. Péndulo invertido doble	4
2.2.3. Péndulo invertido triple	5
2.2.4. Péndulo invertido lineal o rotacional	5
2.3. Aplicaciones	6
2.3.1. Estabilización de cohetes para un aterrizaje controlado	7
3. Modelo matemático de la planta	8
3.1. Introducción	8
3.2. Modelo matemático del péndulo invertido	8
3.2.1. Linealización del modelo alrededor del punto de operación	11
3.2.2. Representación en el espacio de estados	13
3.3. Modelo matemático del actuador	14
3.4. Espacio de estados ampliado con el actuador	18
4. Dimensionamiento mecánico y electrónico de la planta	21
4.1. Introducción	21
4.2. Sistema mecánico de la planta	21
4.3. Sistema de procesamiento digital	24
4.4. Selección de los sensores	26
4.4.1. Sensor de posición angular del péndulo	27
4.4.2. Sensor de posición lineal del carro	31
5. Simulación y análisis de la planta	33
5.1. Introducción	33
5.2. Análisis del sistema en lazo abierto	33
5.3. Análisis del sistema en lazo cerrado	36
6. Diseño de un sistema de control mediante técnica analítica basada en el método LQR	38
6.1. Introducción	38
6.2. Análisis de la planta con actuador en lazo abierto	40
6.3. Diseño del sistema de control LQR	41

6.4. Simulaciones y análisis del sistema de control LQR obtenido	45
7. Configuración de la Raspberry Pi	52
7.1. Introducción	52
7.2. Descripción de los bloques de Simulink dedicados a Raspberry Pi y utilizados en este proyecto	57
7.2.1. GPIO Read	57
7.2.2. Encoder	58
7.2.3. PWM	58
7.2.4. SPI Master Transfer	59
7.2.5. UDP Receive y UDP Send	62
7.3. Configuración del modelo para ejecutarlo en la Raspberry Pi	64
7.4. Comandos útiles de Matlab para trabajar con la Raspberry Pi	65
8. Construcción del sistema real	66
8.1. Introducción	66
8.2. Dimensionamiento del motor de corriente continua	66
8.3. Montaje estructural	68
8.4. Montaje del sistema electrónico	71
8.5. Calibración de los sensores	73
8.5.1. Sensor de posición lineal	73
8.5.2. Sensor de posición angular	76
9. Pruebas experimentales con el sistema real	80
9.1. Introducción	80
9.2. Diagrama de control utilizado en los ensayos experimentales	80
9.3. Controlador utilizado en la fase experimental	82
9.4. Resultados experimentales	83
9.4.1. Resultados del ensayo N ^o 1 (Controlador tipo P)	83
9.4.2. Resultados del ensayo N ^o 2 (Controlador tipo PI)	85
9.4.3. Resultados del ensayo N ^o 3 (Controlador tipo PID)	87
9.5. Comparación de resultados	90
10. Conclusiones	93
11. Bibliografía	96
Anexos	98

Índice de figuras

2.1. Péndulo invertido simple	3
2.2. Péndulo invertido doble	4
2.3. Péndulo invertido triple	5
2.4. Péndulo invertido lineal y rotacional	6
2.5. Péndulo invertido lineal y rotacional	6
2.6. Cohete Starship instantes antes del aterrizaje	7
3.1. Diagrama de cuerpo libre del péndulo invertido simple.	9
3.2. Diagrama de cuerpo libre dividido en dos partes	10
3.3. Modelo eléctrico-mecánico de un motor de corriente continua	15
3.4. Diagrama de bloques de un motor CC	16
4.1. Piezas laterales de la planta	22
4.2. Diseño del carro	22
4.3. Rodamiento lineal	23
4.4. Ensamblaje del diseño mecánico de la planta	23
4.5. Raspberry Pi 4B	25
4.6. <i>PinOut</i> Raspberry Pi 4B	26
4.7. Ejemplos de codificadores incrementales. [6]	27
4.8. Determinación del sentido de giro en un codificador incremental mediante dos opto-sensores desfasados angularmente. [6]	28
4.9. Corte de un codificador absoluto basado en el corte de los haces de luz entre emisores LED y fototransistores. [6]	29
4.10. Sensor de movimiento inercial MPU6050	29
4.11. Sensor de posición de efecto hall del fabricante <i>TT Electronics</i>	30
4.12. Codificador incremental relativo KY-040	31
5.1. Mapa de polos y ceros del sistema en lazo abierto 5.3	34
5.2. Respuesta del sistema en lazo abierto ante una entrada tipo impulso	35
5.3. Mapa de polos y ceros del sistema en lazo cerrado sin controlador	37
6.1. Diagrama de bloques de un sistema de control por realimentación del vector de estados	38
6.2. Respuesta de la planta con actuador ante una entrada tipo Delta de Dirac	40
6.3. Diagrama de bloques de control por realimentación del vector de estados	41
6.4. Mapa de polos y ceros del sistema de control en lazo cerrado para el controlador LQR.	44
6.5. Diagrama de simulación del sistema péndulo-motor linealizado con control LQR.	45
6.6. Diagrama de simulación de la planta linealizada en el espacio de estados.	46
6.7. Respuesta de la posición angular del péndulo y la posición lineal del carro ante una perturbación de 5 voltios en el segundo 1 con una duración de 0.01 segundos.	46

6.8.	Esfuerzo de control sin efecto de saturación y con la saturación.	47
6.9.	Diagrama de bloques del modelo no lineal del sistema péndulo-carro. . .	48
6.10.	Diagrama de bloques del modelo no lineal del sistema péndulo-carro con actuador.	48
6.11.	Diagrama de simulación creado para comparar la respuesta del modelo lineal con la del modelo no lineal.	49
6.12.	Respuesta del sistema de control con el modelo no lineal y modelo linealizado ante un impulso de 5 voltios en la señal de control.	49
6.13.	Respuesta del sistema de control con el modelo no lineal y modelo linealizado ante una señal de duración determinada que hace al sistema no lineal perder la estabilidad.	50
7.1.	Administrador de complementos de Matlab.	53
7.2.	Consola de comandos de la Raspberry Pi.	54
7.3.	Configuración de inicio de la Raspberry Pi.	55
7.4.	Configuración de los protocolos de comunicación de la Raspberry Pi. . .	56
7.5.	Configuración de parámetros del bloque GPIO Write.	57
7.6.	Configuración de parámetros del bloque <i>Encoder</i>	58
7.7.	Configuración de parámetros del bloque <i>PWM</i>	59
7.8.	Codificación de las direcciones del ADC <i>MCP3008</i>	60
7.9.	Comunicación SPI con el ADC <i>MCP3008</i> . Modo 0	61
7.10.	Diagrama de simulación para testear el bloque <i>SPI Master Transfer</i> . . .	61
7.11.	Bloques utilizados en la Raspberry Pi para la comunicación por protocolo UDP	62
7.12.	Bloques utilizados en el Host para la comunicación por protocolo UDP . . .	63
8.1.	Resultados de revoluciones por minuto y par motor para una simulación de 5 segundos con una perturbación en el segundo 1	67
8.2.	Motor de corriente continua del fabricante <i>Seeed Studio</i> de 12 voltios y 1000 rpm (Modelo: JGA25-370)	68
8.3.	Imagen del sistema real construido	69
8.4.	Imagen del sistema real construido con el péndulo en su posición vertical . . .	69
8.5.	Ubicación del motor y sensor de desplazamiento lineal	70
8.6.	Ubicación del sensor de posición angular	70
8.7.	Conexión entre el ADC <i>MCP3201</i> y la Raspberry Pi 4B	71
8.8.	Controlador de motores <i>L298N</i>	72
8.9.	Conexión entre el controlador de motor <i>L298N</i> y la Raspberry Pi 4B . .	73
8.10.	Medida inicial de la distancia entre el carro y una pieza estructural fija. . .	74
8.11.	Respuesta obtenida del sensor de desplazamiento lineal ante un desplazamiento de 100 milímetros.	75
8.12.	Respuesta obtenida del sensor de desplazamiento lineal ante un desplazamiento de 100 milímetros con el factor de corrección.	75
8.13.	Diagrama de simulación utilizado para realizar el ajuste del sensor de desplazamiento lineal.	76
8.14.	Medidas tomadas por el sensor de posición angular en un periodo de 10 segundos.	77
8.15.	Medidas tomadas por el sensor de desplazamiento angular sin calibrar con el péndulo inclinado a ± 45 grados.	78

8.16. Medidas tomadas por el sensor de desplazamiento angular calibrado con el péndulo inclinado a ± 45 grados.	79
9.1. Diagrama de control utilizado en los ensayos experimentales.	80
9.2. Estructura paralela de un controlador tipo PID.	82
9.3. Respuesta del sistema de control tipo P ante una perturbación de duración 0.05 segundos a los 10 segundos de simulación	84
9.4. Señal de control obtenida de la respuesta de la Figura 9.3	84
9.5. Respuesta del sistema de control tipo P ante una perturbación de duración 0.1 segundos a los 10 segundos de simulación	85
9.6. Respuesta del sistema de control tipo PI ante una perturbación de duración 0.1 segundos a los 10 segundos de simulación	86
9.7. Señal de control obtenida en la respuesta de la Figura 9.6	87
9.8. Respuesta del sistema de control tipo PID ante una perturbación de duración 0.1 segundos a los 10 segundos de simulación	88
9.9. Señal de control obtenida en la respuesta de la Figura 9.8	89
9.10. Respuesta del sistema de control tipo PID ante una perturbación de duración 0.15 segundos a los 10 segundos de simulación	90
9.11. Comparación de los sistemas de control tipo PI y PID para una perturbación de 0.1 segundos a los 10 segundos de ensayo con una duración total de 40 segundos	91
9.12. Comparación de los sistemas de control tipo PI y PID para una perturbación de 0.1 segundos cada 20 segundos de ensayo con una duración total de 120 segundos	92

Índice de tablas

3.1. Simbología y breve descripción de las variables y constantes del sistema péndulo invertido carro.	9
3.2. Simbología y breve descripción de las constantes y variables de un motor de corriente continua	15
4.1. Constantes del sistema péndulo-carro.	24
5.1. Resultados del análisis en lazo abierto del péndulo invertido.	35
5.2. Resultados del análisis en lazo cerrado del péndulo invertido.	36
6.1. Constantes motor de corriente continua comercial.	40
6.2. Resultados del análisis en lazo cerrado de la planta con actuador.	41
6.3. Resultados del análisis en lazo cerrado de la planta con actuador.	44
7.1. Comparación de las medidas de tensión tomadas por un voltímetro y por el ADC	62
9.1. Constantes ajustadas para el controlador tipo P	83
9.2. Constantes ajustadas para el controlador tipo PI	86
9.3. Constantes ajustadas para el controlador tipo PID	87

Capítulo 1

Introducción general

1.1. Introducción

En el presente trabajo de fin de grado se va a realizar todo el procedimiento para la construcción y control de un péndulo invertido sobre un carro con desplazamiento lineal, el cual se encuentra accionado por un motor de corriente continua.

El péndulo invertido es un sistema no lineal e inestable en su punto de equilibrio. En la actualidad, existe una gran variedad de sistemas que cuentan de alguna forma con un control de estabilidad similar al de un péndulo invertido, tanto en la vida cotidiana como a nivel industrial. Es por todo ello que se ha elegido este sistema como prototipo para la investigación y desarrollo de un sistema de control implementado a una planta real.

Durante el proceso de desarrollo del presente trabajo se realizará un estudio del modelo matemático de la planta, el diseño y simulación de un sistema de control mediante un método analítico y, para finalizar, la construcción de la planta y su control experimental.

Se pretende que tanto las simulaciones como la planta real sirvan como punto de partida para futuros trabajos de investigación de técnicas de control en sistemas inestables.

1.2. Motivación

En la ingeniería de control no siempre es posible disponer de un modelo matemático preciso de la planta mediante el cual poder realizar el diseño del sistema de control, ya que la caracterización de la misma lleva a un proceso largo y costoso.

Existen sistemas que debido al uso para el cual están destinados pueden suponer un peligro para la sociedad si su correcto funcionamiento no ha sido verificado con precisión. Es el caso, por ejemplo, de la industria aeroespacial en la cual existe una alta dedicación a la obtención de un modelo muy preciso del sistema. A partir de dicho modelo se pueden realizar las simulaciones necesarias que validen el correcto funcionamiento sin necesidad de llevar a cabo pruebas con el sistema real que resulten altamente costosas.

Sin embargo, no siempre se disponen de los recursos o los intereses necesarios para realizar dicho proceso de caracterización de la planta. En ese caso es necesario acudir a técnicas experimentales para llevar a cabo un control sobre el propio sistema.

Con la elaboración de este trabajo se pretende exponer este mismo problema en un sistema no lineal e inestable al no disponer de un modelo preciso de la planta. Además de realizar el procedimiento experimental para poder obtener el control de la planta real. Por último, se pretenden ampliar los conocimientos en materia de ingeniería de control conociendo de primera mano lo que se suele encontrar a nivel industrial.

1.3. Objetivos

Los objetivos del presente trabajo son la construcción y control de un péndulo invertido. Este sistema consiste en un péndulo acoplado a un carro, el cual se encuentra accionado por un motor de corriente continua y que se desplaza linealmente por una barra con longitud dada.

El péndulo invertido, por su naturaleza, tiene una respuesta inestable en la posición vertical, siendo este el punto para el cual se pretende diseñar el sistema de control.

1.4. Alcance

Para llegar a los objetivos que se pretenden alcanzar en el presente trabajo se contemplan los siguientes aspectos que hacen posible su alcance:

- Obtención del modelo matemático de la planta mediante el estudio de la misma.
- Simulación y análisis del comportamiento dinámico en el software Matlab/Simulink.
- Diseño de un sistema de control mediante técnica analítica, concretamente tipo LQR.
- Dimensionamiento de los componentes necesarios para la construcción de la planta.
- Diseño de un sistema de control mediante técnica experimental en la planta real.
- Análisis de los resultados obtenidos en simulación y en la planta real.
- Dentro del alcance no se contempla la caracterización del actuador.
- El estudio mecánico y estructural de la planta para su posterior construcción no se contemplan en el desarrollo del presente trabajo.

Capítulo 2

Estudio previo

2.1. El péndulo invertido

En la teoría de control, el péndulo invertido se trata de un problema muy conocido e importante debido a la no linealidad e inestabilidad del sistema en su punto de equilibrio. Es un sistema especialmente utilizado en el campo académico para la introducción de sistemas no lineales y comparación entre control en lazo abierto y lazo cerrado.

El péndulo invertido está constituido por una barra articulada en la base con un grado de libertad. En el extremo superior se concentra una masa superior a la de la barra. El objetivo del sistema de control es mantener la barra en su posición vertical aproximando el ángulo θ a 0.

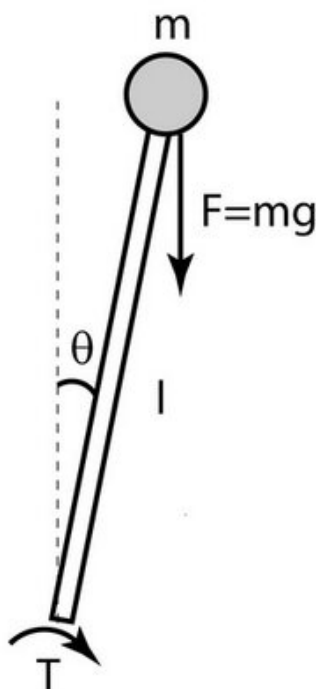


Figura 2.1: Péndulo invertido simple

Si se sitúa el péndulo en su posición vertical (punto de equilibrio) este se mantendrá en la posición hasta que se vea afectado por una fuerza externa. La pérdida de la estabilidad del péndulo producirá un movimiento oscilatorio del mismo hasta llegar a detenerse en 180° debido a las propias fuerzas externas. En el caso de que el

sistema fuese ideal, sin fuerzas de rozamiento, este se mantendría oscilando sin llegar a detenerse.

El control del sistema se realizará en torno a un punto de operación el cual se considera que en lazo cerrado será asintóticamente estable de forma local o en una vecindad del punto de equilibrio. Este punto será 0° respecto a la vertical. Una fuerza externa demasiado grande hará que el sistema se aleje del punto de operación para el cual se ha linealizado llevándolo a su inestabilidad.

2.2. Variantes del péndulo invertido

Actualmente se pueden encontrar diversas variantes del péndulo invertido, donde se ve modificado el sistema actuador e incluso los grados de libertad del péndulo. A continuación, se presentan algunas de estas modificaciones, las cuales pueden tener mayor similitud con la estudiada en este trabajo.

2.2.1. Péndulo invertido simple

Se trata del sistema más estudiado, siendo el que se va a desarrollar en el presente trabajo. Se encuentra constituido por una barra articulada en uno de sus extremos y dispone de un grado de libertad. [2.1](#)

2.2.2. Péndulo invertido doble

En este caso, el péndulo se encuentra constituido por 2 barras unidas entre ellas por una unión rotacional. En este caso se tienen 2 grados de libertad añadiéndole un extra de dificultad en el control. Un ejemplo de este sistema se puede ver en [\[2\]](#) donde los autores realizan un diseño, simulación y control de un péndulo invertido doble lineal.

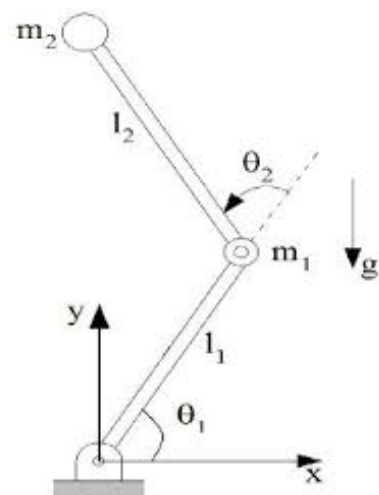


Figura 2.2: Péndulo invertido doble

2.2.3. Péndulo invertido triple

Se aumenta la dificultad en el desarrollo de un sistema de control ya que en este caso se disponen de 3 grados de libertad. El objetivo en la estrategia de control sigue siendo el mismo que en los casos anteriores, mantener las 3 barras en su posición vertical. Un estudio del modelo matemático y simulaciones del mismo se presentan en el trabajo [7].

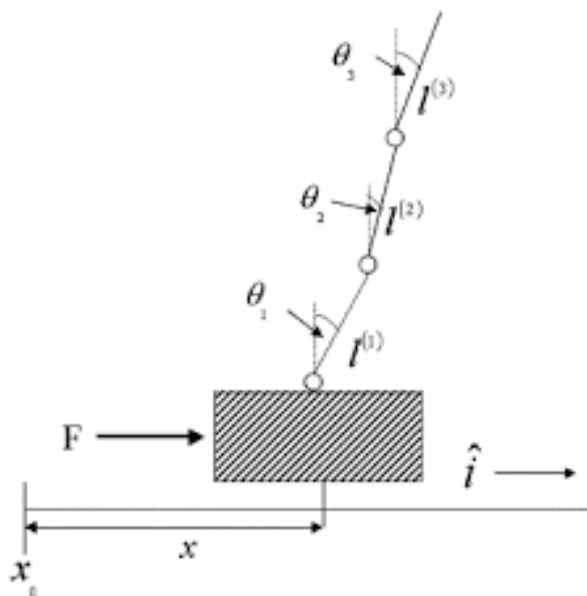


Figura 2.3: Péndulo invertido triple

2.2.4. Péndulo invertido lineal o rotacional

Se pueden encontrar variantes del sistema según se apliquen las fuerzas que establecen al sistema en su posición de equilibrio. En el caso del péndulo invertido rotacional, el actuador gira la base del péndulo con un movimiento rotacional generando una fuerza aplicada al péndulo. En el trabajo [11] los autores realizan un estudio y control sobre un péndulo invertido rotacional.

El péndulo invertido lineal (2.4a) será la variante que se estudie en este trabajo, cuyo principio se basa en contrarrestar las fuerzas que hacen perder la estabilidad al péndulo moviendo en el eje horizontal y linealmente el carro sobre el que se encuentra el péndulo invertido.

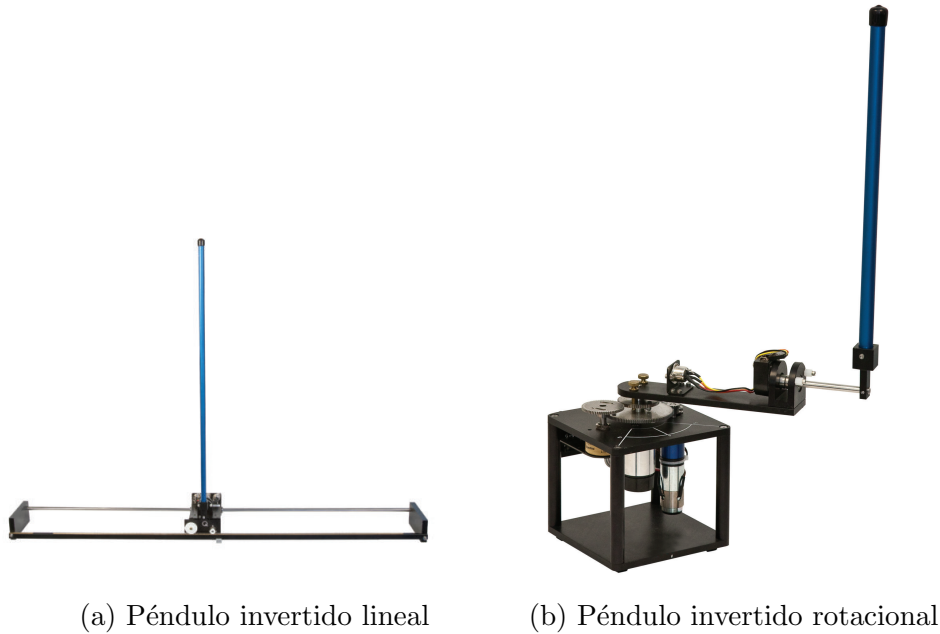


Figura 2.4: Péndulo invertido lineal y rotacional

2.3. Aplicaciones

Los sectores más destacados en los que se puede encontrar un sistema de control similar al de un péndulo invertido es: en la industria aeroespacial, en el sistema de estabilización de los cohetes tanto en el lanzamiento como en el aterrizaje (Proyecto *Starship* actualmente en desarrollo [17]); en la robótica, especialmente en prototipos educativos como, por ejemplo, el robot balancín [14]; o en el transporte con los vehículos personales conocidos como ‘Segway’ (Figura 2.5).

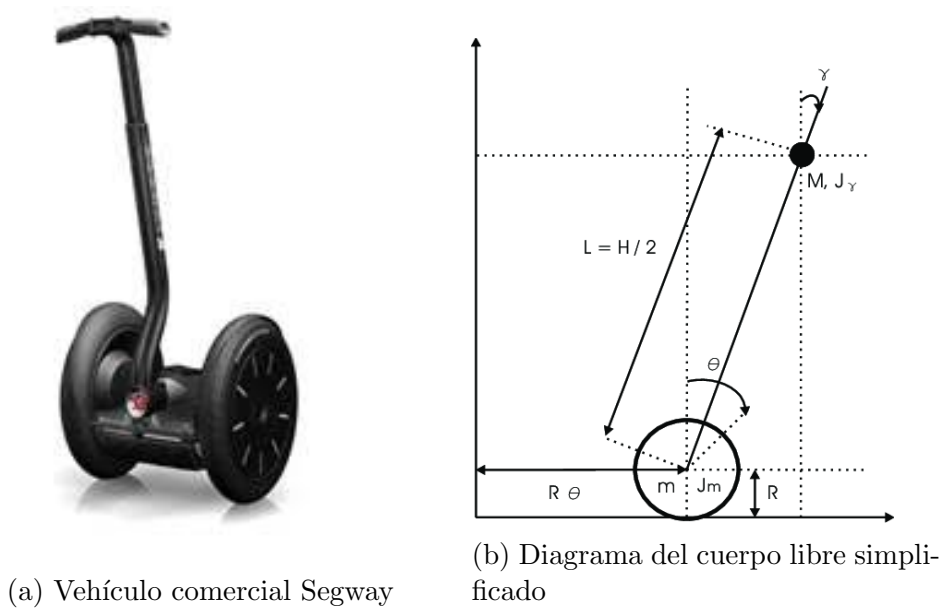


Figura 2.5: Péndulo invertido lineal y rotacional

2.3.1. Estabilización de cohetes para un aterrizaje controlado

En el ámbito aeroespacial, se puede observar como el comportamiento dinámico de un cohete se puede asemejar al de un péndulo invertido. Este debe ser estabilizado a partir de sus propulsores y mantener la trayectoria indicada (ángulo de inclinación). Sin embargo, este problema de estabilización en la actualidad sigue en continuo desarrollo.

En el lanzamiento de un cohete al espacio existen elementos como, por ejemplo, el tanque de combustible, que se destruye siendo imposible su reutilización y generando pérdidas elevadas a la compañía. Es por ello que la empresa 'SpaceX' en la actualidad se encuentra desarrollando un cohete capaz de ser reutilizado [17]. Para ello han desarrollado un sistema de control capaz de estabilizar el sistema propulsor, una vez se ha desprendido de la carga útil, consiguiendo así un aterrizaje controlado. Con ello consiguen minimizar las pérdidas y reutilizar el cohete tras un mantenimiento y puesta a punto del mismo.

El problema de control que presenta es parecido al de un péndulo invertido. En el trabajo [4] se presenta dicho problema en mayor profundidad.



Figura 2.6: Cohete Starship instantes antes del aterrizaje

Capítulo 3

Modelo matemático de la planta

3.1. Introducción

Para poder realizar un análisis del sistema en cuestión y poder ajustar un controlador que lo haga estable, es necesario obtener el modelo matemático de la planta. Para ello, se va a analizar el sistema péndulo-carro y obtener un modelo linealizado para el punto de operación 0 grados. Además, se obtendrá una representación del modelo en el espacio de estados.

Con el fin de obtener una buena aproximación del sistema, se estudiará además el modelo matemático que rige la dinámica de un motor de corriente continua. Este componente será el actuador de la planta. El modelo matemático final será el conjunto formado por péndulo-carro-motor.

Al final de este apartado, se tendrá la función de transferencia del sistema péndulo carro y la de un motor de corriente continua. Además, para que sea de utilidad en futuros estudios, se obtendrá una representación en el espacio de estados del sistema péndulo-carro ampliado con el actuador.

3.2. Modelo matemático del péndulo invertido

Para realizar el diseño del controlador se va a realizar un estudio del comportamiento de la planta a partir del diagrama de cuerpo libre del mismo. Para facilitar el modelado se hace un análisis individual de cada uno de los cuerpos de la planta, es decir, se va a analizar por un lado el carro y por otro lado el péndulo. El procedimiento llevado a cabo y los resultados obtenidos han sido contrastados con el desarrollado en el libro de Ingeniería de control moderna [13].

En la [Tabla 3.1](#) se presentan los símbolos y parámetros utilizados en el análisis.

Como se puede observar en la [Figura 3.1](#), la masa del péndulo se concentra en la parte superior de la varilla siendo este su centro de gravedad. De esta forma el modelo resultante es simplificado

Símbolo	Parámetro	Unidad de medida
M	Masa del carro	[Kg]
m	Masa del péndulo	[Kg]
g	Aceleración gravitatoria	[m/s ²]
u	Fuerza que se aplica sobre el carro	[N]
Fr	Fuerza de rozamiento	[N]
x	Posición del carro	[m]
b	Coefficiente de fricción del carro	[N/m/s]
H	Fuerza de reacción horizontal	[N]
V	Fuerza de reacción vertical	[N]
l	Longitud de la barra	[m]
I	Momento de inercia del péndulo	[kg · m ²]
θ	Ángulo del péndulo respecto a la vertical	[rad]

Tabla 3.1: Simbología y breve descripción de las variables y constantes del sistema péndulo invertido carro.

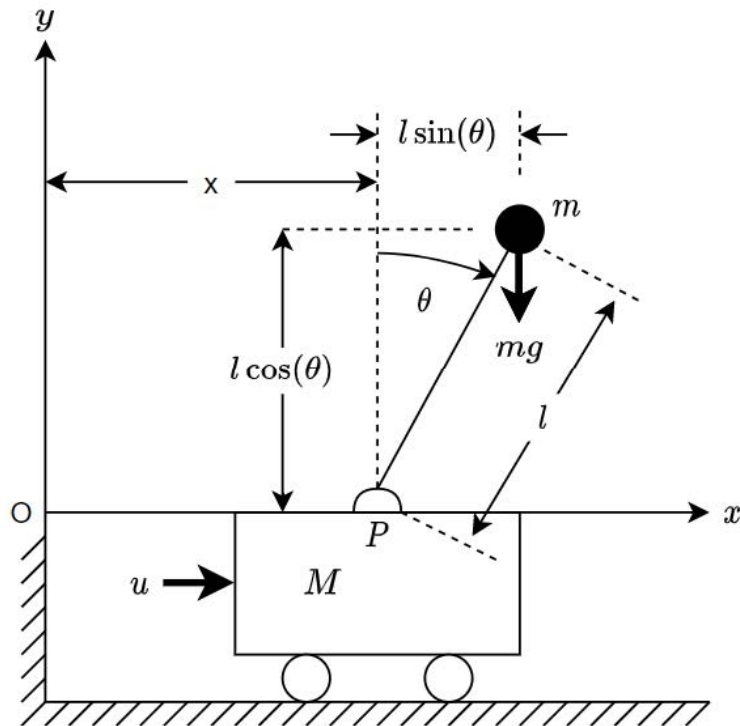


Figura 3.1: Diagrama de cuerpo libre del péndulo invertido simple.

Por lo tanto, se pueden obtener las primeras ecuaciones que representan la posición del centro de gravedad, que coincide con el centro de masa, respecto al punto O que será la referencia del sistema.

$$X_G = x + l \cdot \sin \theta \quad (3.1)$$

$$Y_G = l \cdot \cos \theta \quad (3.2)$$

En la siguiente figura se muestra el diagrama de cuerpo libre de las partes que componen el sistema. Se realiza de esta forma con el fin de facilitar el análisis.

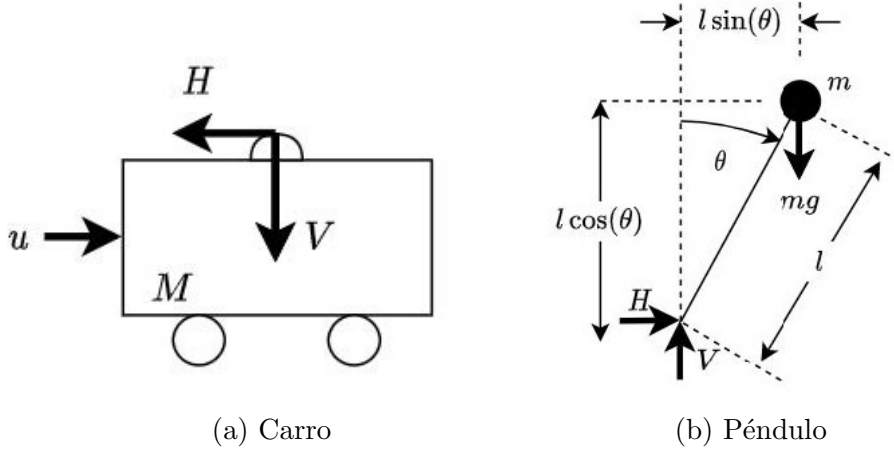


Figura 3.2: Diagrama de cuerpo libre dividido en dos partes

Aplicando la segunda ley de Newton para las fuerzas horizontales en el carro se obtiene la siguiente expresión:

$$\begin{aligned} \sum F_{x_{carro}} &= M \cdot a \\ u - Fr - H &= M \cdot \ddot{x} \end{aligned} \quad (3.3)$$

donde,

$$Fr = b \cdot \dot{x} \quad (3.4)$$

A continuación, se le aplica al péndulo la segunda ley de Newton para la rotación, la cual establece que la suma de momentos de las fuerzas que actúan sobre un cuerpo rígido alrededor de un punto G cualquiera, es igual al producto del momento de inercia por la aceleración angular.

$$\begin{aligned} \sum M_G &= I \cdot \ddot{\theta} \\ V \cdot l \cdot \sin \theta - H \cdot l \cdot \cos \theta &= I \cdot \ddot{\theta} \end{aligned} \quad (3.5)$$

Para el centro de gravedad de la barra, que se encuentra en su extremo, se le aplica la segunda ley de Newton obteniendo así las ecuaciones que representan el movimiento horizontal y vertical del mismo.

$$\begin{aligned}
\sum F_{x_{\text{pendulo}}} &= m \cdot a \\
H &= m \cdot \ddot{X}_G \\
H &= m \cdot \frac{d^2(x + l \cdot \sin \theta)}{dt^2} \\
H &= m \cdot (\ddot{x} + l \cdot (\ddot{\theta} \cdot \cos \theta - \sin \theta \cdot \dot{\theta}^2))
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
\sum F_{y_{\text{pendulo}}} &= m \cdot a \\
V - m \cdot g &= m \cdot \ddot{Y}_G \\
V - m \cdot g &= m \cdot \frac{d^2(l \cdot \cos \theta)}{dt^2} \\
V &= m \cdot g - m \cdot l \cdot (\cos \theta \cdot \dot{\theta}^2 + \sin \theta \cdot \ddot{\theta})
\end{aligned} \tag{3.7}$$

Se sustituyen las ecuaciones 3.6 y 3.7 en la ecuación 3.5, obteniendo así la primera de las ecuaciones que rigen la dinámica del sistema. En esta expresión se encuentra relacionada la velocidad angular con la posición angular y la aceleración lineal del carro.

$$\ddot{\theta} = \frac{m \cdot l \cdot (\sin \theta \cdot g - \cos \theta \cdot \ddot{x})}{I + m \cdot l^2} \tag{3.8}$$

Para obtener la segunda expresión de la dinámica del péndulo se sustituyen las ecuaciones 3.4 y 3.6 en 3.3, dando como resultado:

$$\ddot{x} = \frac{u - b \cdot \dot{x} + m \cdot l \cdot (\sin \theta \cdot \dot{\theta}^2 - \cos \theta \cdot \ddot{\theta})}{M + m} \tag{3.9}$$

En este caso, se tiene una relación entre la aceleración lineal del carro, la velocidad de este, la aceleración, velocidad, y posición angular, y el esfuerzo de control.

Por lo tanto, las ecuaciones 3.8 y 3.9 corresponden al modelo no lineal del péndulo invertido.

3.2.1. Linealización del modelo alrededor del punto de operación

El principal objetivo de control del sistema será mantener al péndulo en su posición vertical superior o punto de inestabilidad, por lo que se supone que el ángulo será muy próximo a 0 pudiéndose realizar, por lo tanto, las siguientes aproximaciones:

$$\begin{cases} \cos \theta \approx 1 \\ \sin \theta \approx \theta \\ \dot{\theta}^2 \approx 0 \end{cases}$$

Además, suponiendo que todo el peso de la barra se concentra en su punto superior, se puede despreciar el momento de inercia de la barra:

$$I \approx 0$$

Se sustituyen estas aproximaciones en las ecuaciones 3.8 y 3.9, que son las que representan la dinámica del sistema, dando como resultado las siguientes expresiones:

$$\ddot{\theta} = \frac{\theta \cdot g - \ddot{x}}{l} \quad (3.10)$$

$$\ddot{x} = \frac{u - b \cdot \dot{x} - m \cdot l \cdot \ddot{\theta}}{M + m} \quad (3.11)$$

Relacionando las ecuaciones obtenidas entre sí se pueden modificar como:

$$\ddot{\theta} = \frac{\theta \cdot g \cdot (M + m) - u + b \cdot \dot{x}}{M \cdot l} \quad (3.12)$$

$$\ddot{x} = \frac{u - b \cdot \dot{x} - m \cdot \theta \cdot g}{M} \quad (3.13)$$

En las expresiones que se han obtenido se supone que son conocidos los valores numéricos de las magnitudes y de los parámetros. Por otro lado, está la señal u que es la salida del actuador. Esta señal será dada en Newton, pues corresponde a una fuerza, por lo que se debe encontrar una expresión que relacione la tensión de entrada del motor con la fuerza que aplica a la planta.

Este procedimiento ha sido validado en el desarrollo realizado por los autores en [5] y [13] donde se obtiene el modelo matemático y linealización de un péndulo invertido. Sin embargo, en el modelo presentado no tienen en cuenta la fuerza de rozamiento, considerando que se trata de un sistema ideal. Para obtener una mejor aproximación al sistema real se ha tenido en cuenta dicha fuerza ya que se espera que tenga cierta repercusión en los resultados.

Por otro lado, los autores en [3] tienen en cuenta, además de la fuerza de rozamiento que afecta al carro, la fuerza de rozamiento rotacional de la barra. En el presente trabajo se han considerado despreciables dichas fuerzas por simplificación del modelo.

Para obtener la función de transferencia del sistema, mediante la cual se relaciona la entrada y la salida, se aplica la transformada de Laplace a las ecuaciones 3.12 y 3.13, dando como resultado:

$$\Theta(s) \cdot s^2 = \frac{\Theta(s) \cdot g \cdot (M + m) - U(s) + b \cdot X(s) \cdot s}{M \cdot l} \quad (3.14)$$

$$X(s) \cdot s^2 = \frac{U(s) - b \cdot X(s) \cdot s - m \cdot \Theta(s) \cdot g}{M} \quad (3.15)$$

A continuación, se obtiene la relación entrada/salida tanto para el ángulo de inclinación del péndulo como para la posición del carro, dando como resultados las dos funciones de transferencias que se muestran a continuación:

$$G_{pendulo}(s) = \frac{\Theta(s)}{U(s)} = \frac{s}{-M \cdot l \cdot s^3 - b \cdot l \cdot s^2 + (M + m) \cdot g \cdot s + b \cdot g} \quad (3.16)$$

$$G_{carro}(s) = \frac{X(s)}{U(s)} = \frac{-l \cdot s^2 + g}{(-M \cdot l \cdot s^3 - b \cdot l \cdot s^2 + (M + m) \cdot g \cdot s + b \cdot g) \cdot s} \quad (3.17)$$

Aunque en un principio el trabajo se centrará en el control de la inclinación del péndulo, será necesario conocer la posición del carro y cómo varía en el tiempo para saber si el controlador que se está desarrollando es realizable.

3.2.2. Representación en el espacio de estados

Una representación en el espacio de estados define la relación entre entradas, salidas y variables de estados de un modelo matemático a partir de ecuaciones diferenciales de primer orden en forma matricial.

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u(t) \\ y = C \cdot x + D \cdot u(t) \end{cases} \quad (3.18)$$

Las matrices de dicha representación son:

- A = Matriz dinámica
- B = Matriz de control
- C = Matriz de lectura
- D = Matriz de paso

Para realizar una representación del sistema es necesario obtener una asignación de las variables de estados en términos de θ y x ,

$$\begin{cases} x_1 = \theta \\ x_2 = \dot{\theta} \\ x_3 = x \\ x_4 = \dot{x} \end{cases} \quad (3.19)$$

donde θ representa el ángulo del péndulo respecto a su posición vertical, y x a la posición del carro respecto a la posición inicial. Por lo tanto, se consideran ambas variables como las salidas del sistema:

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta \\ x \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \quad (3.20)$$

A partir de las ecuaciones 3.12 y 3.13, y teniendo en cuenta la asignación de las variables que se ha realizado, se pueden obtener las siguientes asignaciones:

$$\dot{x}_1 = x_2 \quad (3.21)$$

$$\dot{x}_2 = \frac{(M+m) \cdot g}{M \cdot l} \cdot x_1 + \frac{b}{M \cdot l} \cdot x_4 - \frac{1}{M \cdot l} \cdot u(t) \quad (3.22)$$

$$\dot{x}_3 = x_4 \quad (3.23)$$

$$\dot{x}_4 = -\frac{m \cdot g}{M} \cdot x_1 - \frac{b}{M} \cdot x_4 + \frac{1}{M} \cdot u(t) \quad (3.24)$$

Por lo tanto, la representación en el espacio de estados viene dada por las siguientes ecuaciones vectoriales:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m) \cdot g}{M \cdot l} & 0 & 0 & \frac{b}{M \cdot l} \\ 0 & 0 & 0 & 1 \\ -\frac{m \cdot g}{M} & 0 & 0 & -\frac{b}{M} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{M \cdot l} \\ 0 \\ \frac{1}{M} \end{bmatrix} \cdot u(t) \quad (3.25)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (3.26)$$

En la representación de estados obtenida no se tiene en cuenta la dinámica del actuador. Posteriormente, cuando se haya obtenido el modelo del motor, se ampliarán las matrices con la dinámica del motor.

Para el diseño del controlador se utilizarán las funciones de transferencia obtenidas anteriormente. De igual modo, la representación en el espacio de estados lograda será útil para el diseño de controladores en futuros trabajos.

3.3. Modelo matemático del actuador

Como actuador para la planta se ha seleccionado un motor de corriente continua, el cual será el responsable de transmitir una fuerza al carro del péndulo. Por este motivo, se va a buscar una relación tensión-par la cual nos permita realizar un control de la fuerza aplicada al sistema.

Los motores de corriente continua son muy utilizados, debido a su bajo coste de adquisición, para diversos sistemas de prototipos mecánicos y electrónicos. Un punto importante al utilizar este tipo de motores es conocer los parámetros del mismo para así poder diseñar sistemas de control eficientes y robustos.

Los parámetros utilizados en el análisis del modelo del motor se presentan en la siguiente tabla.

Símbolo	Parámetro	Unidad de medida
R_a	Resistencia de armadura	$[\Omega]$
L_a	Inductancia de armadura	$[H]$
E_a	Fuerza contraelectromotriz	$[V]$
i_a	Corriente de armadura	$[A]$
T_m	Torque del motor	$[Nm]$
B	Coefficiente de fricción	$[N \cdot m/s]$
J	Momento de inercia	$[Kg \cdot m^2]$
ω	Velocidad angular del motor	$[rad/s]$
K_a	Constante contraelectromotriz	$[V/rad/s]$
K_m	Constante de torque	$[Nm/A]$

Tabla 3.2: Simbología y breve descripción de las constantes y variables de un motor de corriente continua

En la siguiente figura se presenta el modelo de circuito eléctrico-mecánico de un motor de corriente continua. A partir del análisis de este se obtendrán las ecuaciones matemáticas que modelan el motor.

Cabe destacar que el motor de corriente continua tiene características eléctricas y mecánicas por las que será necesario el uso de ecuaciones diferenciales para obtener el modelo en cuestión. Para obtener dichas ecuaciones se han utilizado los apuntes de las clases de la asignatura de Regulación Automática de la Universidad de Cádiz [9].

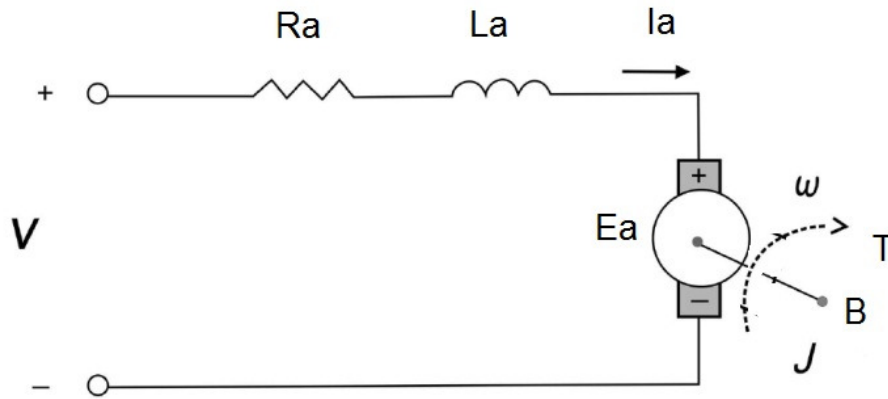


Figura 3.3: Modelo eléctrico-mecánico de un motor de corriente continua

Si se aplica un análisis por malla a la parte eléctrica del diagrama de la [Figura 3.3](#) se obtiene la siguiente ecuación:

$$V(t) = R_a \cdot i_a(t) + L_a \cdot \frac{d(i_a(t))}{dt} + E_a(t) \quad (3.27)$$

Y por otro lado, de la parte mecánica del diagrama se obtiene:

$$T_m(t) = J \cdot \dot{\omega}(t) + B \cdot \omega(t) \quad (3.28)$$

A continuación, se proponen relaciones para obtener la interacción entre la parte eléctrica y la parte mecánica.

La relación entre la tensión inducida en la armadura (E_a) y la velocidad angular del motor viene dada por la siguiente expresión, donde se da por hecho que existe una relación proporcional entre la tensión inducida y la velocidad angular.

$$E_a(t) = K_a \cdot \omega(t) \quad (3.29)$$

A partir de la siguiente expresión se relaciona el torque mecánico con la corriente eléctrica. Al igual que en la anterior, se da por hecho que tienen una relación proporcional.

$$T_m(t) = K_m \cdot i_a(t) \quad (3.30)$$

Finalmente, las ecuaciones 3.27, 3.28, 3.29 y 3.30 que se han obtenido modelan el comportamiento dinámico de un motor de corriente continua, el cual será utilizado como actuador del péndulo invertido.

De un motor de corriente continua se pueden obtener 4 salidas a partir de una entrada, es decir, es un sistema SIMO¹. En el siguiente diagrama se muestran las posibles salidas que se pueden obtener a partir de una tensión de entrada.

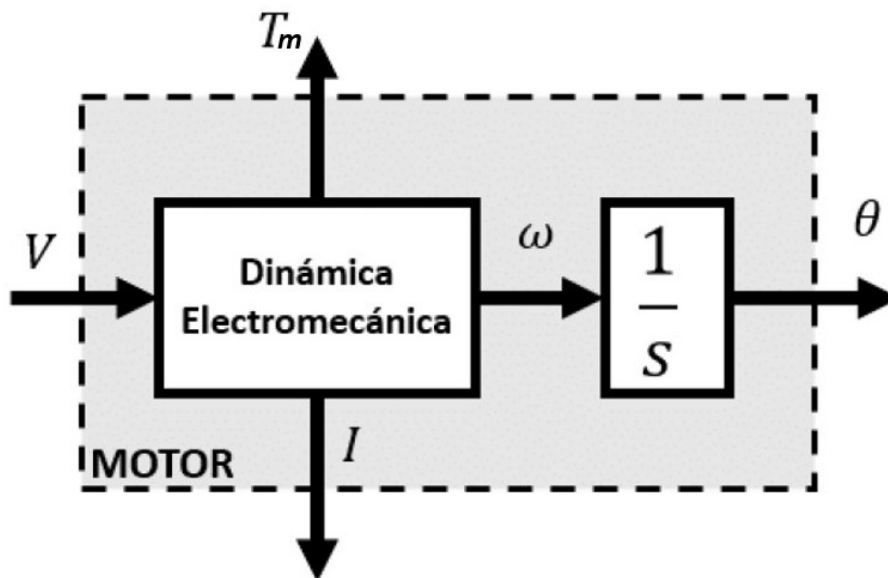


Figura 3.4: Diagrama de bloques de un motor CC

Para el caso de este proyecto se busca la función de transferencia que relacione la tensión de entrada (V) con el torque (T_m) ya que la fuerza que actúa sobre el carro es

¹SIMO - Single Input Multiple Output

medida en Newtons. Por lo tanto, se comienza aplicando la transformada de Laplace a las ecuaciones que rigen la dinámica del motor de corriente continua.

$$V(s) = R_a \cdot i_a(s) + L_a \cdot s \cdot i(s) + E_a(s) \quad (3.31)$$

$$T_m(s) = J \cdot s \cdot \omega(s) + B \cdot \omega(s) \quad (3.32)$$

$$E_a(t) = K_a \cdot \omega(s) \quad (3.33)$$

$$T_m(s) = K_m \cdot i_a(s) \quad (3.34)$$

Las ecuaciones 3.31, 3.32, 3.33 y 3.34 corresponden a la transformada de Laplace de las ecuaciones 3.27, 3.28, 3.29 y 3.30 respectivamente.

Una vez obtenida la representación en el dominio de Laplace se pueden obtener las funciones de transferencia del sistema, en este caso la que relaciona la tensión con el torque.

Se sustituyen la ecuaciones 3.33 y 3.34 en 3.31, obteniendo así la siguiente expresión:

$$V(s) = R_a \cdot \frac{T_m(s)}{K_m} + L_a \cdot s \cdot \frac{T_m(s)}{K_m} + K_a \cdot \omega(s) \quad (3.35)$$

De la ecuación 3.32 se despeja la velocidad angular (ω) y se sustituye el resultado en la ecuación 3.35.

$$V(s) = R_a \cdot \frac{T_m(s)}{K_m} + L_a \cdot s \cdot \frac{T_m(s)}{K_m} + K_a \cdot \frac{T_m(s)}{J \cdot s + B} \quad (3.36)$$

Si se reorganiza la expresión obtenida para buscar la relación T_m/V , queda la siguiente función:

$$\frac{T_m(s)}{V(s)} = \frac{K_m \cdot (J \cdot s + B)}{L_a \cdot J \cdot s^2 + (R_a \cdot J + L_a \cdot B) \cdot s + R_a \cdot B + K_m \cdot K_a} \quad (3.37)$$

El procedimiento llevado a cabo para obtener el modelo matemático de un motor de corriente continua ha sido comprobado por el desarrollado en [1].

Cabe destacar, que si se quisiese realizar, por ejemplo, un control por realimentación del vector de estados se debería medir la variable $T_m(s)$ para realimentarla. La medida de este parámetro se puede realizar directamente o indirectamente, siendo esta última forma la más económica y fácil de implementar. Este método consiste en medir la corriente (i_a) y multiplicarla por la constante de torque (K_m) para obtener así el valor del torque (Ecuación 3.30).

3.4. Espacio de estados ampliado con el actuador

Como se puede observar, las ecuaciones lineales del péndulo invertido dependen de la señal de control $u(t)$, pero a su vez, esta señal de control está dada por el motor (actuador). La relación entre la salida del motor y la entrada del sistema péndulo invertido es la siguiente:

$$u(t) = \frac{T_m(t)}{Radio} \quad (3.38)$$

En la planta real no habrá posibilidad de tener medida del torque, sin embargo este está relacionado directamente con la corriente, valor más sencillo para ser medido. Por lo tanto, la expresión queda de la siguiente forma:

$$u(t) = \frac{K_m \cdot i_a(t)}{Radio} \quad (3.39)$$

El valor *Radio* corresponde al radio de la rueda dentada acoplada al eje del motor.

A partir de las ecuaciones 3.27, 3.28, 3.29 y 3.30 se obtiene la expresión que relaciona la tensión de entrada del motor con el torque de salida. Tras sustituir 3.29 y 3.30 en 3.27, se obtiene:

$$V(t) = R_a \cdot i_a(t) + L_a \cdot \dot{i}_a(t) + K_a \cdot \omega(t) \quad (3.40)$$

En la expresión obtenida se sustituye la ecuación 3.28 por $\omega(t)$.

$$V(t) = R_a \cdot i_a(t) + L_a \cdot \dot{i}_a(t) - \frac{K_a \cdot (J \cdot \dot{\omega}(t) - K_m \cdot i_a(t))}{B} \quad (3.41)$$

A continuación, se despeja la derivada de la corriente $\dot{i}_a(t)$.

$$\dot{i}_a(t) = \frac{1}{L_a} \cdot V(t) - \left(\frac{R_a}{L_a} + \frac{K_a \cdot K_m}{B \cdot L_a} \right) \cdot i_a(t) + \frac{K_a \cdot J}{B \cdot L_a} \cdot \dot{\omega}(t) \quad (3.42)$$

La aceleración angular del motor se encuentra también relacionada con la aceleración lineal del carro a partir del radio de la rueda dentada.

$$\dot{\omega}(t) = \ddot{x}(t) \cdot Radio \quad (3.43)$$

Por lo tanto, sustituyendo la ecuación 3.13, obtenida del análisis del péndulo invertido, y multiplicada por el radio de la rueda dentada, se obtiene la siguiente expresión:

$$\dot{i}_a(t) = \frac{1}{L_a} \cdot V(t) - \left(\frac{R_a}{L_a} + \frac{K_a \cdot K_m}{B \cdot L_a} \right) \cdot i_a(t) + \frac{K_a \cdot J}{B \cdot L_a} \cdot \frac{u - b \cdot \dot{x} - m \cdot \theta \cdot g}{M} \cdot Radio \quad (3.44)$$

Finalmente, se sustituye la ecuación 3.39 y se obtiene la primera expresión para realizar la representación en el espacio de estados.

$$\begin{aligned} \dot{i}_a(t) = & \frac{1}{L_a} \cdot V(t) + \left(\frac{K_a \cdot J \cdot K_m}{B \cdot L_a \cdot M} - \frac{R_a}{L_a} - \frac{K_a \cdot K_m}{B \cdot L_a} \right) \cdot i_a(t) - \\ & - \frac{K_a \cdot J \cdot b \cdot Radio}{B \cdot L_a \cdot M} \cdot \dot{x}(t) - \frac{K_a \cdot J \cdot m \cdot g \cdot Radio}{B \cdot L_a \cdot M} \cdot \theta(t) \end{aligned} \quad (3.45)$$

Las dos ecuaciones que faltan son las obtenidas en el análisis del péndulo invertido, a las cuales hay que sustituir el valor de la entrada $u(t)$ por la relación obtenida en la ecuación 3.39.

$$\ddot{\theta}(t) = \frac{g \cdot (M + m)}{M \cdot l} \cdot \theta(t) - \frac{K_m}{Radio \cdot M \cdot l} \cdot i_a(t) + \frac{b}{M \cdot l} \cdot \dot{x}(t) \quad (3.46)$$

$$\ddot{x}(t) = \frac{K_m}{Radio \cdot M} \cdot i_a(t) - \frac{b}{M} \cdot \dot{x}(t) - \frac{m \cdot g}{M} \cdot \theta(t) \quad (3.47)$$

A partir de las ecuaciones 3.45, 3.46 y 3.47 se procede a realizar la representación en el espacio de estados. La asignación de las variables de estados son las siguientes:

$$\begin{cases} x_1 = \theta \\ x_2 = \dot{\theta} \\ x_3 = x \\ x_4 = \dot{x} \\ x_5 = i_a \end{cases} \quad (3.48)$$

donde θ representa el ángulo del péndulo respecto a su posición vertical, x a la posición del carro respecto a la posición inicial y i_a a la corriente de excitación del motor. Por lo tanto, se consideran las tres variables como las salidas del sistema:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \theta \\ x \\ i_a \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} \quad (3.49)$$

A partir de las ecuaciones 3.12 y 3.13, y teniendo en cuenta la asignación de las variables que se ha realizado, se pueden obtener las siguientes asignaciones:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{g \cdot (M + m)}{M \cdot l} \cdot x_1 - \frac{K_m}{Radio \cdot M \cdot l} \cdot x_5 + \frac{b}{M \cdot l} \cdot x_4 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{K_m}{Radio \cdot M} \cdot x_5 - \frac{b}{M} \cdot x_4 - \frac{m \cdot g}{M} \cdot x_1 \\
\dot{x}_5 &= \frac{1}{L_a} \cdot V(t) + \left(\frac{K_a \cdot J \cdot K_m}{B \cdot L_a \cdot M} - \frac{R_a}{L_a} - \frac{K_a \cdot K_m}{B \cdot L_a} \right) \cdot x_5 - \frac{K_a \cdot J \cdot b \cdot Radio}{B \cdot L_a \cdot M} \cdot x_4 - \\
&\quad - \frac{K_a \cdot J \cdot m \cdot g \cdot Radio}{B \cdot L_a \cdot M} \cdot x_1
\end{aligned}$$

Finalmente, se obtienen las matrices que forman la representación en el espacio de estados.

$$\begin{cases} \dot{x} = A \cdot x + B \cdot V(t) \\ y = C \cdot x + D \cdot V(t) \end{cases} \quad (3.50)$$

donde,

- A = Matriz dinámica
- B = Matriz de control
- C = Matriz de lectura
- D = Matriz de paso

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{g \cdot (M+m)}{M \cdot l} & 0 & 0 & \frac{b}{M \cdot l} & -\frac{K_m}{Radio \cdot M \cdot l} \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{m \cdot g}{M} & 0 & 0 & -\frac{b}{M} & \frac{K_m}{Radio \cdot M} \\ -\frac{K_a \cdot J \cdot m \cdot g \cdot Radio}{B \cdot L_a \cdot M} & 0 & 0 & -\frac{K_a \cdot J \cdot b \cdot Radio}{B \cdot L_a \cdot M} & \frac{K_a \cdot J \cdot K_m}{B \cdot L_a \cdot M} - \frac{R_a}{L_a} - \frac{K_a \cdot K_m}{B \cdot L_a} \end{bmatrix} \quad (3.51)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} \quad (3.52)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.53)$$

$$D = 0 \quad (3.54)$$

Capítulo 4

Dimensionamiento mecánico y electrónico de la planta

4.1. Introducción

Antes de proceder con el cálculo de un sistema de control que estabilice al péndulo en simulación, se va a realizar el dimensionamiento de los componentes mecánicos y electrónicos para conocer sus limitaciones. De esta forma, se podrá obtener unos resultados que se aproximen a los que se obtengan en las pruebas de la planta real.

Con el dimensionamiento mecánico se obtendrá información sobre pesos y dimensiones de las distintas partes de la planta, datos que son necesarios en el modelo obtenido en el [Capítulo 3](#).

Por otra parte, la elección de los sensores destinados a medir las variables y el hardware encargado de obtener esta información, será de utilidad en simulación para conocer la sensibilidad en las medidas, valores límites medibles y el periodo de muestreo.

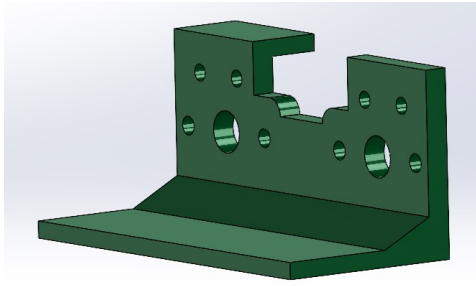
En este capítulo no se dimensionará el actuador. Este se estimará cuando se conozcan datos de fuerza y velocidad necesarios para estabilizar la planta. Es decir, tras las primeras simulaciones con un sistema de control estable se podrá dimensionar el actuador.

4.2. Sistema mecánico de la planta

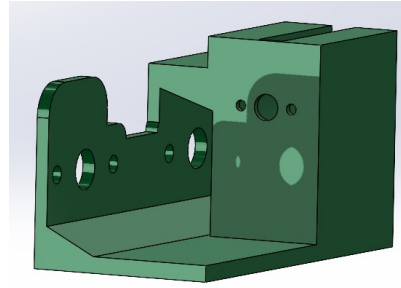
El presente trabajo no se centra en obtener un buen diseño mecánico, teniendo en cuenta todo el estudio que ello supone para que este sea robusto. El objetivo de este diseño será obtener una planta funcional que pueda ser analizada y sirva de estudio para la validación de distintos controladores. Es por ello que no se va a entrar en detalle en cuanto al diseño se refiere.

La idea principal del diseño mecánico es que el carro se desplace sobre dos varillas redondas lisas. Sobre el carro estará situado el péndulo que saldrá por el frontal del mismo. El motor transmitirá la fuerza al carro mediante una correa dentada. A continuación, se presentan las piezas del diseño mecánico.

Las varillas lisas redondas estarán sujetas por dos piezas en los extremos de la planta.



(a) Lateral izquierdo



(b) Lateral derecho

Figura 4.1: Piezas laterales de la planta

Como se puede observar, ambas piezas disponen de dos orificios en los cuales apoyarán las varillas lisas. Los orificios más pequeños son para atornillar los terminales de las varillas lisas. En el ensamblaje final se visualizará con mayor detalle.

Cabe destacar que ambas piezas no son idénticas, ya que el lateral derecho llevará montado el motor de corriente continua y el izquierdo simplemente una rueda dentada para pasar la correa.

El carro central es una de las piezas más importantes, debido a la necesidad de deslizarse con suavidad por las varillas, lleva montado el péndulo, y tiene que contar con un sistema para sujetar la correa dentada.

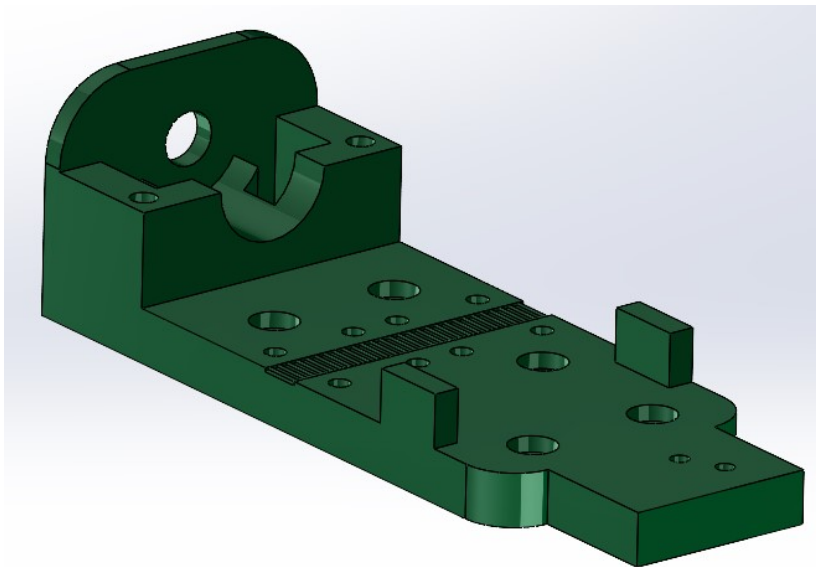


Figura 4.2: Diseño del carro

Para sujetar la correa dentada, como se puede observar en la figura, se ha realizado un carril dentado en la superficie del carro. La correa se posará sobre este carril dentado y será presionada por una pieza que, a su vez, estará atornillada al carro. Además, cuenta con el molde de un sensor tipo potenciómetro para ser instalado en el carro. La electrónica será dimensionada posteriormente.

El carro lleva atornillado dos rodamientos lineales en la base para hacerlo deslizar con suavidad por las varillas lisas.



Figura 4.3: Rodamiento lineal

Estas piezas son las más importantes del diseño, sin embargo, el ensamblaje final está compuesto por más piezas de menor importancia pero necesarias para el correcto funcionamiento de la planta.

En la siguiente figura se muestra el ensamblaje completo de todas las piezas además de otras que no han sido diseñadas, como por ejemplo: las varillas, los rodamientos, los terminales de las varillas, rueda dentada, etc.

La longitud total útil para el desplazamiento del carro es de 425 milímetros. Por lo tanto, este dato será una limitación en el diseño del controlador.

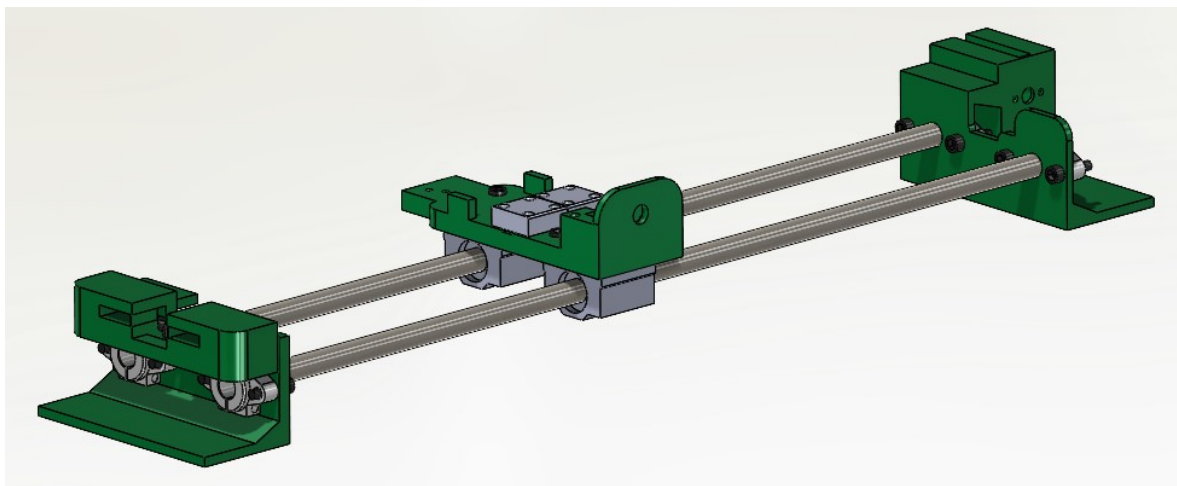


Figura 4.4: Ensamblaje del diseño mecánico de la planta

Cabe destacar que existen muchas posibilidades de diseño para un péndulo invertido sobre carro con desplazamiento lineal. El diseño se ha realizado como se muestra

en la figura 4.4 por la disponibilidad de los materiales y la simplicidad de los sistemas, pero se podría haber optado por otro sistema.

El sistema de deslizamiento se ha realizado con rodamiento lineal y varillas lisas, pero otra opción podría haber sido con guías lineales. Y para el sistema de transmisión de fuerza del motor se ha utilizado una correa dentada, aunque otra opción podría ser un husillo que desplace el carro.

Como se ha comentado, existen muchas configuraciones posibles, pero se ha realizado de esta forma por la disponibilidad de los materiales y la simplicidad de los sistemas.

Símbolo	Valor	Unidad de medida
M	0.56	[Kg]
m	0.35	[Kg]
g	9.81	[m/s ²]
b	0.05	[N/m/s]
l	0.547	[m]

Tabla 4.1: Constantes del sistema péndulo-carro.

En la tabla 4.1 se recogen los valores aproximados para el peso y longitud del péndulo, y el peso del carro.

Respecto al coeficiente de rozamiento se utilizará el mismo valor que utilizaron los autores en el artículo [3]. Se ha aproximado a este valor ya que se espera que el coeficiente de rozamiento no sea constante en todo el recorrido y, por lo tanto, no se podrá caracterizar. Es por ello que se tomará esta aproximación, o se considerará una perturbación en el sistema.

4.3. Sistema de procesamiento digital

El sistema electrónico integrado será el dispositivo encargado de recibir las medidas obtenidas de los sensores, tratarlas, ejecutar el algoritmo de control diseñado y mandar la señal de control calculada al actuador. Además, será el encargado de establecer una comunicación con el PC host para monitorizar las pruebas y ajustar el control.

En este punto no se ha realizado un estudio previo de las posibilidades y se ha decidido optar por el hardware Raspberry Pi 4B [15] debido a la experiencia adquirida en trabajos realizados durante los periodos de alumno colaborador en la Universidad de Cádiz con el grupo de investigación GAPSIS. En dichos trabajos se utilizó la Raspberry Pi junto con el software Matlab/Simulink para simulaciones de sistemas de control en 'Hardware In The Loop'. Por lo tanto, debido a la familiarización que se tiene con el dispositivo, se opta por dicha elección.

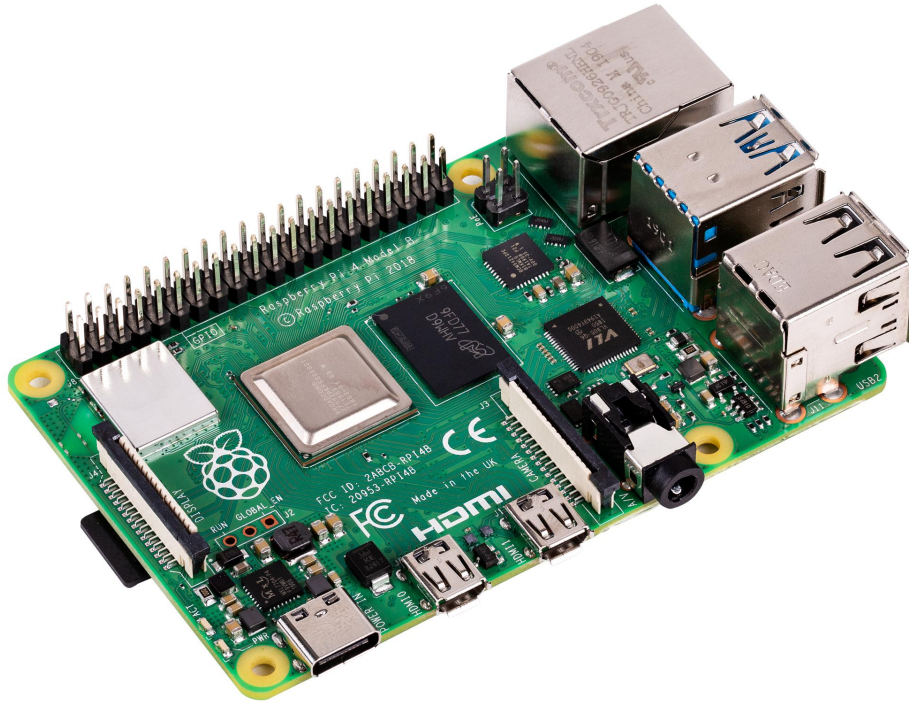


Figura 4.5: Raspberry Pi 4B

La Raspberry Pi está considerada un ordenador básico de tamaño reducido. Su relación calidad-precio ha hecho que sea un producto muy utilizado, tanto a nivel profesional como para usuarios sin amplios conocimientos de informática.

Esta versión cuenta con un procesador ARM Cortex-172 con cuatro núcleos a 1.5 GHz y una memoria RAM de 4 Gb. En este proyecto se tendrá la necesidad de hacer funcionar el algoritmo de control a una frecuencia alta para poder mantener la estabilidad en la planta.

Sin embargo, a diferencia de otros sistemas, la Raspberry Pi no cuenta con ADC¹ integrados. Por lo que será necesario contar con un circuito integrado ADC que se encargue de esta función, en concreto se utilizará el *MCP3201*. Este ADC, que cuenta con 12 bits de resolución, se conectará mediante protocolo SPI² a la Raspberry Pi para obtener la medida de aquellos sensores que tengan salida de tipo analógica.

Respecto a los tipos de señales de salida de la Raspberry Pi, se disponen de salidas digitales y con PWM. Para el caso de la señal de control será necesario usar una salida PWM.

En la siguiente imagen se muestra el *PinOut* de la Raspberry Pi 4B.

¹ADC - Analog Digital Converter

²SPI - Serial Peripheral Interface

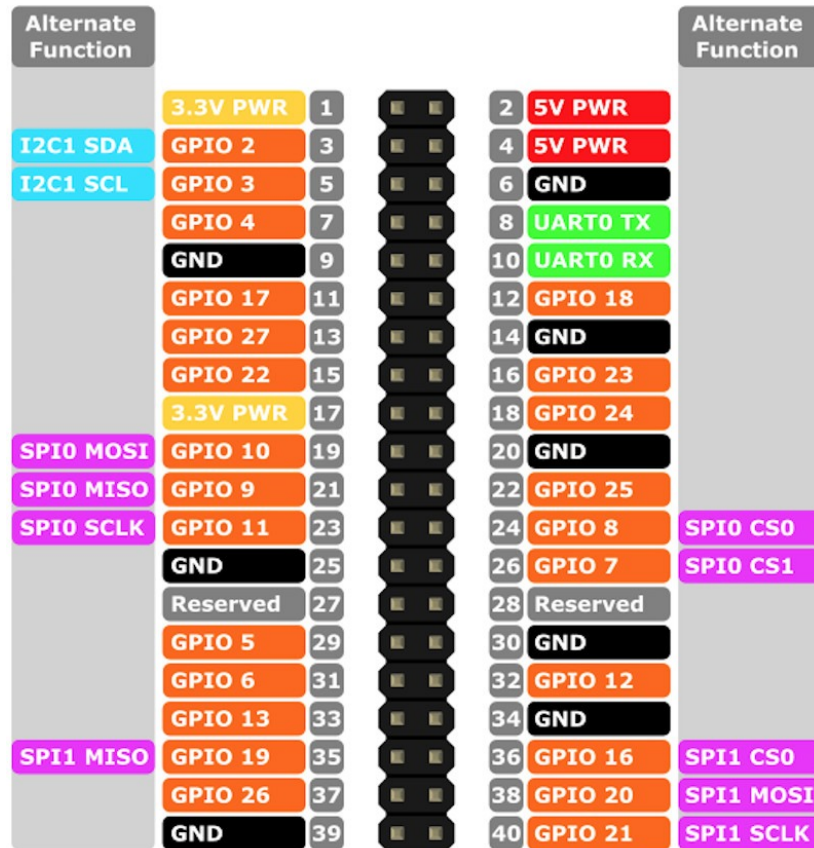


Figura 4.6: *PinOut* Raspberry Pi 4B

Posteriormente se realizará la configuración de la Raspberry Pi con el software Matlab/Simulink.

4.4. Selección de los sensores

Para cumplir con el alcance del trabajo será únicamente necesario medir dos variables físicas: la posición angular del péndulo, que corresponde a la salida en la función de transferencia 3.16, y la posición lineal del carro que corresponde a la salida de la función de transferencia 3.17. Sin embargo, dependiendo del método de control, no son las únicas variables que pueden ser necesarias.

En el caso de diseñar un control por realimentación del vector de estados, como se ha visto en el Capítulo 3, sería necesario medir cuatro variables de estados (Ecuación 3.19) si no se tiene en cuenta la dinámica del actuador, o cinco variables de estados si se tuviese en cuenta (Ecuación 3.48). Este método de control analítico supone la necesidad de obtener la medida de las derivadas tanto de la posición angular del péndulo como de la posición lineal del carro. Si además se tiene en cuenta la dinámica del motor de corriente continua, se necesitaría medir la corriente que demanda el propio actuador.

Otra forma de poder realizar un control por realimentación del vector de estados

sin tener medida de todas las variables es a partir de un observador de estados. Sin embargo, ello supone la necesidad de un modelo muy preciso de la planta y que en muchas ocasiones conlleva a un proceso de caracterización muy complejo. Esto es así ya que el observador de estados se basa en el modelo de la planta para estimar las variables.

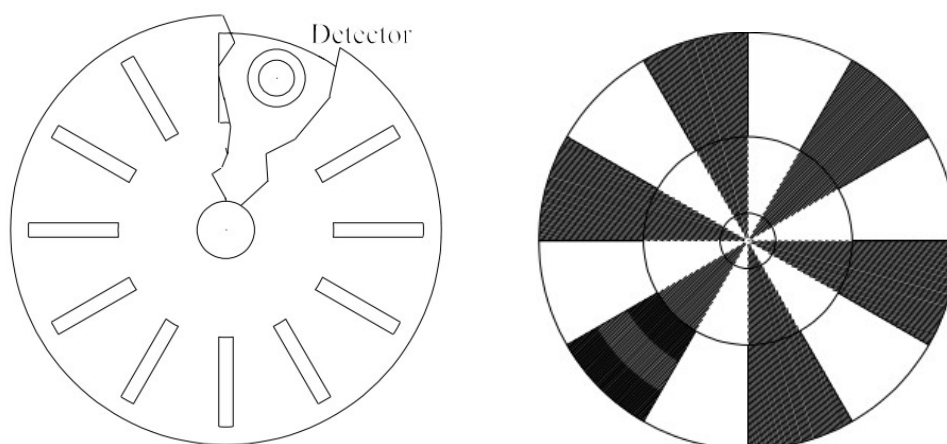
4.4.1. Sensor de posición angular del péndulo

Esta variable será la de mayor peso en el diseño del controlador ya que el objetivo principal será estabilizar el péndulo. Al tratarse de un sistema inestable, y que además la mínima perturbación hace que el sistema pierda su punto de estabilidad, será necesario que el sensor cuente con una resolución alta para poder detectar pequeños cambios en la inclinación del péndulo.

Las opciones que se han estudiado para medir la posición angular del péndulo son las siguientes:

- **Codificador incremental relativo**

También conocido en la práctica como encoder incremental. Su funcionamiento principal consiste en captar el paso de ranuras que constituyen una rueda horadada mediante un detector como se muestra en la [Figura 4.7](#).



(a) Contador tipo incremental por corte de haz entre emisor y receptor de luz (b) Ejemplo del disco de un contador incremental del tipo reflexivo

Figura 4.7: Ejemplos de codificadores incrementales. [6]

Dependiendo del número de ranuras con los que cuente la rueda, se tendrá mayor o menor resolución en la medida.

Para determinar el sentido de giro en este tipo de sensores, estos llevan instalados dos detectores desfasados entre sí. La información obtenida del sensor debe ser tratada para calcular tanto el desplazamiento como la dirección. En la siguiente figura se muestra la lectura que se obtiene de un codificador incremental relativo tras girarlo en un sentido y, posteriormente, en el sentido inverso.

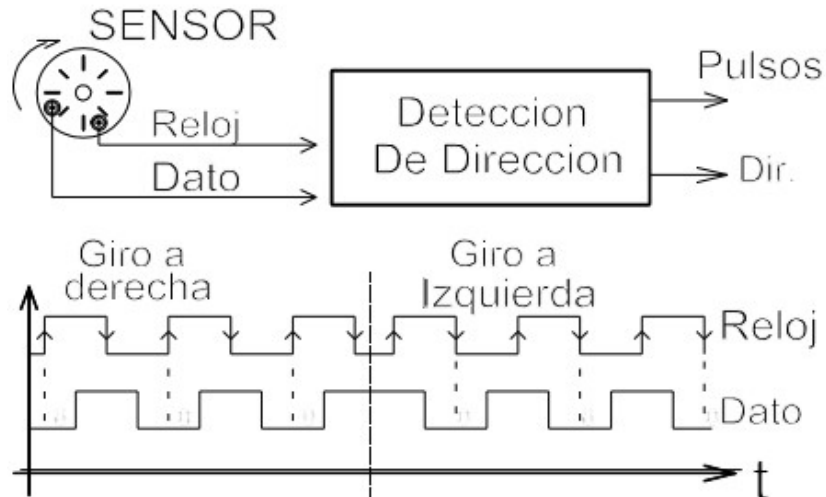


Figura 4.8: Determinación del sentido de giro en un codificador incremental mediante dos opto-sensores desfasados angularmente. [6]

En el mercado se puede encontrar un amplio rango de codificadores incrementales relativos con distintos números de ranuras, pero existe un compromiso en su elección.

Como se puede suponer, cuanto mayor resolución de la medida angular del péndulo se tenga, antes se podrá actuar sobre el control del mismo ya que se detectarán pequeñas variaciones. Sin embargo, esto conlleva a que el sistema deberá ser capaz de tomar datos a una alta frecuencia para no perder la referencia del sensor.

- **Codificador incremental absoluto**

También conocido en la práctica como encoder absoluto. Se trata de un sensor digital ya que proporciona la posición angular absoluta al devolver un número binario que depende del giro.

En este caso se tiene una gran ventaja frente al codificador relativo, puesto que se obtienen valores absolutos de posición angular. De este modo no es necesario que el sistema capte la información a una frecuencia elevada. Sin embargo, no es fácil encontrar en el mercado un codificador absoluto que nos dé la resolución necesaria para efectuar el control.

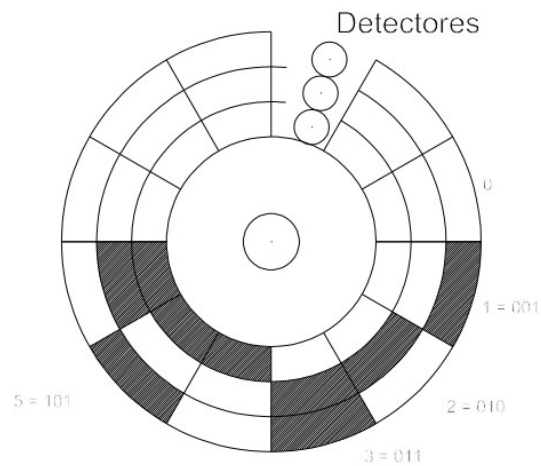


Figura 4.9: Corte de un codificador absoluto basado en el corte de los haces de luz entre emisores LED y fototransistores. [6]

- **Unidad de movimiento inercial (IMU)**

Una unidad de movimiento inercial es un dispositivo capaz de obtener velocidades angulares y aceleraciones. Por lo tanto, el valor de la posición angular del péndulo se obtendría mediante estimación.

Esta opción se deja reflejada porque podría ser de interés en el caso de realizar un control por realimentación del vector de estados en el cual se precise medir las velocidades angulares.

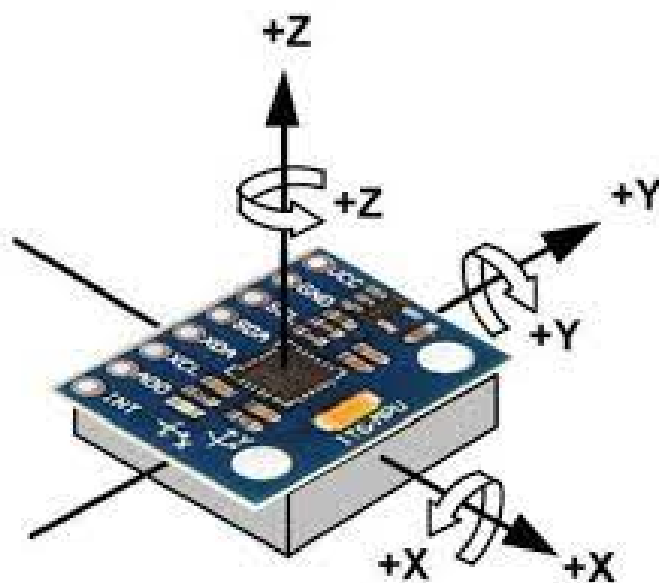


Figura 4.10: Sensor de movimiento inercial MPU6050

■ Sensor de posición de efecto hall

Por último, se plantea el uso de un sensor de posición de efecto hall. Su uso está basado en la medida del campo magnético de un imán o de una banda magnética. En el mercado se pueden encontrar de diferentes niveles de precisión y son de aplicación directa. Es decir, no será necesario tratar la información ya que este devuelve un valor de tensión dependiendo de la posición angular del eje de rotación. La relación posición angular - tensión es lineal.

Con este tipo de sensor se pueden obtener resoluciones altas y valores absolutos.



Figura 4.11: Sensor de posición de efecto hall del fabricante *TT Electronics*.

Teniendo en cuenta que el sistema péndulo invertido es un sistema inestable y que pequeñas variaciones en la posición angular del péndulo puede suponer la inestabilidad del sistema, es necesario tener una medida de dicha posición con una alta resolución. Además, nunca se debe perder la referencia en la medida debido a que ello supondría la inestabilidad de la planta.

Es por todo ello por lo que tanto el codificador incremental relativo como el absoluto no son posibles soluciones. Se pueden encontrar codificadores relativos en el mercado con grandes resoluciones, o mediante un sistema de engranajes mejorar la resolución. Pero si el sistema no es capaz de trabajar a una frecuencia lo suficientemente alta para medir cada variación, al final se acabará teniendo un error acumulado que llevará a la planta a la inestabilidad.

En el caso de los codificadores absolutos no existen soluciones comerciales que tengan una resolución lo suficientemente alta. Además, la relación precio-resolución en estos productos es exponencial debido a la dificultad en la fabricación para aumentar la resolución.

Es por ello que se ha decidido usar un sensor de posición de efecto hall. Este proporciona simplicidad, alta resolución y valores absolutos. Además, el coste de adquisición es bajo. En concreto el modelo seleccionado es el **6127V1A120L.5**, mostrado en la [Figura 4.11](#).

Esta familia de productos cuenta con una resolución de 0.088° para un trayecto de 360° y 0.011° para un trayecto de 45 grados. El trayecto disponible en el modelo

seleccionado es de 120° . Con este trayecto se tiene mayor resolución que para 360° , además de no ser necesaria mayor trayectoria ya que el péndulo se encontraría en una situación incontrolable.

Mecánicamente, no existe rozamiento en el eje más allá del propio rodamiento del que dispone el sensor.

4.4.2. Sensor de posición lineal del carro

Para que el controlador sea capaz de mantener el carro del sistema en el centro de su recorrido, será necesario conocer la posición del carro. Cabe destacar que este control no es necesario para estabilizar el péndulo, pero teniendo en cuenta que la planta tiene un recorrido limitado, se intentará que el sistema tienda a desplazarse al centro de su recorrido permisible.

Por lo tanto, teniendo en cuenta que no es una medida crítica para la estabilización de la planta, se tiene un abanico más amplio a la hora de seleccionar el sensor.

Aprovechando el estudio realizado para la posición angular del péndulo y teniendo en cuenta que las soluciones estudiadas también son válidas para medir la posición lineal del carro, se opta por usar uno de los ya estudiados.

Considerando que es necesario que el sensor mida todo el trayecto angular, que no se necesita gran resolución, y que es necesario conocer la dirección de giro, para este caso se utilizará un codificador incremental relativo.

Este tipo de sensor si es de pocos pasos por revolución el precio es bajo y la fiabilidad alta. La opción comercial elegida es el *KY-040*, típico dispositivo utilizado en aplicaciones académicas.

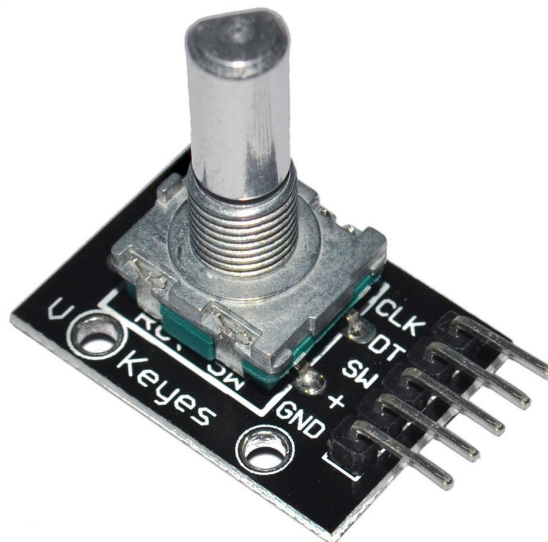


Figura 4.12: Codificador incremental relativo KY-040

Este sensor cuenta con 20 pasos por revolución y tiene tres salidas. Las tres son

de tipo digital, dos de ellas devuelven una señal cuadrada con desfase. A partir de estas dos señales se puede conocer el desplazamiento y la dirección. La otra salida, que no tiene uso en este proyecto, devuelve una señal de nivel alto cuando se pulsa el sensor.

Este dispositivo se encontrará acoplado al eje del motor y el tratamiento de las señales para obtener el desplazamiento se realizará de forma digital en la Raspberry Pi. Hay que tener en cuenta que será necesario conocer el radio del engranaje acoplado al motor y que mueve la correa dentada para poder calcular el desplazamiento.

Capítulo 5

Simulación y análisis de la planta

5.1. Introducción

Obtenidos los parámetros físicos de la planta se puede realizar un análisis de la misma para conocer la respuesta temporal ante una entrada tipo impulso tanto en lazo abierto como en lazo cerrado. En este sistema no tiene sentido realizar el análisis con otro tipo de entrada debido a la inestabilidad que presenta.

Por lo tanto, para proceder al análisis es necesario obtener las funciones de transferencia que hasta el momento se habían dejado sin determinar las constantes.

Tras sustituir las constantes del sistema péndulo-carro de la tabla [Tabla 4.1](#) en las ecuaciones que rigen la dinámica de la planta sin tener en cuenta el actuador ([Ecuación 3.16](#) y [Ecuación 3.17](#)), se obtienen las siguientes funciones de transferencia:

$$G_{pendulo}(s) = \frac{\Theta(s)}{U(s)} = \frac{-s}{0,3063 \cdot s^3 + 0,02735 \cdot s^2 - 8,927 \cdot s - 0,4905} \quad (5.1)$$

$$G_{carro}(s) = \frac{X(s)}{U(s)} = \frac{0,547 \cdot s^2 - 9,81}{0,3063 \cdot s^4 + 0,02735 \cdot s^3 - 8,927 \cdot s^2 - 0,4905 \cdot s} \quad (5.2)$$

5.2. Análisis del sistema en lazo abierto

Un sistema en lazo abierto es aquel en el que la salida medida no es comparada con la señal de referencia. Si no existe comparación de la señal de salida respecto a la de entrada no se podrá corregir dicha señal y, por lo tanto, en un sistema inestable como el que se está estudiando, este tenderá a la inestabilidad.

Para obtener la ganancia, polos y ceros del sistema en lazo abierto, se hace uso del comando `zpk()` en el software Matlab, obteniéndose así:

$$G_{pendulo}(s) = \frac{-3,2646 \cdot s}{(s - 5,381) \cdot (s + 5,416) \cdot (s + 0,05494)} \quad (5.3)$$

$$G_{carro}(s) = \frac{1,7857 \cdot (s - 4,235) \cdot (s + 4,235)}{(s - 5,381) \cdot (s + 5,416) \cdot (s + 0,05494) \cdot s} \quad (5.4)$$

Como se puede observar, el sistema en ambas funciones tiene polos positivos, lo que demuestra que se trata de un sistema inestable.

Se puede ver una representación de los polos y ceros de la función de transferencia del péndulo, la cual presenta mayor interés, con el comando `pzmap()` en el software Matlab.

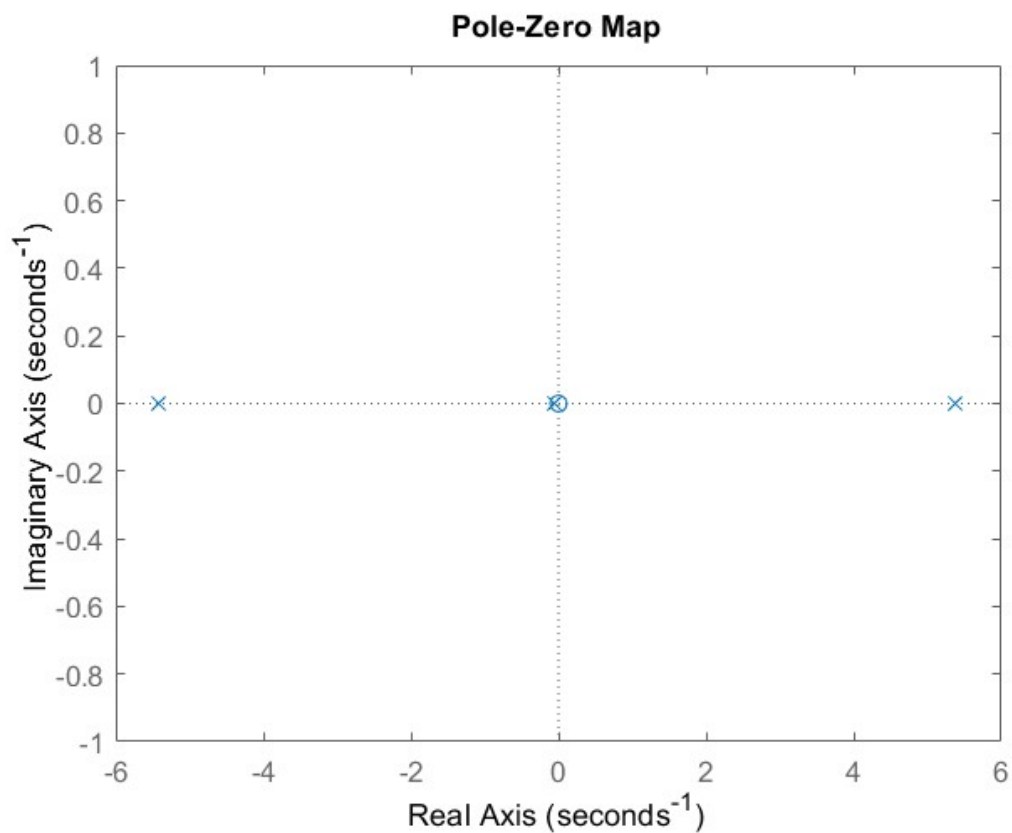


Figura 5.1: Mapa de polos y ceros del sistema en lazo abierto [5.3](#)

En la siguiente figura se puede observar como el sistema ante una entrada tipo impulso Delta de Dirac responde de forma inestable llegando a divergir. Es la respuesta que se espera obtener del sistema que se está estudiando ante cualquier perturbación cuando el péndulo se encuentra en su posición vertical y sin la aplicación de un controlador.

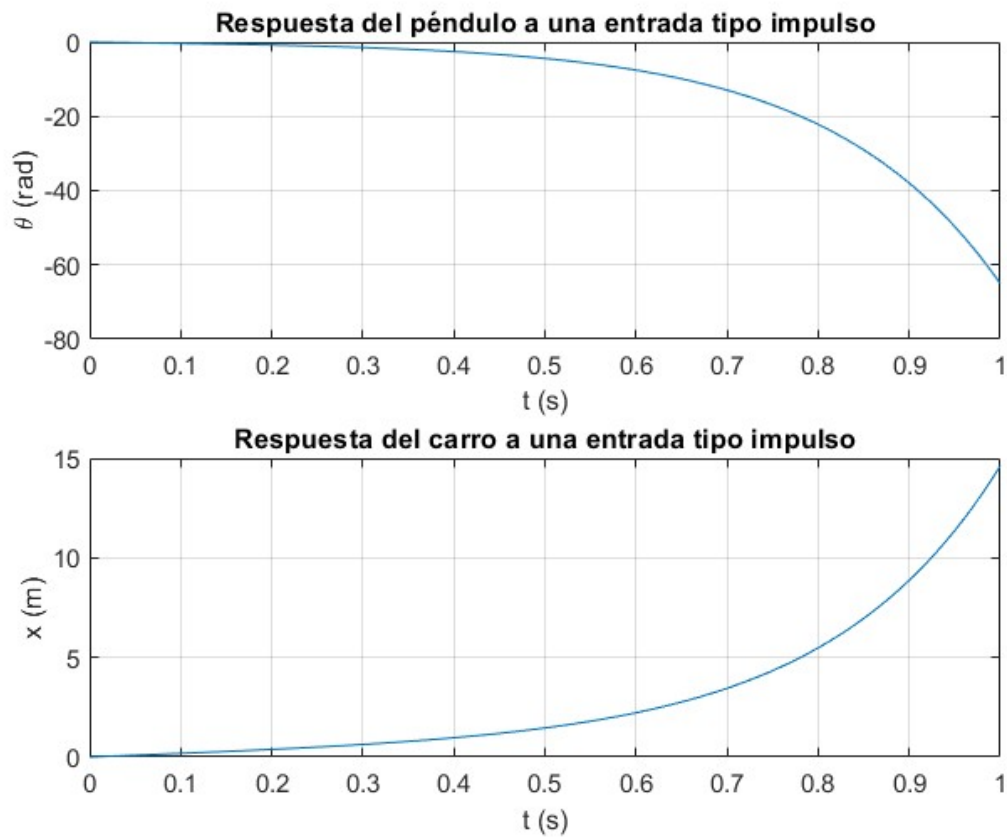


Figura 5.2: Respuesta del sistema en lazo abierto ante una entrada tipo impulso

Se puede observar que, al excitar al sistema con una fuerza aplicada en el carro, el péndulo tiende a modificar su ángulo de forma negativa en un tramo de tiempo relativamente corto. Que el ángulo sea negativo se debe a que la fuerza se aplica en dirección positiva (se impuso positivo a la derecha en el análisis del sistema), por lo tanto el péndulo oscilará en dirección inversa a las agujas del reloj (se impuso positiva la dirección de las agujas del reloj en el análisis del sistema).

Cabe destacar que esta respuesta es para el sistema linealizado. Como se puede intuir, el sistema real no tendría esta respuesta ya que su estabilidad la encontraría en la posición inversa a la que se encuentra.

Mediante el uso del comando *damp()* en el software Matlab, se pueden verificar los polos y ceros del sistema, constante de tiempo, frecuencia y amortiguamiento.

Polos	Amortiguamiento	Frecuencia [rad/s]	Constante de tiempo [s]
-0,0549	1	0,0549	18,2
5,38	-1	5,38	-0,186
-5,42	1	5,42	0,185

Tabla 5.1: Resultados del análisis en lazo abierto del péndulo invertido.

Como se puede observar, el sistema presenta un polo muy cercano al origen (-

0.0549) el cual se puede aproximar a una acción integral, y otro polo en -5.42. Un sistema cuenta con acción integral cuando tiene uno de los polos en el origen. Además el sistema cuenta con un polo positivo (5.38), lo que significa que se trata de un sistema inestable.

Existen dos polos estables, uno situado en -0.0549, que se puede aproximar por un polo en el origen, y otro cuya constante de tiempo asociada es 0.185 segundos. Para elegir el periodo de muestreo a usar en el sistema de control en tiempo discreto, Tm , se utilizan como valores orientativos los resultantes de dividir la constante de tiempo asociada, $\tau = 0,185$ *segundos*, por 10 y por 20; y a partir de los valores obtenidos decidir un valor coincidente o próximo a los obtenidos al hacer $\tau/10$, $\tau/20$ ([9], [12]).

5.3. Análisis del sistema en lazo cerrado

Debido a que la dinámica del actuador es desconocida se va a suponer que este presenta una dinámica mucho más rápida que la del péndulo invertido, sin modificar en magnitud al valor de entrada y mostrando en todo momento una respuesta asintóticamente estable sin overshoot. Por lo tanto, se supone que la función de transferencia del actuador es 1.

Tras calcular la función de transferencia del sistema en lazo cerrado para el caso de la salida de la posición angular del péndulo, se obtienen los siguientes polos.

Polos	Amortiguamiento	Frecuencia [rad/s]	Constante de tiempo [s]
-0,0494	1	-0,0494	20,2
-0,0549	1	-0,0549	18,2
5,38	-1	5,38	-0,186
-5,42	1	-5,42	0,185
5,67	-1	5,67	-0,176
-5,71	1	-5,71	0,175

Tabla 5.2: Resultados del análisis en lazo cerrado del péndulo invertido.

En la tabla 5.2 se observa como el sistema tiene dos polos positivos, lo que indica que se trata de un sistema inestable y es necesario el uso de un controlador que modifique dichos polos positivos del sistema, logrando que la planta presente una respuesta temporal estable.

En la siguiente figura se muestra el mapa de polos y ceros del sistema en lazo cerrado para la salida de la posición angular del péndulo.

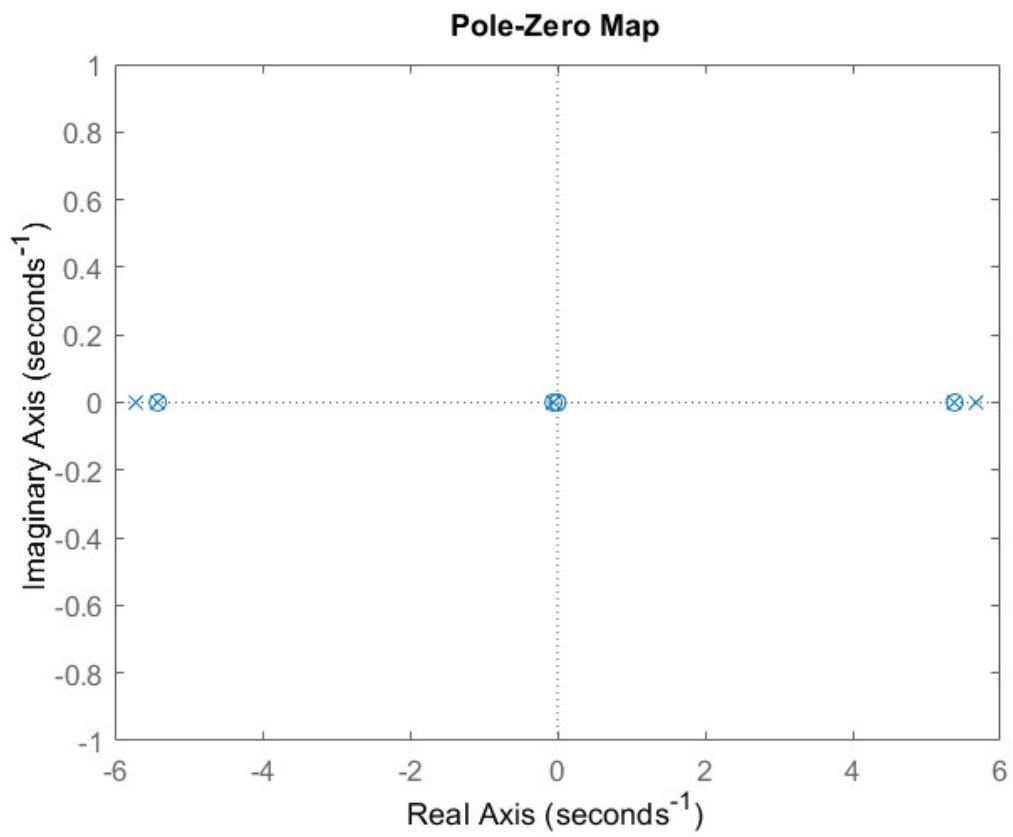


Figura 5.3: Mapa de polos y ceros del sistema en lazo cerrado sin controlador

Capítulo 6

Diseño de un sistema de control mediante técnica analítica basada en el método LQR

6.1. Introducción

Como se ha demostrado en el apartado anterior, el sistema presenta inestabilidad tanto en lazo abierto como en lazo cerrado sin controlador. Por lo tanto, en este apartado se va a proceder a calcular un controlador mediante una técnica analítica que haga que el sistema tenga todos los polos con signo negativo, es decir, que estabilice a la planta.

El controlador que se va a obtener será por realimentación del vector de estados, en concreto, un control tipo LQR¹. El objetivo de un controlador por realimentación del vector de estados es el de obtener un sistema estable. Para ello, se realimenta el vector de estados y se multiplica por una matriz de ganancia K_c en el lazo cerrado. Como resultado se obtendrá un sistema con unos polos modificados con parte real negativa.

El procedimiento y los conocimientos necesarios para el desarrollo de este controlador se han obtenido en las asignaturas [9] y [8].

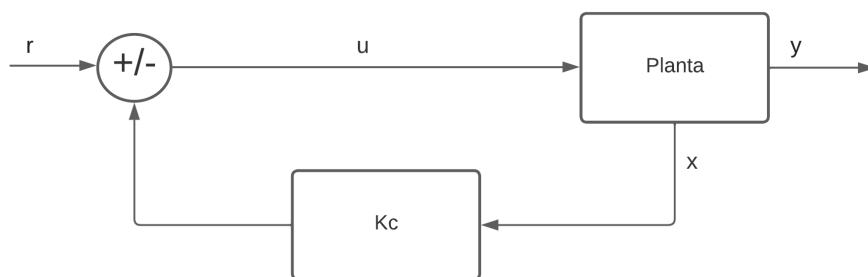


Figura 6.1: Diagrama de bloques de un sistema de control por realimentación del vector de estados

Para realizar el diseño de dicho controlador se van a utilizar las matrices que forman la representación en el espacio de estados que se calcularon en el capítulo 3. Dichas matrices se presentan de nuevo a continuación, ya que serán la base para el cálculo del controlador

¹LQR - Linear-quadratic regulator

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u(t) \\ y = C \cdot x + D \cdot u(t) \end{cases} \quad (6.1)$$

donde,

- A = Matriz dinámica
- B = Matriz de control
- C = Matriz de lectura
- D = Matriz de paso

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{g \cdot (M+m)}{M \cdot l} & 0 & 0 & \frac{b}{M \cdot l} & -\frac{K_m}{Radio \cdot M \cdot l} \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{m \cdot g}{M} & 0 & 0 & -\frac{b}{M} & \frac{K_m}{Radio \cdot M} \\ -\frac{K_a \cdot J \cdot m \cdot g \cdot Radio}{B \cdot L_a \cdot M} & 0 & 0 & -\frac{K_a \cdot J \cdot b \cdot Radio}{B \cdot L_a \cdot M} & \frac{K_a \cdot J \cdot K_m}{B \cdot L_a \cdot M} - \frac{R_a}{L_a} - \frac{K_a \cdot K_m}{B \cdot L_a} \end{bmatrix} \quad (6.2)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} \quad (6.3)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

$$D = 0 \quad (6.5)$$

Antes de continuar, cabe destacar que en el diseño de un controlador por métodos analíticos conlleva un trabajo previo de caracterización muy laborioso, esto es así debido a que el control está basado en el modelo de la planta. Si el modelo no es preciso, el controlador funcionará bien en simulación, pero no necesariamente ocurrirá lo mismo cuando se pruebe en la planta real. Es decir, el potencial de aplicación de un control LQR se basa en tener un modelo matemático preciso.

De igual modo, el controlador que se obtenga en este apartado no será el que se utilice en las pruebas con la planta real ya que no se dispondrá de los sensores necesarios para medir todas las variables del vector de estados. Además, aunque se ha intentado realizar una buena caracterización de la planta, existen imperfecciones que hacen que el modelo obtenido no sea lo suficientemente preciso.

Para poder calcular el controlador es necesario caracterizar el actuador. Para ello se utilizarán los parámetros que nos da el fabricante sobre un motor comercial de corriente continua que se encuentra instalado en un banco de trabajo (Servosistema Feedback ES-151) del laboratorio de Regulación Automática de la Universidad de Cádiz.

Símbolo	Valor	Unidad de medida
J	0.2	$[Kg \cdot m^2]$
B	0.0001	$[N \cdot m/s]$
K_m	1.5	$[Nm/A]$
K_a	1.5	$[V/rad/s]$
R_a	0.2	$[\Omega]$
L_a	0.002	$[H]$

Tabla 6.1: Constantes motor de corriente continua comercial.

6.2. Análisis de la planta con actuador en lazo abierto

Para ver la respuesta del sistema en lazo abierto se excita con una entrada tipo impulso Delta de Dirac. El resultado obtenido es similar al sistema péndulo invertido sin motor.

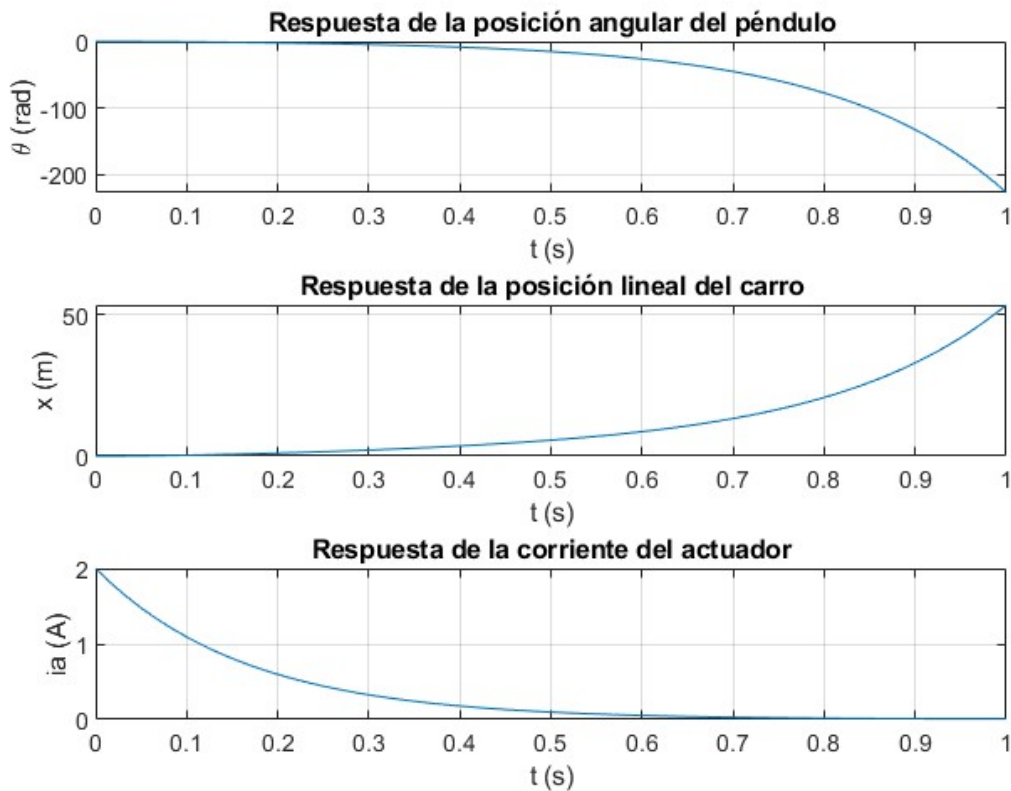


Figura 6.2: Respuesta de la planta con actuador ante una entrada tipo Delta de Dirac

La entrada impulso supone una entrada de tensión en el sistema de 1 voltio en un instante de tiempo ínfimo. Como se puede observar en la figura 6.2, el péndulo ante

la entrada responde con un decremento en su posición angular, lo que supondría que el péndulo giraría en el sentido inverso de las agujas del reloj. Esto tiene sentido porque el péndulo en un inicio se encuentra en una posición de inestabilidad centrado en la vertical. Tras la excitación, el carro tenderá a desplazarse de forma positiva hacia la derecha obteniéndose la respuesta que se ve en la [Figura 6.2](#).

Para obtener la frecuencia natural y constante de tiempo de los polos del sistema, se utiliza la función `damp()` en Matlab, obteniendo así los siguientes resultados:

Polos	Frecuencia [rad/s]	Constante de tiempo [s]
0	0	<i>Inf</i>
-0,0549	0,0549	18,2
5,38	5,38	-0,186
-5,42	5,42	0,185
-6,04	6,04	0,166

Tabla 6.2: Resultados del análisis en lazo cerrado de la planta con actuador.

Del sistema en lazo abierto se obtienen 5 polos, uno de ellos en el origen, lo que significa que el sistema cuenta con acción integral. La inestabilidad del sistema se encuentra en el polo positivo.

6.3. Diseño del sistema de control LQR

El modelo del sistema a controlar es el ya expuesto anteriormente y que se recuerda en la siguiente ecuación.

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u(t) \\ y = C \cdot x + D \cdot u(t) \end{cases} \quad (6.6)$$

donde $u(t)$ en este caso será la tensión de entrada al sistema.

El problema se va a tratar como un problema de regulación, es decir, no habrá seguimiento de set-point. Por lo tanto, el diagrama de simulación para el control por realimentación del vector de estados tendrá la siguiente forma:

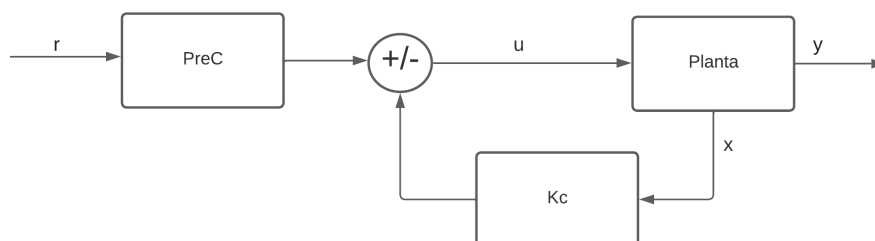


Figura 6.3: Diagrama de bloques de control por realimentación del vector de estados

El precompensador es utilizado para compensar la ganancia del sistema. En el desarrollo de este control, la regulación se hará para la condición de equilibrio 0. Es por ello que se podrá obviar el cálculo del precompensador.

Para realizar el control mediante la realimentación del vector de estados con la matriz K_c será necesario medir todas las variables de estados. Si esto no fuese posible, sería necesario el diseño de un observador de estados que estime los estados no medibles por los respectivos sensores.

Siguiendo el método expuesto en la figura 6.3, la ley de control resultante viene impuesta por la siguiente expresión:

$$u(t) = -K_c \cdot x(t) + PreC \cdot r(t) \quad (6.7)$$

A continuación, se sustituye la ley de control de la ecuación 6.7 en el sistema de ecuaciones que representa al sistema en el espacio de estados.

$$\begin{cases} \dot{x} = A \cdot x + B \cdot (-K_c \cdot x + PreC \cdot r) \\ y = C \cdot x + D \cdot (-K_c \cdot x + PreC \cdot r) \end{cases} \quad (6.8)$$

Operando se llega a las siguientes expresiones donde se distinguen las matrices del sistema en lazo cerrado.

$$\begin{cases} \dot{x} = \underbrace{(A - B \cdot K_c)}_{A_{yr}} \cdot x + \underbrace{(B \cdot PreC)}_{B_{yr}} \cdot r \\ y = \underbrace{(C - D \cdot K_c)}_{C_{yr}} \cdot x + \underbrace{(D \cdot PreC)}_{D_{yr}} \cdot r \end{cases} \quad (6.9)$$

En el sistema de ecuaciones obtenidos se desconoce, por el momento: el precompensador, el cual se obtiene igualando la ecuación de un sistema en lazo cerrado en $s = 0$ a la matriz identidad; y la matriz de realimentación del vector de estados K_c .

La matriz K_c se va a obtener mediante la técnica de diseño de control LQR, la cual se basa en una realimentación del vector de estados. Esta técnica se centra en la obtención de una matriz (K_c) que haga al sistema asintóticamente estable. Para ello, el control LQR utiliza una función de coste,

$$J = \int_{t_0=0}^{t_f \rightarrow \infty} (x^T \cdot Q_c \cdot x + u^T \cdot R_c \cdot u) \cdot dt \quad (6.10)$$

donde se deben seleccionar unos valores para las matrices Q_c y R_c de forma arbitraria, donde Q_c le dará peso a la medida del error, y R_c le dará peso al esfuerzo de control. La idea es llegar a obtener unos parámetros con los cuales se consiga una sintonía que satisfaga los requisitos del diseñador.

En el caso estudiado, la dimensión de la matriz Q_c será de 5 ya que se tienen 5 estados, y la dimensión de la matriz R_c será 1, coincidiendo así con el número de entradas del sistema.

A continuación, se eligen unos valores de forma arbitraria con los cuales empezar un proceso de sintonización. Se le va a dar importancia solo a la medida del error de la posición del carro y la posición angular del péndulo, teniendo esta última un mayor peso en el sistema.

$$Q_c = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 \\ 0 & 0 & 0 & 0 & q_5 \end{bmatrix} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.11)$$

$$R_c = 0,1 \quad (6.12)$$

A partir de la matriz Q_c , cuya diagonal le da peso a las medidas del error de cada estado, y R_c , que le dará peso al esfuerzo de control, se obtiene la matriz K_c .

Realizar el cálculo que obtiene la matriz K_c resultaría resolver una ecuación algebraica de Riccati (ARE). Para ello, el software Matlab ofrece el comando $lqr()$ que devuelve la solución a dicho cálculo a partir de las matrices A , B , Q_c y R_c .

$$K_c = [-57,6342 \quad -10,5598 \quad -10 \quad -10,0826 \quad 15,3784] \quad (6.13)$$

A partir de la matriz K_c se pueden obtener las matrices A y C en lazo cerrado.

$$\begin{cases} A_{yr} = A - B \cdot K_c \\ C_{yr} = C - D \cdot K_c \end{cases} \quad (6.14)$$

Para obtener las matrices B_{yr} D_{yr} es necesario calcular antes la matriz del precompensador, la cual se obtiene igualando la ecuación del sistema en lazo cerrado para $s = 0$ a la matriz identidad.

$$G_{yr}(s) = C_{yr}(s \cdot I - A_{yr})^{-1} \cdot B_{yr} + D_{yr} \quad (6.15)$$

$$\begin{aligned} G_{yr}(s = 0) &= I \\ C_{yr} \cdot (-A_{yr})^{-1} \cdot B_{yr} + D_{yr} &= I \\ C_{yr} \cdot (-A_{yr})^{-1} \cdot B \cdot PreC + D \cdot PreC &= I \\ [C_{yr} \cdot (-A_{yr})^{-1} \cdot B + D] \cdot PreC &= I \\ [C_{yr} \cdot (-A_{yr})^{-1} \cdot B + D]^{-1} &= PreC \end{aligned} \quad (6.16)$$

Como se explicó anteriormente, al tratarse de un problema de regulación cuyo propósito es estabilizarse en su condición de equilibrio, se puede obviar el cálculo del precompensador. Por ello, las matrices en lazo cerrado son las siguientes:

$$\begin{cases} A_{yr} = A - B \cdot K_c \\ B_{yr} = B \cdot PreC = B \cdot 0 = 0 \\ C_{yr} = C - D \cdot K_c = C - 0 \cdot K_c = C \\ D_{yr} = D \cdot PreC = 0 \cdot 0 = 0 \end{cases} \quad (6.17)$$

Con las matrices obtenidas se obtiene la representación en lazo cerrado del sistema con controlador LQR.

A continuación, se calculan los polos para verificar que el sistema sea estable.

Polos	Amortiguamiento	Frecuencia [rad/s]	Constante de tiempo [s]
$-1,34 + 1,13i$	0,765	1,75	0,748
$-1,34 - 1,13i$	0,765	1,75	0,748
$-8,57 + 13,1i$	0,547	15,7	0,117
$-8,57 - 13,1i$	0,547	15,7	0,117
$-17,1$	—	17,1	0,0586

Tabla 6.3: Resultados del análisis en lazo cerrado de la planta con actuador.

Como se puede observar, el sistema en lazo cerrado cuenta con todos los polos con parte real negativa; por lo tanto, se puede decir que el sistema de control calculado de forma analítica es asintóticamente estable en lazo cerrado. Además, cuenta con dos polos complejos conjugados en distintas frecuencias. Estos polos indican que el sistema tendrá una respuesta amortiguada.

En la siguiente figura se muestra el mapa de polos y ceros del sistema de control obtenido, donde se ven representados estos valores.

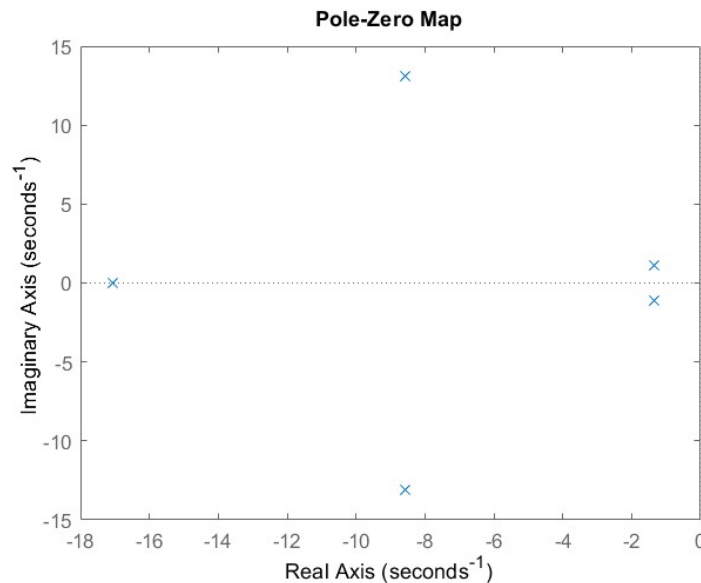


Figura 6.4: Mapa de polos y ceros del sistema de control en lazo cerrado para el controlador LQR.

Para elegir un paso de integración correcto con el cual realizar las simulaciones, se va a emplear la constante de tiempo más pequeña que dispone el sistema. De la [Tabla 6.3](#) se obtiene que la constante de tiempo asociada al sistema más pequeña es 0.0586 segundos. Por lo tanto, en las simulaciones se deberá utilizar un paso de integración menor que este valor.

6.4. Simulaciones y análisis del sistema de control LQR obtenido

Para realizar un análisis y comparar los resultados del modelo linealizado y no linealizado, se va a realizar una serie de simulaciones en el Software Matlab/Simulink en las cuales se podrá comparar tiempos de respuestas, magnitud de la acción de control y estabilidad de la planta.

El diagrama de simulación utilizado para verificar el funcionamiento del sistema de control diseñado es el que se muestra en la siguiente figura.

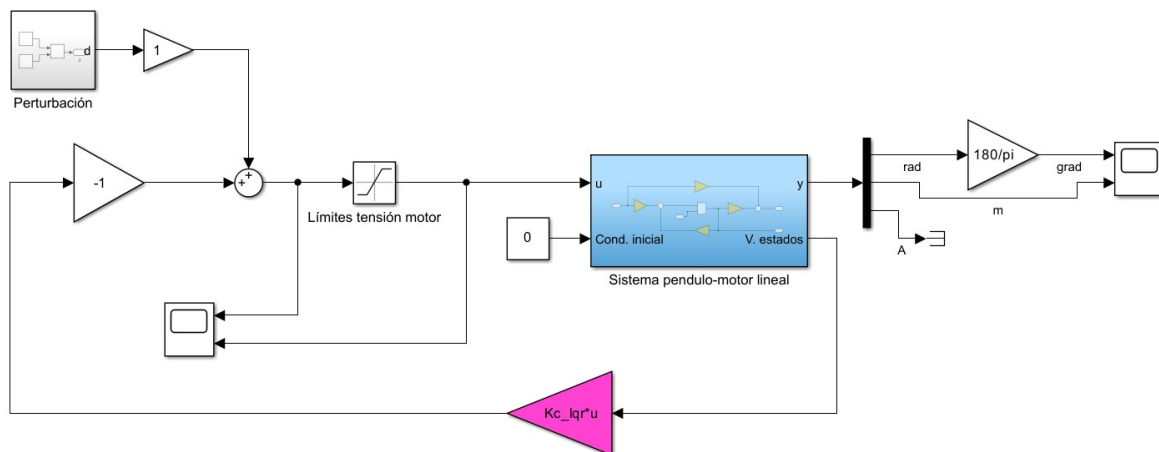


Figura 6.5: Diagrama de simulación del sistema péndulo-motor linealizado con control LQR.

En este caso, se va a simular el sistema de control con la planta linealizada y posteriormente se comparará la respuesta con la planta no lineal. Para la planta lineal se han utilizado las matrices 6.2, 6.3, 6.4 y 6.5. La perturbación a la que se enfrenta ocurre al segundo de simulación con una magnitud de 5 voltios sumados a la señal de control. Esta perturbación perdura en el tiempo 0.01 segundos, mismo valor que el paso de integración.

Como se puede observar en el diagrama de simulación, el precompensador no afectaría en ningún momento al sistema de control puesto que la entrada sería siempre 0. Además, el sistema se termina regulando automáticamente ya que la señal de control no será nula hasta que todos los estados de la planta no sean nulos también, es decir, todos los estados tienen que converger a 0.

El diagrama de simulación de la planta linealizada realizado con las matrices del espacio de estados es el mostrado en la figura 6.6.

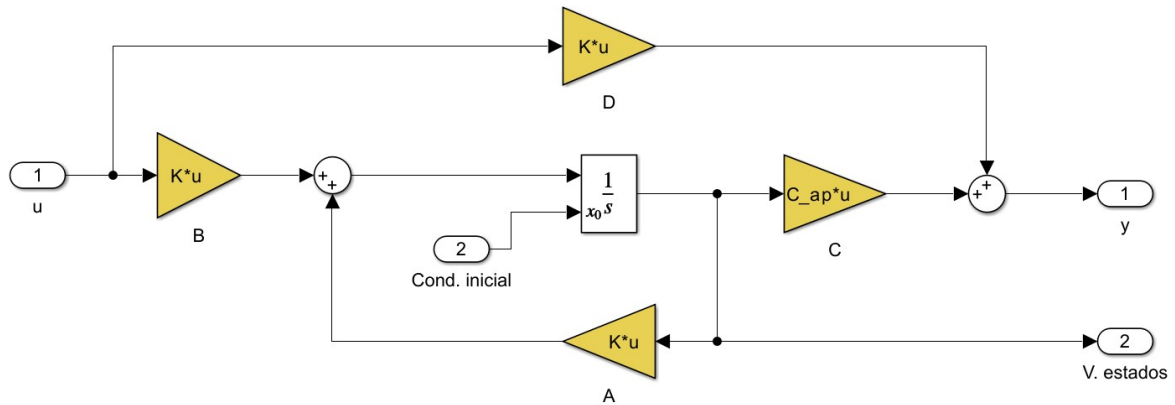


Figura 6.6: Diagrama de simulación de la planta linealizada en el espacio de estados.

Los resultados de dicha simulación para un paso de integración de 0.01 segundos en tiempo continuo y una duración de 5 segundos, son los mostrados en la siguiente figura. El método numérico utilizado para resolver la simulación es el *Runge-Kutta* de cuarto orden.

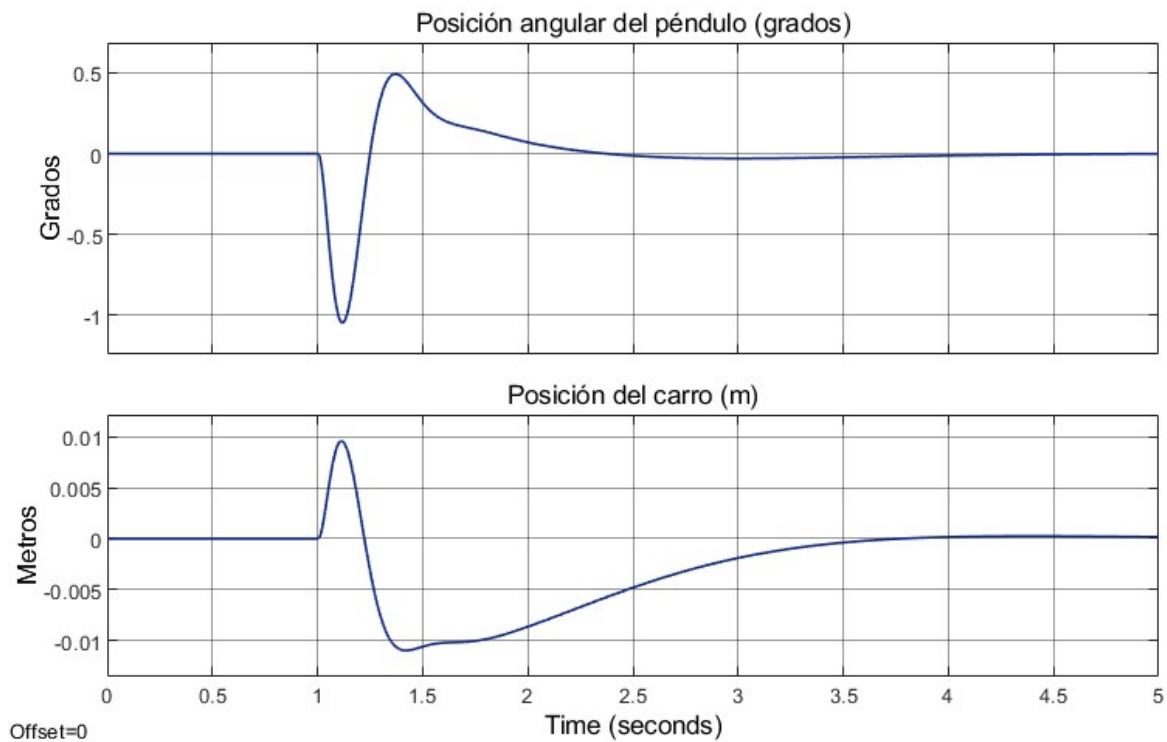


Figura 6.7: Respuesta de la posición angular del péndulo y la posición lineal del carro ante una perturbación de 5 voltios en el segundo 1 con una duración de 0.01 segundos.

Como se puede observar en la figura 6.7, al ocurrir una perturbación en el segundo 1 de simulación el péndulo tiende a desestabilizarse hasta llegar a -1 grado de inclinación, mientras que el carro se desplaza hasta 10 centímetros. Mediante el efecto de la

ley de control calculada, el sistema consigue estabilizarse prácticamente por completo a los 3 segundos desde que se da la perturbación.

En la siguiente figura se muestra la acción de control que hace estabilizarse al sistema. Se observa como dicha acción es mucho más grande que la que es capaz de soportar el actuador, llegando a saturarse a los 12 voltios (tensión típica de funcionamiento de un motor de corriente continua). Esta saturación se alcanza durante un breve periodo de tiempo. Por lo tanto, se puede decir que el sistema de control cumple con las especificaciones.

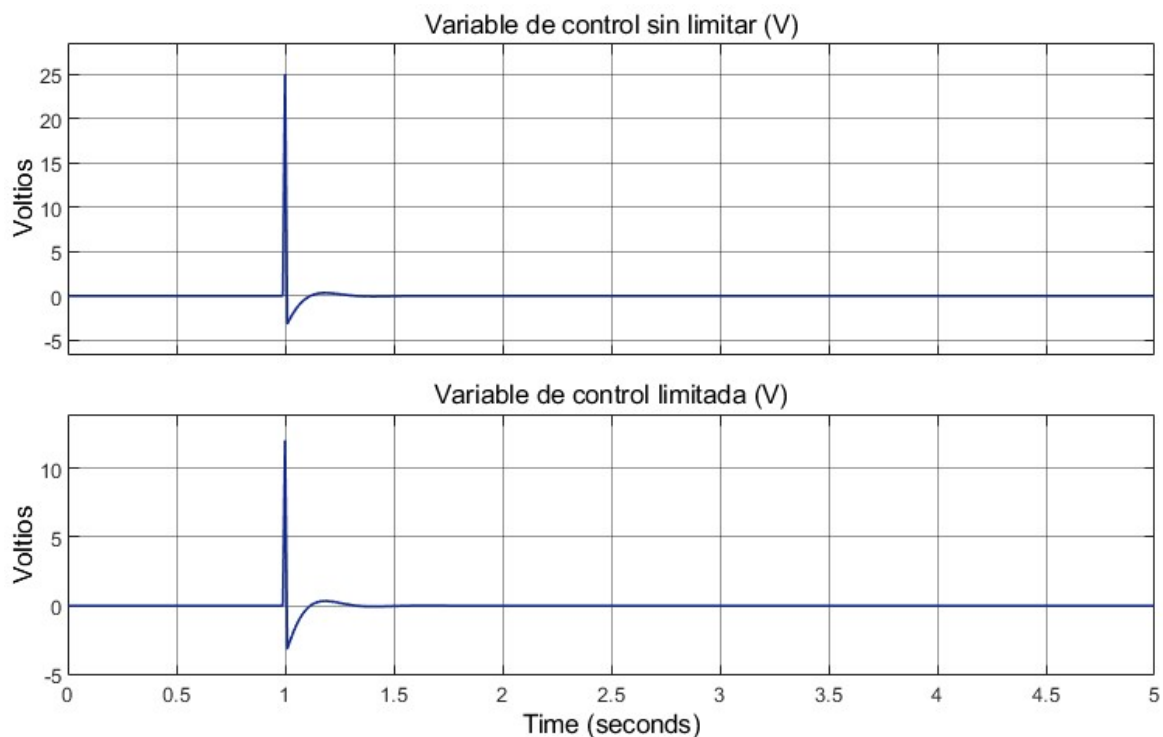


Figura 6.8: Esfuerzo de control sin efecto de saturación y con la saturación.

Para obtener una respuesta que se aproxime más a la planta real, se simula el sistema de control obtenido con el modelo no lineal.

En la siguiente figura se presenta el diagrama de simulación del modelo no lineal de la planta. El modelo no lineal del sistema péndulo-carro se ha realizado mediante bloques de operaciones matemáticas de *Simulink* para representar las ecuaciones 3.8 y 3.9.

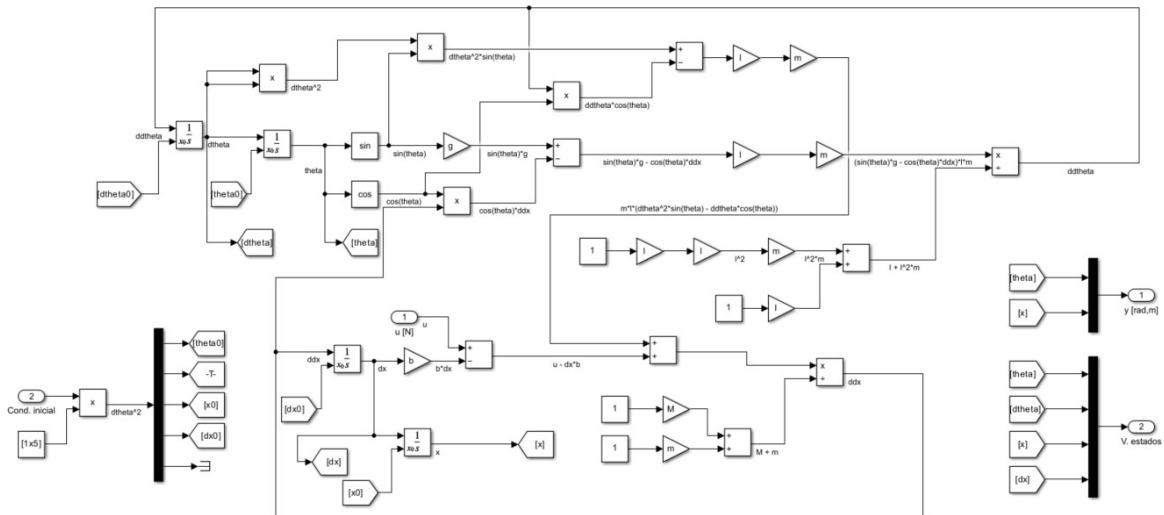


Figura 6.9: Diagrama de bloques del modelo no lineal del sistema péndulo-carro.

Para el modelo no lineal del sistema péndulo-carro se ha tenido en cuenta el momento de inercia de la planta. Este dato no se tenía caracterizado por lo que se ha utilizado el mismo que utilizan los autores del trabajo [3]. El valor del momento de inercia utilizado es $0,0136 \text{ Kg/m}^2$.

Se ha añadido el modelo del motor de corriente continua con su función de transferencia (Ecuación 3.37) cuya salida se divide por el radio para obtener la señal de entrada del péndulo en Newtons. Por otro lado, se ha ampliado el vector de estados con la corriente consumida por el motor que está directamente relacionada con el torque.

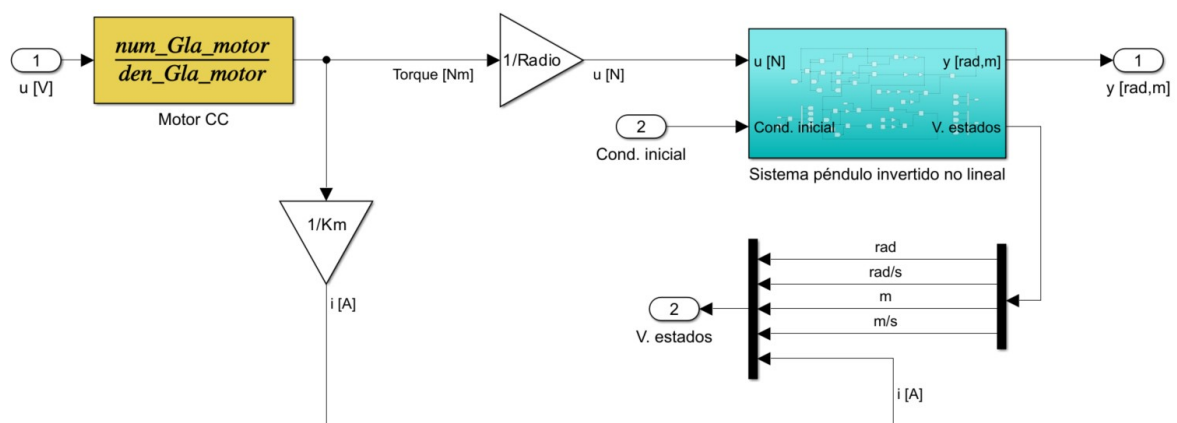


Figura 6.10: Diagrama de bloques del modelo no lineal del sistema péndulo-carro con actuador.

El radio por el que se divide el torque es el de la rueda dentada que se encuentra acoplada al eje del motor; este radio es de 0.005 metros.

En la siguiente figura se muestra el diagrama de simulación con el cual se comparan y analizan las respuestas del modelo no lineal y el modelo linealizado.

Se puede observar como ambos bloques de sistemas de control son exactamente iguales, exceptuando el modelo de la planta utilizado en cada sistema. Además, la señal de perturbación es la misma para ambos sistemas.

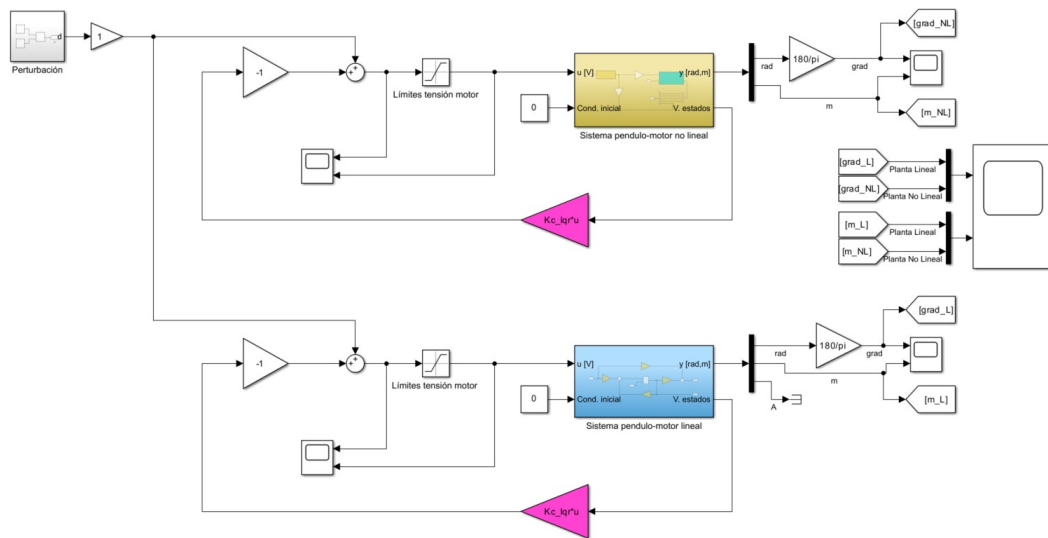


Figura 6.11: Diagrama de simulación creado para comparar la respuesta del modelo lineal con la del modelo no lineal.

Tras una perturbación de tipo impulso de 5 voltios en la señal de control, y simular el sistema con un paso de integración de 0.01 segundos en tiempo continuo, con una duración de 5 segundos, las respuestas obtenidas son las que se muestran en la siguiente figura.

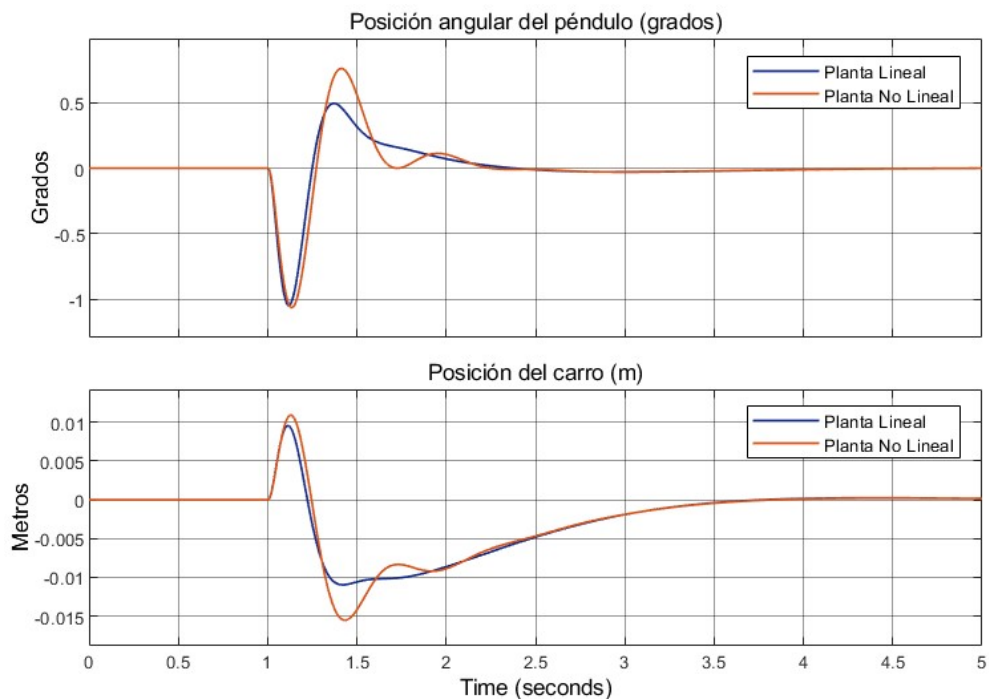


Figura 6.12: Respuesta del sistema de control con el modelo no lineal y modelo linealizado ante un impulso de 5 voltios en la señal de control.

En las respuestas se observa como ambos modelos tienen comportamientos muy parecidos. Estas respuestas empiezan a tener mayor diferencia cuanto más se alejan del punto de operación para el cual fue linealizada la planta.

La diferencia más destacable en los resultados es el overshoot que es algo superior en el modelo no lineal. Sin embargo, ambos sistemas tienden a estabilizarse prácticamente en el mismo periodo de tiempo. Por lo tanto, se puede decir que se ha obtenido un sistema de control válido para la planta en cuestión.

A continuación, para realizar un análisis de la robustez [16] que presenta el sistema, se va a ver qué sucede si esta perturbación perdurase en el tiempo. Para ello, se va a realizar la simulación con un tiempo de muestreo muy pequeño con el objetivo de poder ver el instante exacto en el que el sistema no lineal pierde la estabilidad. Para buscar este instante se va incrementando la duración de la perturbación hasta llegar a mantener ciertas oscilaciones en el sistema.

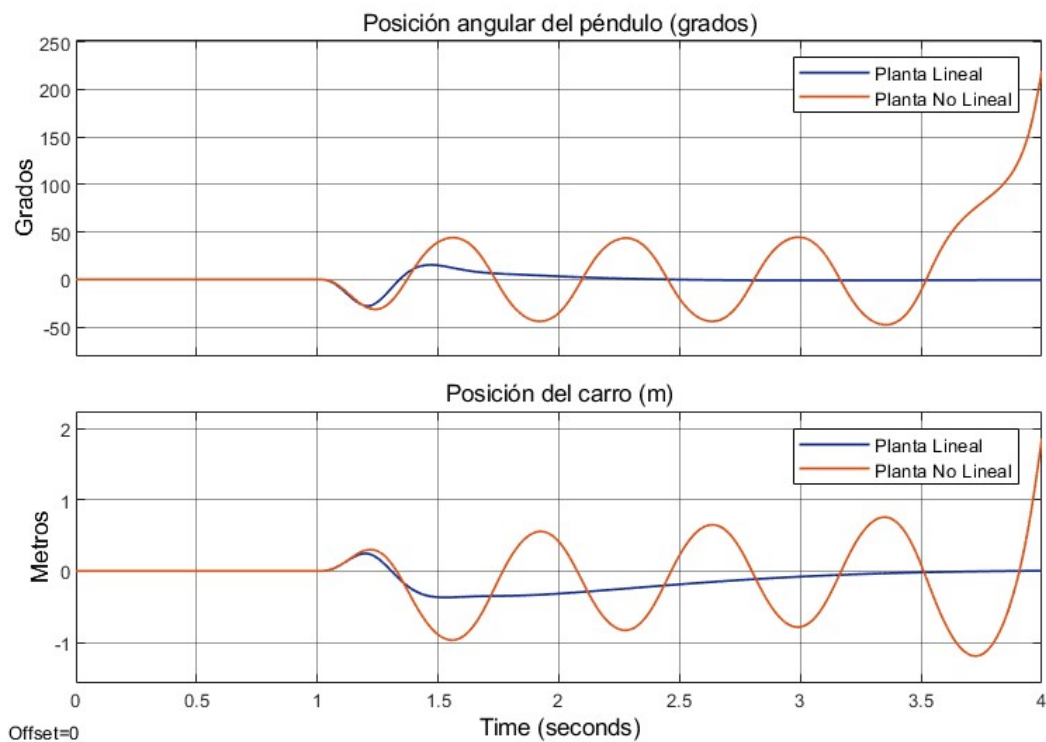


Figura 6.13: Respuesta del sistema de control con el modelo no lineal y modelo linealizado ante una señal de duración determinada que hace al sistema no lineal perder la estabilidad.

Finalmente, se ha encontrado que el modelo no lineal pierde su estabilidad para una perturbación de 5 voltios en la señal de control con una duración de 1.13027 segundos. La respuesta que tiene el modelo no lineal es la que se aproxima a la realidad, ya que el modelo linealizado solo tiene validez para un entorno o vecindad del punto de operación para el cual fue linealizado.

Aunque se ha obtenido un sistema de control para la planta que da buenos resultados en simulación, este no podrá ser utilizado en la práctica ya que se trata de

un control analítico basado en un modelo de la planta muy aproximado. En el caso del trabajo en cuestión, el modelo no se ha podido caracterizar con precisión ya que cuenta con múltiples incertidumbres que dificultan poder disponer de un modelo suficientemente preciso, lo cual queda fuera del alcance de este trabajo.

Otro punto a tener en cuenta, si se quisiese implementar este control, son las variables que necesitan ser medidas. Si se tuviese alguna variable del vector de estados que no puede ser medida sería necesario el uso de un observador de estados. El cálculo de un observador de estados se encuentra basado en el modelo de la planta, por lo que se tendría el mismo problema. Sin un modelo preciso de la planta no podría realizarse el control LQR.

Una alternativa para utilizar el controlador LQR sería intentar ajustar los parámetros de dicho controlador de forma experimental. Sin embargo, este es un proceso más laborioso que el ajuste experimental de los parámetros de un controlador tipo PID. Mientras que en el controlador LQR se tienen que ajustar los 5 parámetros de la matriz K_c (un parámetro por cada variable del vector de estados), en el controlador PID solo se tienen que ajustar 3 parámetros, siendo mucho más sencilla la sintonización de este último.

Capítulo 7

Configuración de la Raspberry Pi

7.1. Introducción

Uno de los motivos por el cual se seleccionó la Raspberry Pi para implementar el sistema es debido a que permite desarrollar algoritmos en Simulink y ejecutarlos de forma autónoma en la Raspberry Pi. El paquete de soporte que ofrece Matlab amplía la biblioteca de bloques de Simulink para configurar la Raspberry Pi, enviar y recibir información por UDP, leer y escribir datos procedentes de sensores, entre otras utilidades.

El uso de Simulink para la programación en la Raspberry Pi permite:

- Desarrollar y simular algoritmos en Simulink y utilizar la generación automática de código para ejecutarlos en el dispositivo.
- Incorporar rutinas de procesamiento de señales, diseño de control, lógica de estados y otras rutinas matemáticas y de ingeniería avanzada en los proyectos de programación en la Raspberry Pi.
- Ajustar y optimizar de forma interactiva los parámetros mientras se ejecuta en la Raspberry Pi.

Esta información ha sido consultada en la propia página de Matlab [10].

Pero para que todo esto sea posible, antes se debe configurar la Raspberry Pi, cargar el sistema operativo correspondiente y las librerías necesarias que permitan realizar estas tareas.

Para este proceso de configuración es necesario disponer de una tarjeta MicroSD en la cual se instalará el sistema operativo. Los pasos para realizar la instalación se indican cuando se inicia la configuración del complemento *Simulink Support Package para Raspberry Pi*. Este complemento se instala desde el administrador de complementos de Matlab.

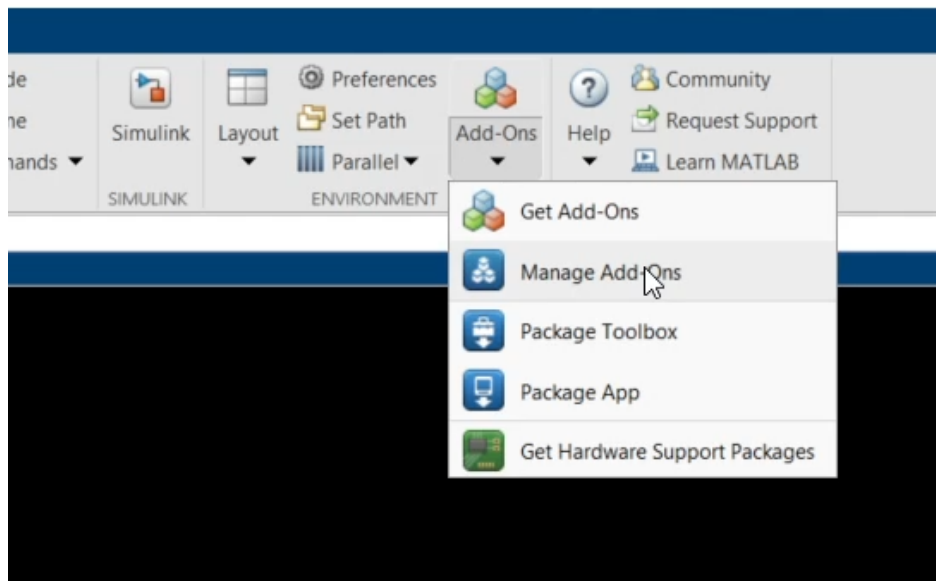


Figura 7.1: Administrador de complementos de Matlab.

Tras seguir los pasos que se indican en la propia configuración del complemento, se tendrá instalado el sistema operativo y las librerías necesarias en la Raspberry Pi.

Una vez se tiene el sistema operativo instalado en la Raspberry Pi, es conveniente realizar unos ajustes antes de trabajar con ella.

Para realizar esta configuración se debe abrir una comunicación por SSH entre el PC y la Raspberry. Esta comunicación se puede abrir con el software *'PuTTY'* o directamente con Matlab, como se va a explicar a continuación.

Para comenzar el proceso de configuración se debe tener la Raspberry Pi conectada a la alimentación y a la red según se haya configurado.

Tras encender la Raspberry Pi se debe esperar un tiempo a que se inicie. Pasados unos minutos se introduce el siguiente comando en la ventana de comandos de Matlab:

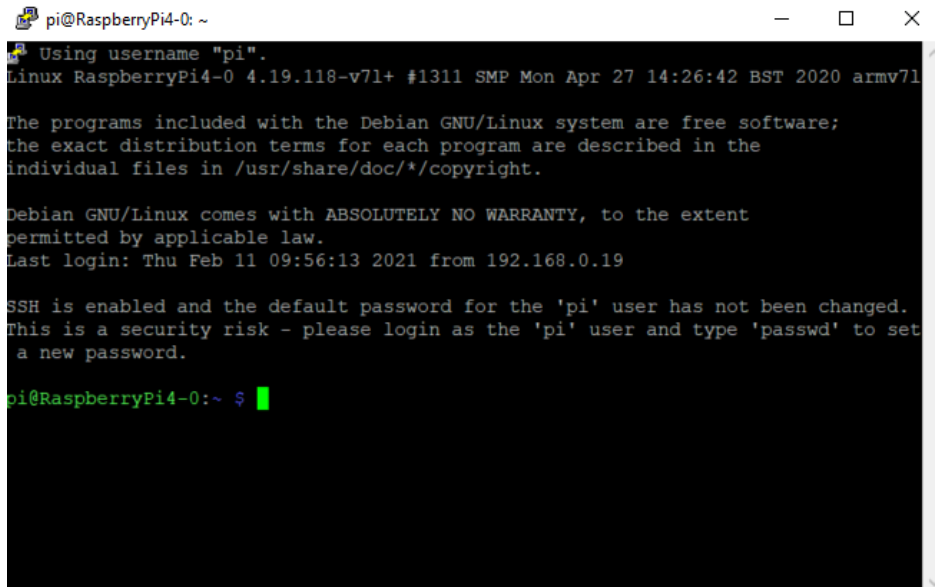
```
rpi = raspberrypi('IP_de_la_Raspberry_Pi')
```

Con este comando se establece una conexión entre la Raspberry Pi y el PC.

Para abrir la consola de la Raspberry Pi, a partir de la cual se hará la configuración, se ejecuta el siguiente comando en Matlab:

```
openShell(rpi)
```

Llegados a este punto se tendrá acceso a la consola de la Raspberry Pi.



```
pi@RaspberryPi4-0: ~
Using username "pi".
Linux RaspberryPi4-0 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 11 09:56:13 2021 from 192.168.0.19

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@RaspberryPi4-0:~ $
```

Figura 7.2: Consola de comandos de la Raspberry Pi.

Las configuraciones que se van a realizar son las que se enumeran a continuación. Cabe destacar que esta configuración es la que se ha realizado para este proyecto. La configuración irá en función de las necesidades del propio proyecto que se esté desarrollando. En este caso la configuración está pensada para ahorrar el mayor número de recursos posibles y que la Raspberry Pi pueda dedicar todo su potencial a ejecutar el algoritmo que se programe desde Simulink.

1. Desactivar Wifi:

Tras ejecutar el siguiente comando en la consola de la Raspberry Pi, se pueden listar las interfaces que se tienen activas.

```
ifconfig
```

Como se podrá observar, tras la primera instalación del sistema operativo, se tendrá el Wifi activo. Por lo tanto, se va a proceder a desactivarlo ya que no va a ser necesario su uso, de este modo se ahorran recursos.

Para desactivarlo se ejecuta el siguiente comando en la consola de la Raspberry Pi:

```
sudo iwconfig wlan0 txpower off
```

2. Desactivar el entorno gráfico:

El entorno gráfico de la Raspberry Pi no será necesario ya que se va a trabajar directamente a partir de la consola de comandos. Por lo tanto, se puede desactivar para, al igual que antes, ahorrar en recursos.

Para ello se introduce el siguiente comando:

```
sudo raspi-config
```

Dentro de '*Boot Options*', y posteriormente '*Desktop / CLI*', se debe seleccionar '*Console Autologin*' para que la Raspberry Pi al iniciarse no solicite la contraseña para acceder. Esta opción mantendrá a la Raspberry Pi en la consola de comando sin abrir el entorno gráfico.

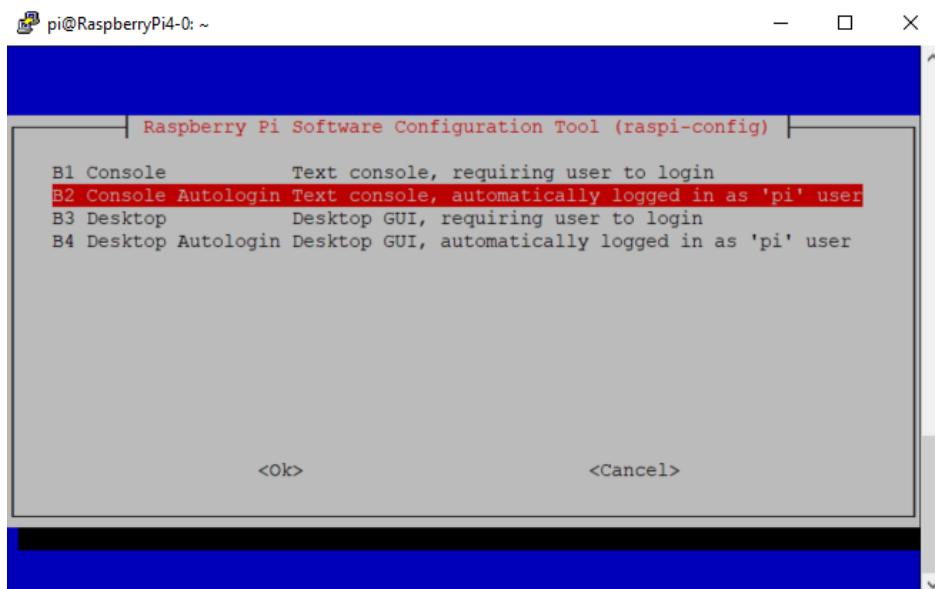


Figura 7.3: Configuración de inicio de la Raspberry Pi.

3. Protocolos de comunicación para periféricos:

En este apartado se podrán activar o desactivar, según las necesidades del proyecto, los protocolos de comunicación que se necesiten utilizar en la Raspberry Pi para comunicarse con los distintos periféricos.

Para ello se introduce el siguiente comando:

```
sudo raspi-config
```

En la opción '*Interfacing Options*' se pueden activar o desactivar los protocolos que dispone la Raspberry Pi.

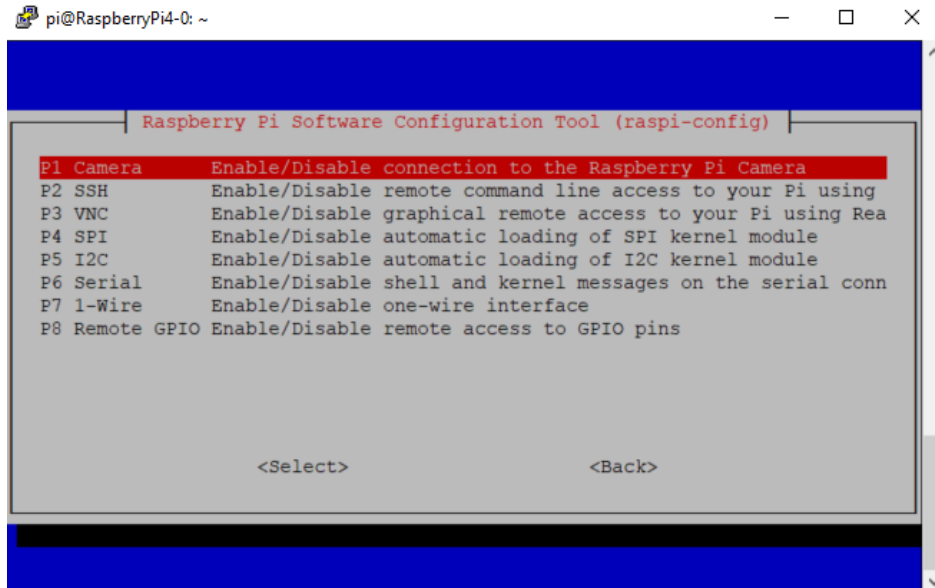


Figura 7.4: Configuración de los protocolos de comunicación de la Raspberry Pi.

En este caso los protocolos que se han dejado activos son:

- **SSH:** Para poder establecer la conexión entre la Raspberry Pi y el PC.
- **SPI:** Para la lectura del sensor analógico. En este punto se entrará en detalle, posteriormente, en la construcción del sistema real.

4. Actualización de la Raspberry Pi:

Tras instalar el sistema operativo de la Raspberry Pi es aconsejable realizar una actualización de la misma para asegurarse que cuenta con las bibliotecas más actualizadas.

Para ello se introduce el siguiente comando:

```
sudo raspi-config
```

Al entrar en la opción '*Update*' el sistema comenzará a actualizarse. Este proceso puede tardar unos minutos.

Una vez finalizado se debe reiniciar la Raspberry Pi para que los cambios surtan efecto. El comando para reiniciar la Raspberry Pi es el siguiente:

```
sudo reboot
```

7.2. Descripción de los bloques de Simulink dedicados a Raspberry Pi y utilizados en este proyecto

Al instalar el complemento *Simulink Support Package for Raspberry Pi Hardware* se incorporan a Simulink unos bloques de simulación dedicados exclusivamente a Raspberry Pi. Dichos bloques facilitan la construcción de modelos de sistemas para posteriormente ser cargados en la Raspberry Pi en formato código.

Este complemento se encuentra en continuo desarrollo, por lo que aún se siguen incorporando nuevos bloques con nuevas utilidades en este complemento.

A continuación, se van a presentar los bloques que se han utilizado en el desarrollo del presente proyecto.

7.2.1. GPIO Read

A partir de este bloque se puede realizar una lectura digital del pin GPIO que se requiera. Su configuración es simple.

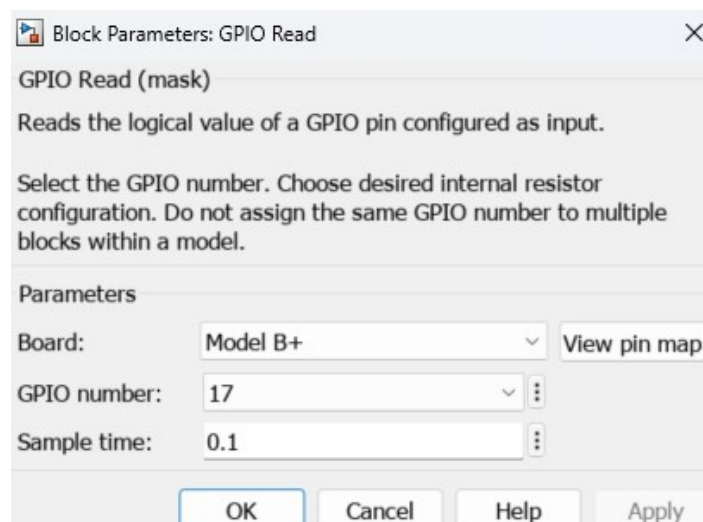


Figura 7.5: Configuración de parámetros del bloque GPIO Write.

Solo se tienen dos parámetros para ajustar. Uno de ellos es el modelo de la Raspberry Pi que se está utilizando, en nuestro caso será para la Raspberry Pi 4 model B. Otro parámetro será el pin al que va asociado este bloque.

Al igual que la mayoría de los bloques de Simulink que lo requieran, es necesario indicar el periodo de muestreo que se desee en dicho bloque. Este siempre debe ser mayor o igual al periodo de muestreo de la simulación.

En la [Figura 4.6](#) se muestra la disposición de los pines GPIO de la Raspberry Pi 4B.

Este bloque estará destinado a la lectura de unos finales de carrera que se instalarán en el sistema real para acotar el recorrido del carro.

7.2.2. Encoder

Se trata de un nuevo bloque que permite realizar directamente la lectura de un codificador incremental relativo sin necesidad de hacer el tratamiento necesario de las señales para obtener la dirección de giro.

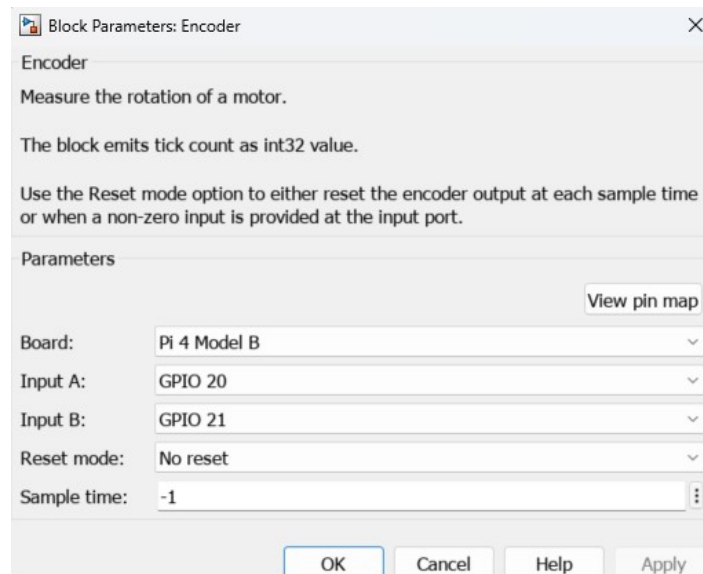


Figura 7.6: Configuración de parámetros del bloque *Encoder*

En este caso se debe seleccionar los dos pines digitales de los cuales se van a obtener las dos señales de medida del sensor en cuestión. Como en el caso del bloque *GPIO Read* se puede configurar el periodo de muestreo y el modelo de la Raspberry que se está utilizando.

El *Reset mode* permite reiniciar las medidas, si fuese necesario, a partir de una señal externa o en un instante de tiempo de simulación indicado.

Este bloque será utilizado para obtener la medida del codificador incremental relativo que se encuentra acoplado al eje del motor con el fin de medir la posición lineal del carro y la dirección de avance.

7.2.3. PWM

Como la mayoría de los sistemas de procesamiento digital, la Raspberry Pi cuenta con un cierto número de pines digitales de salida con capacidad de generar una señal PWM¹.

¹PWM - Pulse-Width Modulation

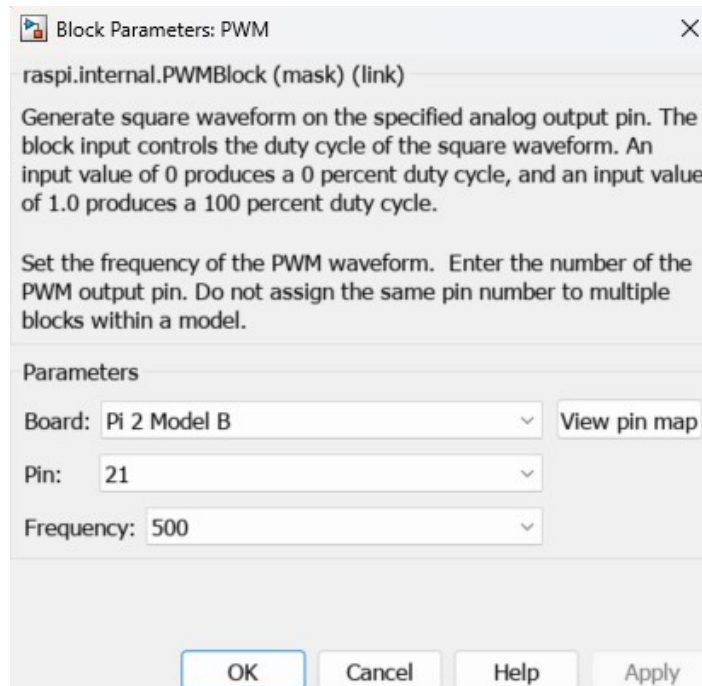


Figura 7.7: Configuración de parámetros del bloque *PWM*

El Bloque *PWM* permite generar una señal PWM a partir de una señal que oscile entre 0 y 1. El valor 1 corresponde a un ciclo de trabajo del 100%. Esta señal se extrae por uno de los pines digitales capacitados para ello.

Los parámetros que se deben configurar son: el pin por el cual se va a generar la señal PWM, el modelo de Raspberry Pi, y la frecuencia a la que se va a generar la señal PWM.

Este bloque será utilizado para generar la señal de control que actúa sobre el motor de corriente continua.

7.2.4. SPI Master Transfer

La Raspberry Pi 4B no dispone de pin de entrada analógica, por lo tanto se va a hacer uso de un conversor analógico-digital (ADC). Para realizar una primera prueba de este bloque, la señal analógica se va a obtener de un potenciómetro de $10\text{ K}\Omega$ y a partir del conversor ADC se obtendrá una señal digital la cual se leerá por el protocolo de comunicación SPI de la Raspberry Pi. Por lo tanto, la configuración que se va a presentar es la realizada para obtener la lectura del ADC.

Antes de continuar cabe destacar que el conversor utilizado *MCP3008* tiene una resolución de 10 bits con un máximo de 200 muestras por segundos a 5 voltios y 75 muestras por segundos a 2.7 voltios. Además, dispone de 8 canales.

Para que el ADC devuelva la lectura realizada se le debe enviar 3 bytes con la codificación necesaria para obtener la lectura del canal deseado:

- **1º Byte:** Byte de inicio. Byte compuesto por ceros, excepto el bit de menos peso que debe ser un uno para indicar el inicio de la lectura.
- **2º Byte:** Byte de dirección de registro. Los 4 bits de más peso corresponden a la dirección de registro de la cual se obtiene la lectura analógica. Los 4 bits restantes se componen por ceros o unos. El valor de estos bits no interfieren en la comunicación.

Las direcciones de registros vienen codificadas de la siguiente forma:

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

Figura 7.8: Codificación de las direcciones del ADC *MCP3008*

- **3º Byte:** Byte compuesto por ceros o unos. Su valor no interfiere en la comunicación.

Tras enviar al ADC esta secuencia de bytes, se recibe una secuencia del mismo tamaño donde viene codificada la lectura en los 10 bits con menos peso.

En la siguiente figura se presenta de forma más clara el modo de funcionamiento del ADC.

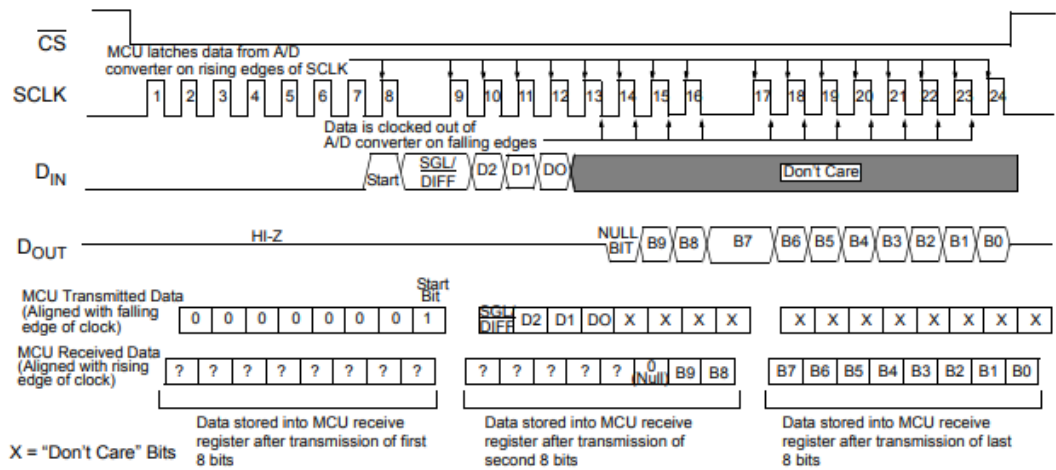


Figura 7.9: Comunicación SPI con el ADC *MCP3008*. Modo 0

Por lo tanto, teniendo en cuenta el funcionamiento del ADC mostrado en la [Figura 7.9](#), para que el bloque *SPI Master Transfer* devuelva la lectura del ADC se le debe mandar la codificación que se observa en la siguiente figura.

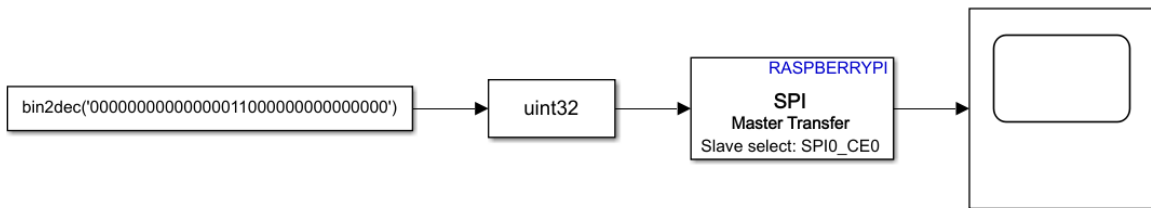


Figura 7.10: Diagrama de simulación para testear el bloque *SPI Master Transfer*

Ya que no es posible enviar una secuencia de 24 bits, se añaden 8 bits nulos a la izquierda para tener un total de 32 bits.

El tamaño que alcanza el bit menos significativo (LSB Size) se obtiene a partir de la siguiente ecuación:

$$LSB\ Size = \frac{V_{REF}}{1024} \tag{7.1}$$

El potenciómetro se encuentra alimentado a 3,3 *voltios*. Por lo tanto, el tamaño del LSB será de 0,0032 *voltios* aproximadamente.

En la siguiente tabla se muestran las mediciones obtenidas del potenciómetro a partir de un voltímetro, y las mediciones realizadas con Simulink a partir del ADC MCP3008.

Medidas	Voltímetro [V]	Simulink [V]
MIN	0	0
1	0.013	0.013
2	0.505	0.501
3	1.008	0.913
4	1.502	1.471
5	2	1.952
6	2.503	2.471
7	3.003	3.001
MAX	3.296	3.297

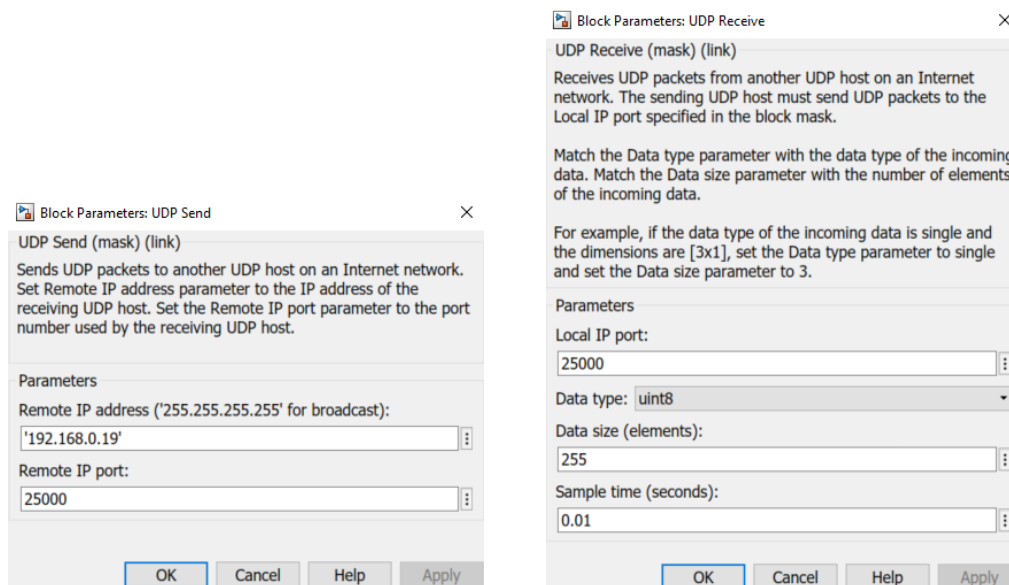
Tabla 7.1: Comparación de las medidas de tensión tomadas por un voltímetro y por el ADC

7.2.5. UDP Receive y UDP Send

La comunicación UDP es un tipo de protocolo de comunicación en red. A diferencia del protocolo de comunicación TCP, el UDP resulta ser una alternativa más simple y más rápida, aunque menos segura. Esto se debe, en general, a que la comunicación UDP no verifica la recepción de los paquetes enviados.

Para realizar la comunicación UDP es necesario que los dispositivos estén conectados por cable ethernet, ya sea directamente o a partir de un *Switch*.

En la siguiente figura se muestran los parámetros de configuración de ambos bloques.



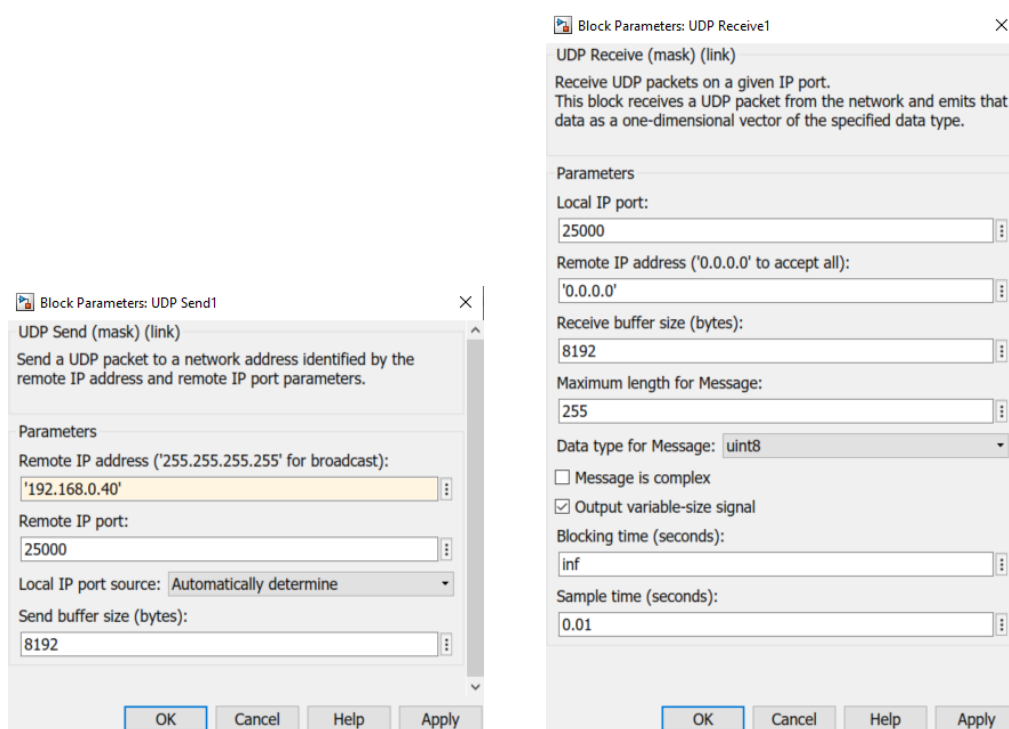
(a) Configuración bloque *UDP Send* (b) Configuración bloque *UDP Receive*

Figura 7.11: Bloques utilizados en la Raspberry Pi para la comunicación por protocolo UDP

- **UDP Send:** En el bloque *UDP Send* tan solo es necesario declarar la dirección de IP a la cual va dirigido el mensaje, y el puerto por el cual se va a emitir.
- **UDP Receive:** Para configurar el bloque *UDP Receive* se introduce el puerto por el cual se va a recibir el mensaje, el tipo de dato, el número de señales que se va a recibir y, por último, la tasa de refresco.

Estos bloques son los utilizados para realizar una comunicación UDP entre dos Raspberry's. En el caso de que se desee abrir una comunicación UDP con un PC, se deberá utilizar el bloque *UDP Receive (Host)* para recibir datos en el PC y el bloque *UDP Send (Host)* para enviar datos desde el PC. Ambos bloques se encuentran en la librería *Embedded Coder*. Solo se utilizarán estos dos bloques en el diagrama de simulación que se ejecute en el PC. En las Raspberry's se seguirán utilizando los bloques destinados a ellas.

En la siguiente figura se muestran los parámetros de configuración de los bloques destinados al PC.



(a) Configuración bloque *UDP Send (Host)* (b) Configuración bloque *UDP Receive (Host)*

Figura 7.12: Bloques utilizados en el Host para la comunicación por protocolo UDP

- **UDP Send (Host):** Al igual que el bloque *UDP Send*, utilizado para las Raspberry's, este bloque presenta la misma configuración de parámetros. La única variación que presenta es la ventaja de poder elegir el tamaño de buffer, es decir, el tamaño de memoria que almacenará los datos de forma temporal.

- **UDP Receive (Host):** Las variaciones que presenta este bloque con el destinado a la Raspberry Pi, son:
 - a) **Receive buffer size:** Al igual que en el bloque anterior se puede ajustar el tamaño de buffer que se desea tener.
 - b) **Message is complex:** Selecciona si los datos que se van a recibir son de tipo complejo o real.
 - c) **Output variable-size signal:** Esta opción permite seleccionar si se va a recibir datos de diferente longitud (marcar casilla), o por el contrario los datos recibidos tendrán todos la misma longitud (desmarcar casilla).
 - d) **Blocking time:** Ofrece la opción de ajustar el tiempo de espera máximo entre muestras. En el caso de llegar al tiempo establecido la simulación finaliza. Por defecto este valor se encuentra establecido en infinito.

7.3. Configuración del modelo para ejecutarlo en la Raspberry Pi

Una vez obtenido el diagrama de simulación que se desea cargar en la Raspberry Pi, es necesario configurar el modelo para que se ejecute en el hardware. Para ello es necesario dirigirse a *Model Settings/Hardware Implementation*.

Los parámetros a configurar son los siguientes:

- **Hardware board:** Se selecciona Raspberry Pi.

A continuación, en la misma ventana, en *Target hardware resources/Board Parameters* se configura,

- **Device Address:** IP del dispositivo en el que se va a cargar el modelo.
- **Username:** pi (por defecto)
- **Password:** raspberry (por defecto)

En la misma ventana, en *Build options* se pueden realizar las siguientes acciones:

- **Build action:** Selecciona si se desea simplemente cargar el modelo en el dispositivo, o cargarlo y ejecutarlo. Si solo se carga se puede ejecutar posteriormente usando el siguiente comando en el *Command Windows* de Matlab.

```
raspberrypi.runModel('modelName')
```

Si por el contrario se desea detener la ejecución del modelo:

```
raspberrypi.stopModel('modelName')
```

- **Build directory:** Selecciona la ruta en la que se va a guardar el modelo.
- **Run on boot:** Si se marca esta opción el modelo se ejecutará automáticamente al iniciar la Raspberry Pi.

Una vez configurado el modelo se puede proceder a cargarlo en la Raspberry Pi. Al realizar la configuración anterior aparecerá automáticamente en la ventana de Simulink una pestaña nueva llamada *Hardware*. En esta pestaña se tiene la opción de cargar el modelo en la Raspberry Pi, cargarlo y además ejecutarlo, o monitorizar la simulación y realizar ajustes de forma remota.

7.4. Comandos útiles de Matlab para trabajar con la Raspberry Pi

- Crea una conexión entre el PC y la Raspberry Pi en el caso de que no se tenga usuario y contraseña definidas.

```
rpi = raspi('192.168.0.4X', 'pi', 'rasberry')
```

- Crea una conexión entre el PC y la Raspberry Pi si el usuario y la contraseña están predefinidas.

```
rpi = raspberrypi('192.168.0.4X')
```

- Apaga la Raspberry Pi que se asigne.

```
raspberrypi('192.168.0.4X').execute('sudo _shutdown_-h_now')
```

- Reinicia la Raspberry Pi que se asigne.

```
raspberrypi('192.168.0.4X').execute('sudo _reboot')
```

- Borra el programa que se ha cargado en la memoria de arranque de la Raspberry Pi.

```
removeRunOnBoot(rpi)
```

- Ejecuta un modelo ya precargado en la Raspberry Pi.

```
raspberrypi.runModel('modelName')
```

- Detiene la ejecución del modelo que se tiene en la Raspberry Pi.

```
raspberrypi.stopModel('modelName')
```

- Carga un modelo en la Raspberry Pi el cual se ejecutará automáticamente cada vez que se inicie la Raspberry Pi.

```
addToRunOnBoot(rpi, 'model_name')
```

- Devuelve el nombre del modelo que se está ejecutando en la Raspberry Pi.

```
getRunOnBoot(rpi)
```

Capítulo 8

Construcción del sistema real

8.1. Introducción

En este capítulo se va a presentar el proceso de construcción del sistema péndulo invertido con el motor de corriente continua como actuador. Además, se realizará la calibración y ajuste de los sensores dimensionados, así como los primeros ensayos que verifiquen el correcto funcionamiento de las distintas partes que componen el sistema, tanto electrónica como mecánica.

Cabe destacar que hasta el momento no ha sido dimensionado el motor de corriente continua debido a que se desconocían los valores máximos aproximados de velocidad angular y par motor que este debe tener. Este punto también será desarrollado en este capítulo a partir de los diagramas de simulación obtenidos en el [Capítulo 6](#). Estas simulaciones fueron realizadas con los parámetros físicos caracterizados de la planta real, por lo tanto, los resultados podrán ser de ayuda para aproximar las características del actuador.

El objetivo final de este capítulo será tener el sistema péndulo invertido construido y ajustado para comenzar, en un posterior capítulo, con el ajuste experimental del controlador.

8.2. Dimensionamiento del motor de corriente continua

Para dimensionar el motor de corriente continua se va a hacer uso de los resultados del sistema de control LQR que se obtuvieron en el [Capítulo 6](#). Para ello, se va a medir la velocidad angular a la cual gira el motor y par máximo que este produce en simulación para estabilizar la planta ante una perturbación.

Partiendo del diagrama de bloques de la [Figura 6.10](#) se puede obtener el par motor midiendo la señal de salida de la función de transferencia *Motor CC*, y las revoluciones por minuto a partir de la variable de estado que indica la velocidad lineal de carro. Esta variable se encuentra en metros por segundos, por lo que a partir de las siguientes expresiones se pueden obtener las revoluciones por minuto del motor.

Se sabe que la velocidad angular está relacionada con la velocidad lineal a partir de la siguiente expresión:

$$w \text{ [rad/s]} = \frac{V \text{ [m/s]}}{\text{Radio [m]}} \quad (8.1)$$

donde el radio, al igual que en expresiones anteriores, corresponde al radio de la rueda dentada que se encuentra acoplada al eje del motor y que transfiere el movimiento angular al carro.

Por otro lado, la velocidad angular se puede pasar a revoluciones por minuto a partir de la siguiente relación:

$$1 \text{ rpm} = \frac{2 \cdot \pi}{60} [\text{rad/s}] \quad (8.2)$$

Estas dos expresiones dan como resultado la siguiente relación, que será utilizada para obtener las revoluciones por minuto del motor.

$$1 \text{ rpm} = \frac{2 \cdot \pi \cdot \text{Radio}}{60} [m/s] \quad (8.3)$$

En la siguiente figura se muestran, para una simulación de 5 segundos con una perturbación en el segundo 1, los resultados de par motor y revoluciones por minuto que presenta el sistema.

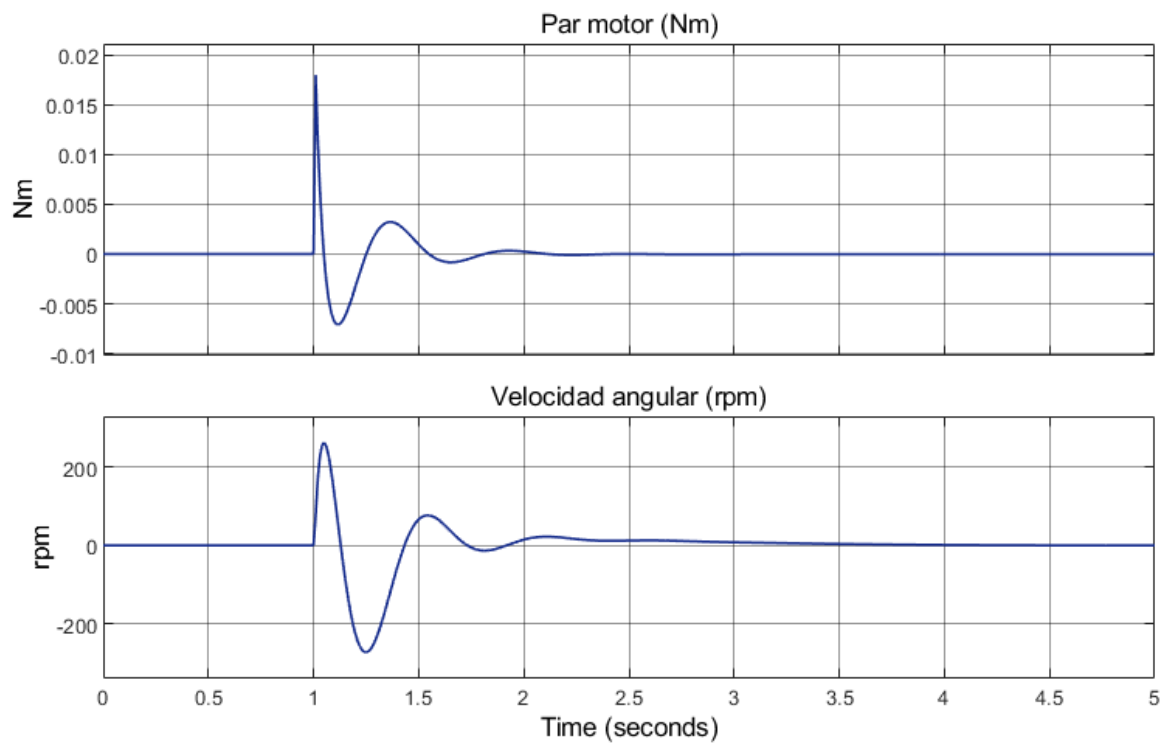


Figura 8.1: Resultados de revoluciones por minuto y par motor para una simulación de 5 segundos con una perturbación en el segundo 1

En la simulación anterior el motor ha llegado a pasar las 200 revoluciones por minuto y el par ejercido ha estado cerca de 0.2 Nm. Estos son solo valores aproximados y no se puede dimensionar el motor con valores límites próximos a los obtenidos.

Finalmente, se ha seleccionado un motor de la marca *Seed Studio*, en concreto el modelo *JGA25-37000*. Este motor presenta unas características muy superiores a

las estimadas. Esto se ha realizado a propósito para no encontrar una limitación en el actuador cuando se esté realizando el ajuste del controlador experimental. De igual modo, no se ha querido sobredimensionar demasiado en velocidad ya que ello podría producir una respuesta inadecuada del motor a velocidades demasiado bajas.



Figura 8.2: Motor de corriente continua del fabricante *Seed Studio* de 12 voltios y 1000 rpm (Modelo: JGA25-370)

El motor cuenta con una relación de reducción 1:10, capaz de alcanzar las 1000 revoluciones por minuto a 12 voltios y un par motor máximo de 0.73 Nm.

8.3. Montaje estructural

Para el montaje estructural se han utilizado las piezas impresas en 3D con tecnología FDM. Todas ellas son de diseño propio y solo algunas se muestran en el [Capítulo 4](#) ya que el presente trabajo no se centra en el diseño estructural de la planta.

Las varillas lineales utilizadas son de 12 milímetros de acero inoxidable para obtener suficiente rigidez en el desplazamiento lineal del carro y evitar las posibles vibraciones.

El carro se desplaza por las varillas gracias a dos rodamientos lineales como el que se muestra en la [Figura 4.3](#).

En la figura que se muestra a continuación se puede observar como queda todo el conjunto de piezas ensambladas.



Figura 8.3: Imagen del sistema real construido

A continuación, una vista del sistema real con el péndulo ubicado en la posición vertical. Punto para el cual se pretende estabilizar.



Figura 8.4: Imagen del sistema real construido con el péndulo en su posición vertical

El motor, junto con el sensor de desplazamiento lineal y la rueda dentada, se encuentran alojados en el interior de una de las piezas laterales que soportan las varillas.

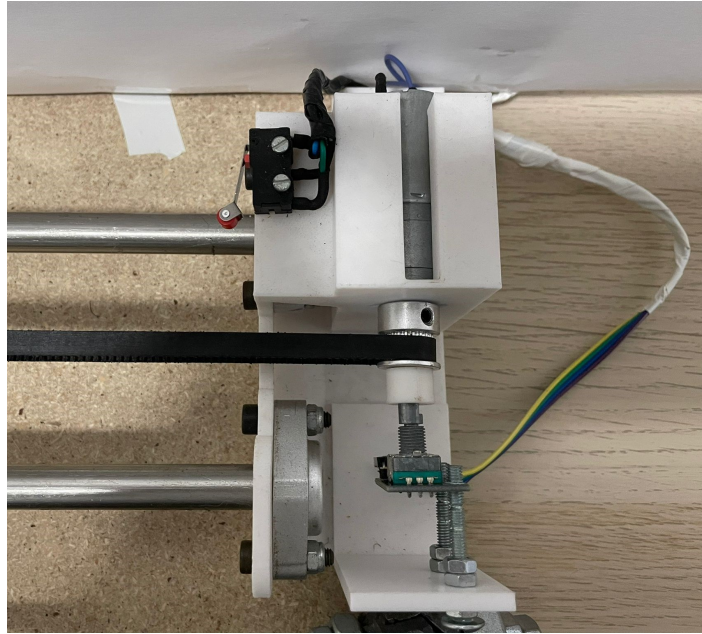


Figura 8.5: Ubicación del motor y sensor de desplazamiento lineal

El sensor de posición angular se encuentra ubicado en el carro como se puede observar en la siguiente figura.

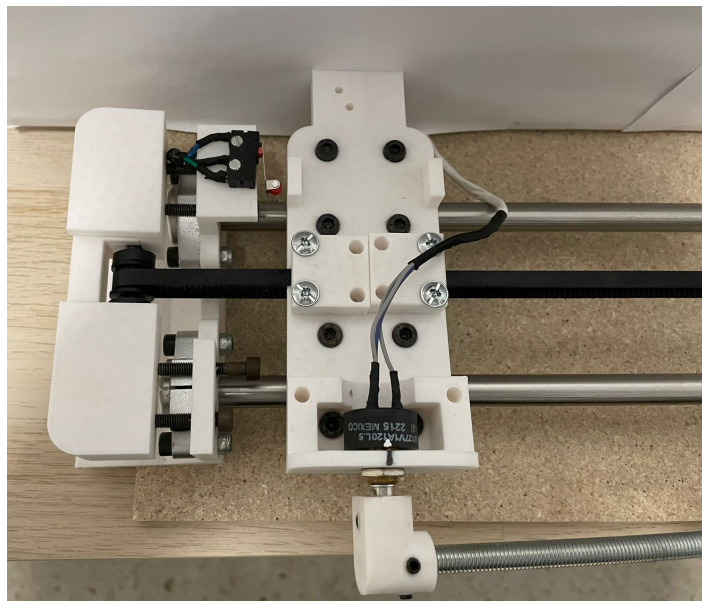


Figura 8.6: Ubicación del sensor de posición angular

En las figuras 8.5 y 8.6 se puede observar como el sistema cuenta con dos finales de carrera en cada extremo del recorrido. El objetivo de estos es detener el desplazamiento del carro en caso de colisión con alguno de ellos. De esta forma se protege al sistema para que no sufra algún daño durante el ajuste del controlador experimental o calibración de sensores.

8.4. Montaje del sistema electrónico

La Raspberry Pi es la encargada de procesar la medida de los sensores y enviar una señal de control al actuador para lograr estabilizar la planta. Es por ello que tanto los sensores como el actuador deben ir conectados a la Raspberry Pi.

- **Sensor de posición lineal:** Este sensor dispone de dos salidas digitales, por lo tanto, puede conectarse directamente a uno de los pines de la Raspberry Pi que admita lectura digital.
- **Finales de carrera:** Al igual que el sensor de posición lineal, se trata de un sensor con salida digital. Su funcionamiento es análogo al de un interruptor.
- **Sensor de posición angular:** A diferencia de los sensores anteriores, la salida de este sensor es analógica. Funciona a una tensión de alimentación comprendida entre 4.5 y 5.5 voltios, y una tensión de salida, en función del ángulo de giro, comprendida entre 0.2 y 4.8 voltios.

Debido a que la Raspberry Pi no cuenta con un ADC integrado, será necesario instalar uno para convertir el valor analógico proporcionado por el sensor a valores digitales.

El sensor utilizado tiene un recorrido total útil de 120 grados con una resolución de 0.03 grados lo que hace un total de 4000 muestras. Por lo tanto, el ADC que se instale debe contar con una resolución de 12 bits.

El ADC que se va a utilizar es el *MCP3201* del fabricante *Microchip*, el cual cuenta con una resolución de 12 bits y un protocolo de comunicación SPI. Este protocolo será el utilizado para establecer la comunicación con la Raspberry Pi y poder tomar las medidas del sensor en cuestión.

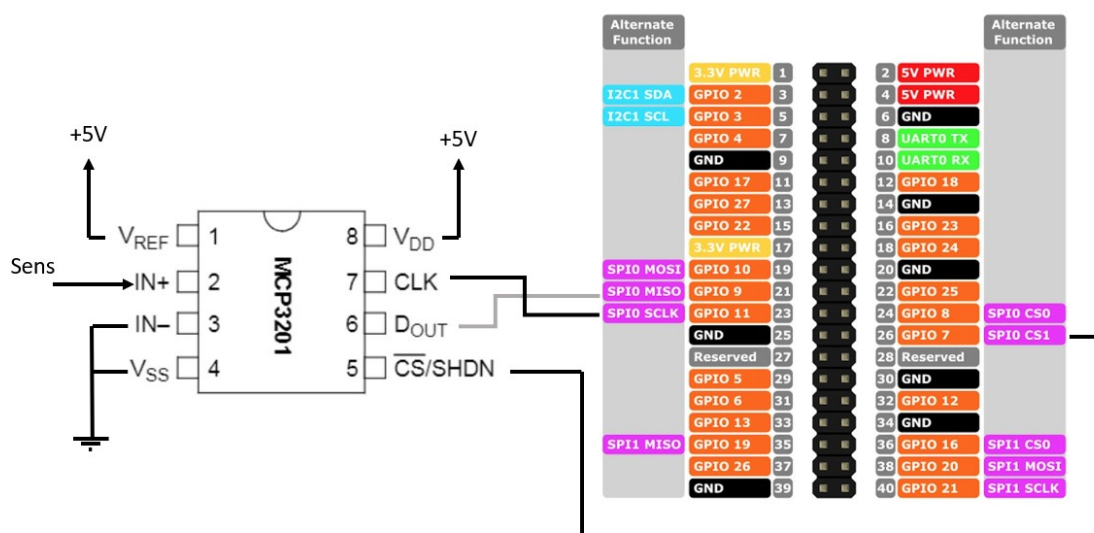


Figura 8.7: Conexión entre el ADC *MCP3201* y la Raspberry Pi 4B

- **Motor de corriente continua:** El motor de corriente continua será controlado

por la Raspberry Pi, por lo tanto, será el único componente que utilizará uno de los pines digitales de la Raspberry Pi como salida.

Un motor de corriente continua varía su par en función de la tensión con la cual se está alimentando. Una forma de variar la tensión media suministrada por la Raspberry Pi al motor es a partir de una señal PWM. Sin embargo, la Raspberry Pi no es capaz de suministrar la potencia necesaria para actuar sobre el motor de corriente continua. Es por ello que será necesario utilizar un controlador de motores que a partir de una señal de PWM de entrada, genere la potencia necesaria y equivalente para que el motor funcione adecuadamente.

El controlador de motores que se va a utilizar será el *L298N* implementado ya en un circuito electrónico con otros componentes que hacen que su uso sea de aplicación directa.

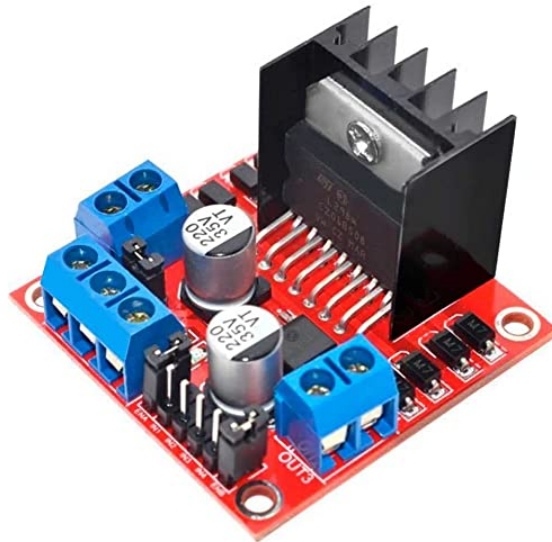


Figura 8.8: Controlador de motores *L298N*

Este dispositivo es capaz de trabajar con tensiones comprendidas entre 5 y 35 voltios, y corriente máxima de 4 amperios. Valores suficientes para que el motor trabaje correctamente. Además, aunque no será necesario en este proyecto, este dispositivo es capaz de actuar sobre dos motores.

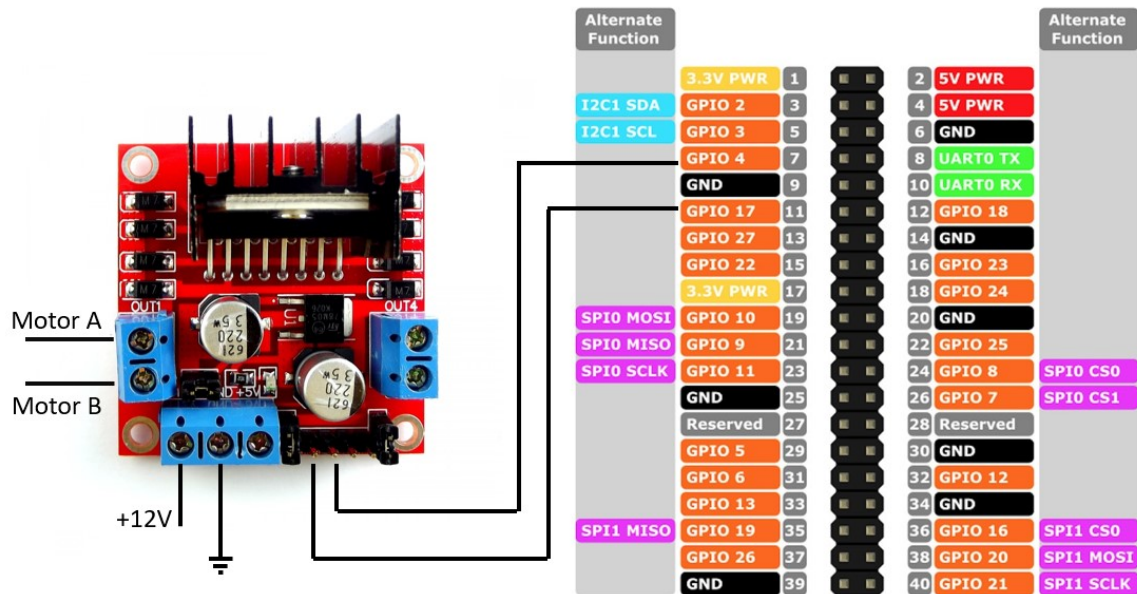


Figura 8.9: Conexión entre el controlador de motor *L298N* y la Raspberry Pi 4B

Como se puede observar en la [Figura 8.9](#) los dos cables del motor (Motor A y Motor B) se encuentran conectados al controlador de motores, el cual es capaz de suministrar la potencia que este necesita para su funcionamiento. La tensión suministrada al motor es controlada mediante PWM por los pines de salida de la Raspberry Pi.

La tensión de alimentación del motor es de 12 voltios como se indicó en su dimensionamiento.

8.5. Calibración de los sensores

Antes de sintonizar el control experimental del péndulo se va a realizar una serie de pruebas y simulaciones con el objetivo de calibrar los sensores y ajustar la posición cero del sensor de posición angular del péndulo.

8.5.1. Sensor de posición lineal

El procedimiento llevado a cabo para realizar la calibración del encoder incremental es el que se describe a continuación.

Se desplaza el péndulo hasta una posición en la que se pueda medir la distancia desplazada con respecto a un punto fijo, en este caso respecto a una de las piezas estructurales laterales.

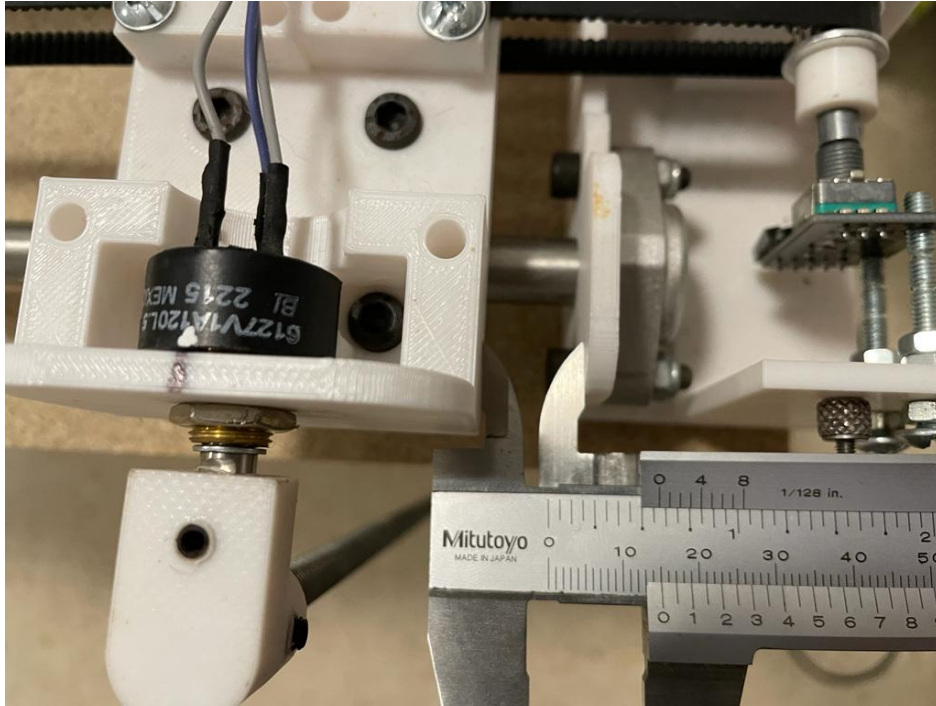


Figura 8.10: Medida inicial de la distancia entre el carro y una pieza estructural fija.

Se toma la medida entre el carro y la pieza estructural con la ayuda de un calibre como se muestra en la [Figura 8.10](#). Como se puede observar, la primera medida que se toma son 15 milímetros.

El sensor utilizado tiene una resolución de 20 pasos por vuelta, por lo que para obtener el valor del desplazamiento del carro se multiplican los pasos por el perímetro de la rueda dentada y se divide por 20 (pasos por revolución del sensor utilizado).

$$Desplazamiento [m] = \frac{2 \cdot \pi \cdot Radio [m]}{20 [pasos/rev]} \cdot Lectura [pasos] \quad (8.4)$$

El valor *Lectura* en la [Ecuación 8.4](#) corresponde al número de pasos que lleva recorrido el sensor.

A continuación, se desplaza el carro un total de 100 milímetros, obteniéndose una medida absoluta con el calibre de 115 milímetros. Este desplazamiento es monitorizado por la Raspberry Pi y representado gráficamente en el software Matlab/Simulink.

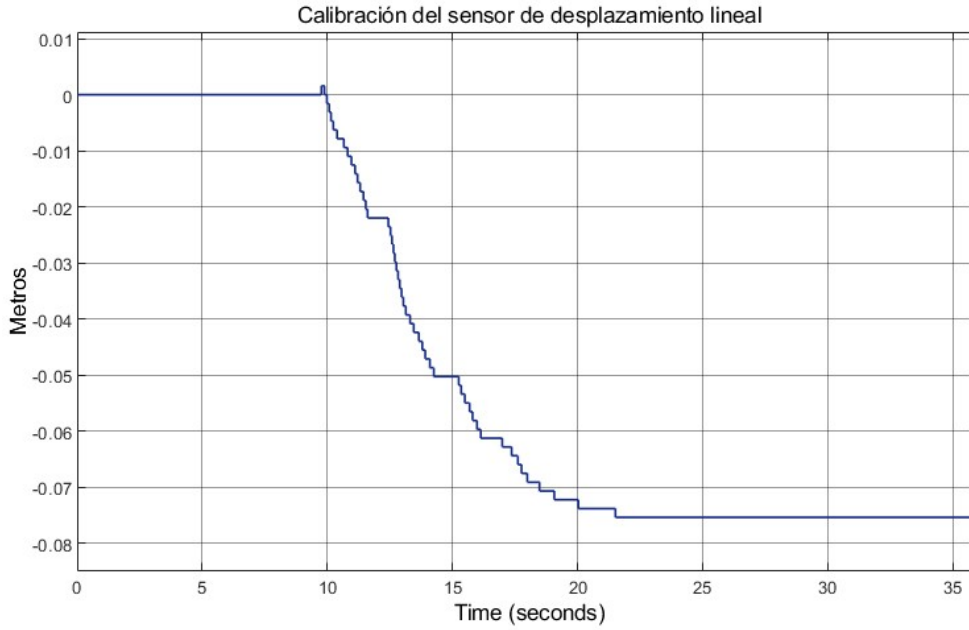


Figura 8.11: Respuesta obtenida del sensor de desplazamiento lineal ante un desplazamiento de 100 milímetros.

El valor medido es 0.077 metros, lo que equivale a 77 milímetros. Por lo tanto, el error absoluto obtenido es -0.023 metros. Para corregir este error se multiplica el valor de desplazamiento medido por el siguiente factor de corrección,

$$\frac{\text{Desplazamiento real [m]}}{\text{Desplazamiento medido [m]}} = \frac{0,1}{0,077} = 1,298 \quad (8.5)$$

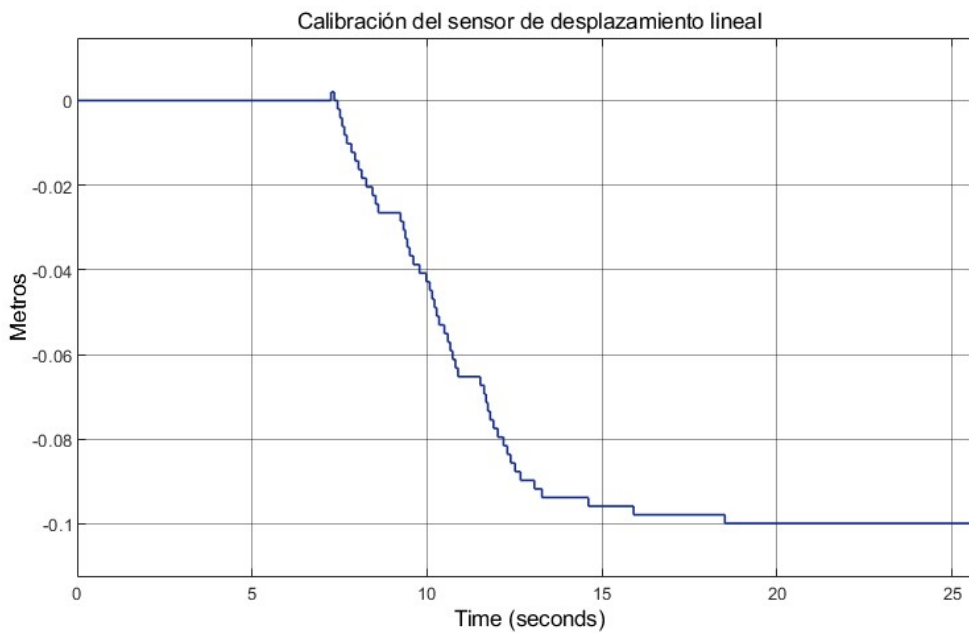


Figura 8.12: Respuesta obtenida del sensor de desplazamiento lineal ante un desplazamiento de 100 milímetros con el factor de corrección.

Tras repetir la prueba aplicando el factor de corrección se obtiene la medida de desplazamiento real. Cabe destacar que este sensor tiene una resolución de 0.0016 metros, lo que equivale a 1.6 milímetros.

El diagrama de simulación obtenido para realizar dichas pruebas es el que se muestra en la siguiente figura. Este diagrama es cargado en la Raspberry Pi, siendo esta la responsable de tomar las medidas y enviarlas al PC por protocolo TDP/IP.

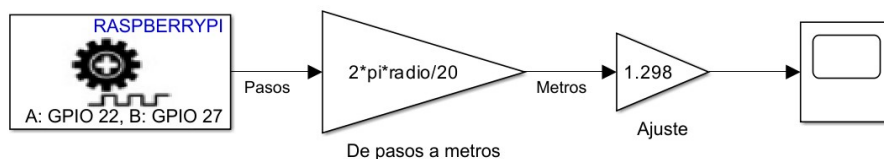


Figura 8.13: Diagrama de simulación utilizado para realizar el ajuste del sensor de desplazamiento lineal.

8.5.2. Sensor de posición angular

El sensor de posición de efecto hall (6127V1A120L.5), utilizado para medir la posición angular del péndulo, tiene una respuesta lineal con un rango de valores comprendido entre 0.2 y 4.8 voltios en función de la posición angular en la que se encuentre. El ángulo máximo que es capaz de medir es 120 grados. Por lo tanto, para tener el mismo recorrido angular en ambas direcciones se va a buscar que, cuando el péndulo se encuentre en su posición vertical, el péndulo devuelva un valor aproximado de 2.5 voltios. Con la ayuda de un voltímetro se ha posicionado el sensor buscando que, cuando el péndulo se encuentre en su posición vertical, la tensión de salida del sensor sea próxima a 2.5 voltios.

Este sensor se encuentra conectado a un ADC para convertir el valor analógico en digital. Este es conectado a la Raspberry Pi por protocolo SPI.

Cuando se inicia la lectura del sensor, el valor que se obtiene está comprendido entre 0 y 4096. Esto es así debido a que el ADC tiene una resolución de 12 bits. Por lo tanto, para obtener el valor en tensión se divide la tensión máxima de referencia del ADC (tensión máxima que es capaz de medir el sensor) por 2^{12}

Para ajustar el cero del sensor se posiciona el péndulo en su posición vertical y se toma la medida a partir del ADC. Esta medida es recibida por la Raspberry Pi por protocolo SPI y enviada al PC mediante protocolo TCP/IP. Los resultados son representados gráficamente en el PC.

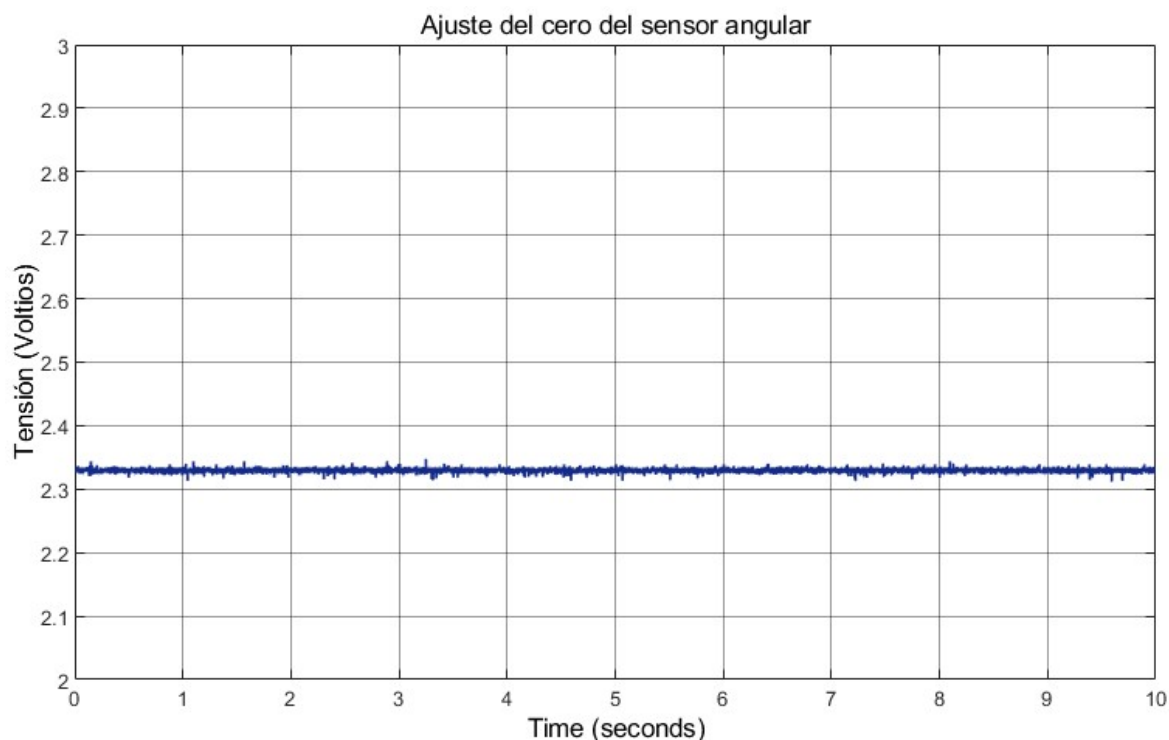


Figura 8.14: Medidas tomadas por el sensor de posición angular en un periodo de 10 segundos.

Los valores obtenidos en un periodo de 10 segundos son registrados en un array para posteriormente realizarle la media aritmética y obtener así un valor medio con el que ajustar el cero. Cabe destacar que el ruido que se aprecia en el sensor es debido al ruido que porta la tensión de referencia con la cual es alimentado el ADC y dicho sensor (5 voltios provenientes de la Raspberry Pi).

El valor obtenido de realizarle la media aritmética a esta serie de valores es 2.3293 voltios. Por consiguiente, para ajustar el cero se tendrá que restar este valor a la tensión medida por el sensor.

Para convertir la medida obtenida en voltios a grados se multiplica por la tensión de referencia (5 voltios) y se divide por el desplazamiento angular máximo que es capaz de medir el sensor (120 grados).

Por último, para realizar la calibración del sensor, se ha utilizado un nivel a 45 grados y se ha representado la medida obtenida por el péndulo cuando este se encontraba en dicha inclinación.

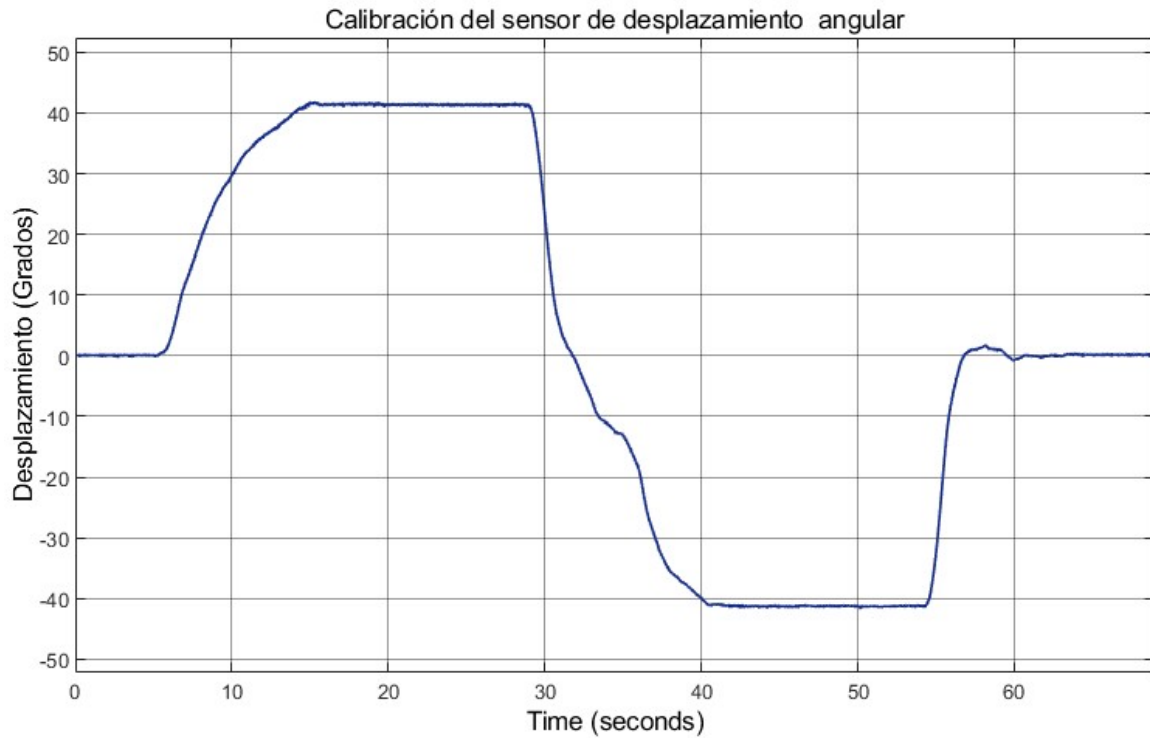


Figura 8.15: Medidas tomadas por el sensor de desplazamiento angular sin calibrar con el péndulo inclinado a ± 45 grados.

Los puntos máximos y mínimos representados en la [Figura 8.15](#) son las inclinaciones del péndulo medidas por el sensor sin calibrar. Las inclinaciones reales del péndulo eran *pm* 45 grados y las medidas son aproximadamente 41.82 grados. El error absoluto obtenido es -3.18 grados.

Para corregir este error se aplica el siguiente factor de corrección:

$$\frac{\text{Inclinacion real [grados]}}{\text{Inclinacion; medida [grados]}} = \frac{45}{41,82} = 1,076 \quad (8.6)$$

Con este factor de corrección aplicado se obtienen los siguientes resultados para la misma prueba.

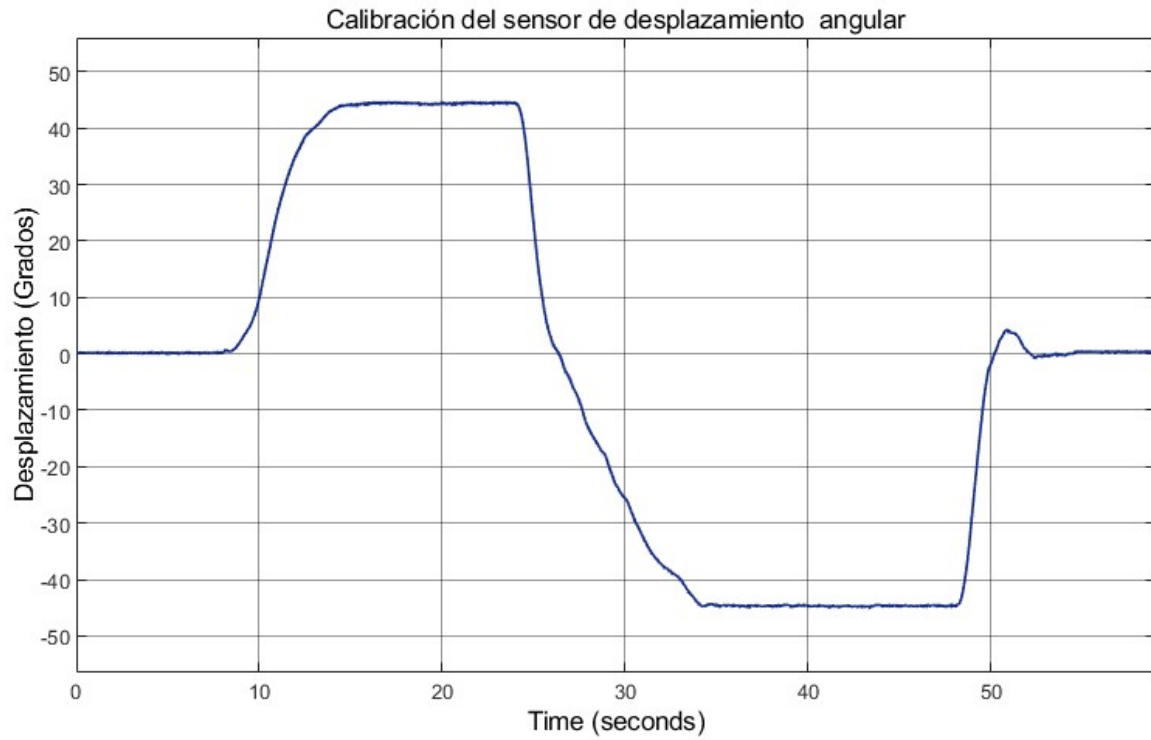


Figura 8.16: Medidas tomadas por el sensor de desplazamiento angular calibrado con el péndulo inclinado a ± 45 grados.

Capítulo 9

Pruebas experimentales con el sistema real

9.1. Introducción

Como se ha explicado en capítulos anteriores, el sistema que se ha construido presenta diversas incertidumbres que no han sido posible caracterizar y tener en cuenta para el diseño del controlador analítico. Es por ello que en la fase experimental de diseño de un sistema de control que haga estable al sistema se va a diseñar un control tipo P, PI o PID ajustando manualmente sus parámetros según la respuesta que presente el sistema.

9.2. Diagrama de control utilizado en los ensayos experimentales

Antes de continuar con los ensayos y resultados, se va a presentar el diagrama de control utilizado para la fase experimental y explicar la función de algunos de los bloques utilizados que pueden presentar mayor interés.

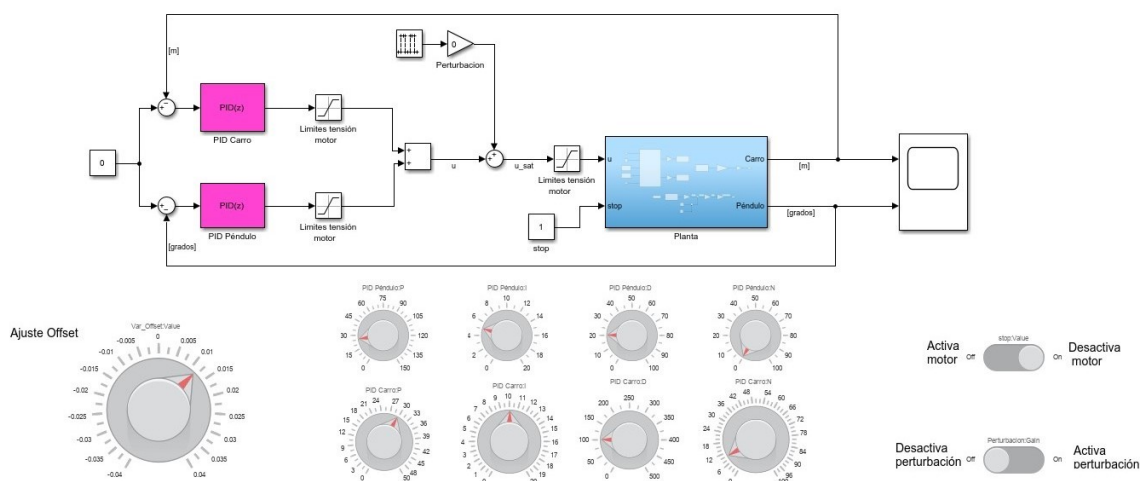


Figura 9.1: Diagrama de control utilizado en los ensayos experimentales.

En esta primera figura se observa como el diagrama de control está constituido por dos controladores PID. Estos bloques permiten ajustar manualmente los parámetros del

controlador en tiempo discreto. Cada uno de los bloques calcula una señal de control según el error medido del desplazamiento del carro o de la inclinación del péndulo. El objetivo es estabilizar el péndulo en 0 grados y el carro en 0 metros (mitad del recorrido), por lo tanto se está tratando como un problema de regulación.

Los parámetros de los controladores se pueden ajustar manualmente, mientras se está ejecutando el control en la Raspberry Pi, a partir de los potenciómetros que se muestran en la [Figura 9.1](#).

Los valores de la señal de control que se obtienen se encuentran entre ± 12 voltios. Sin embargo, estos valores no pueden ser generados por la Raspberry Pi. La señal que se tiene que generar debe estar comprendida entre 0 y 1, siendo 1 el 100 % del valor de PWM. Además, dependiendo de si la señal de control es positiva o negativa, se deberá generar la señal de PWM por una salida u otra de la Raspberry Pi para así poder controlar la dirección de giro en función del signo de la señal de control.

Para realizar lo descrito se ha desarrollado el siguiente código en el bloque de Simulink *Matlab Function*.

```

1 function [motor_der , motor_izq] = fcn(u, Vmin, end_izq , end_der , stop)
2
3 % Los finales de carrera tienen logica negada
4 % Evita que el motor funcione si hay un final de carrera pulsado y evita
5 % la zona muerta de tension que tiene el motor.
6 % Dependiendo de si la senal de control es negativa o positiva , establece
7 % la direccion de giro.
8 if (u > 0) && (end_der == 1) && (stop==0)
9     motor_der = u + Vmin;
10    motor_izq = 0;
11 elseif (u < 0) && (end_izq == 1) && (stop==0)
12    motor_izq = abs(u - Vmin);
13    motor_der = 0;
14 else
15    motor_izq = 0;
16    motor_der = 0;
17 end

```

Las salidas de la función son la dirección de giro del motor. Dependiendo del signo de la señal de control, dicha señal se generará por una salida u otra. Además, por seguridad, se han implementado unos finales de carrera en el sistema para evitar que el motor siga funcionando en el caso de que el carro llegue al límite de recorrido disponible.

Otro parámetro que se tiene en cuenta en este código es la zona muerta del motor de corriente continua. Este valor corresponde a la tensión mínima para la cual el motor empieza a moverse. Para calcularlo se ha alimentado el motor directamente a partir de una fuente de alimentación de laboratorio y se ha ido incrementando la tensión hasta que el motor ha empezado a presentar signos de movimiento.

En el bloque *Planta*, además de ejecutarse este código, se realiza la lectura de los sensores y se generan las señales PWM de salida.

9.3. Controlador utilizado en la fase experimental

Para el diseño de un control experimental se ha utilizado un controlador tipo P, PI y PID. Se ha seleccionado este tipo de controlador para la fase experimental debido a que los autores en [13] recomiendan su uso cuando el modelo matemático de la planta es desconocido y, por lo tanto, no se pueden emplear métodos analíticos. Además, cabe destacar que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID o PID modificado ([13]).

Existen tres tipos de estructuras de controlador PID: estructura ISA, paralela y serie. Para el caso que se presenta se va a utilizar la estructura paralela. Esta estructura es ampliamente utilizada en la industria debido a que se puede implementar fácilmente en forma analógica o digital.

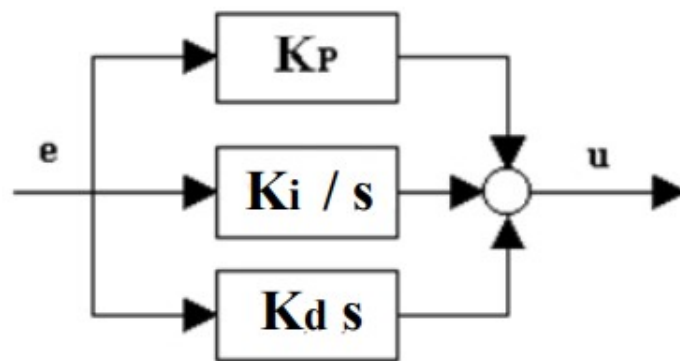


Figura 9.2: Estructura paralela de un controlador tipo PID.

Como se puede ver en la [Figura 9.2](#) cada acción (proporcional, integral y derivativa) viene acompañada de una constante. Estas serán las que se modificarán en el proceso de ajuste del controlador, dándole peso a cada acción, hasta conseguir la estabilización de la planta en régimen permanente.

- **Acción proporcional:** Consiste en realizar el producto de la señal de error por la constante de proporcionalidad con el objetivo de generar una señal de control con la magnitud necesaria para lograr estabilizar el sistema. No siempre es posible conseguir dicha estabilización con el uso único de la acción proporcional.
- **Acción integral:** Con la acción integral es posible corregir el error estacionario que no es capaz de corregir la acción proporcional.
- **Acción derivativa:** Esta acción es difícil de ajustar y es usada cuando se desea anticipar a la respuesta futura del error. Se consigue suavizar la señal de control aproximando la señal de error a una recta.

Existe otro parámetro modificable en el bloque de Simulink para el ajuste del PID. Este parámetro es la N y corresponde al coeficiente de filtro derivativo, que se trata de un filtro paso bajo para la acción derivativa.

9.4. Resultados experimentales

Para realizar el ajuste experimental del controlador se han realizado una serie de ensayos documentados en los cuales se han modificado los parámetros de los controladores P, PI y PID, hasta lograr la estabilización de la planta.

Una vez se encuentra estabilizada la planta se excita al sistema con una perturbación en la señal de control que hace que esta se sature durante un tiempo establecido, de esta forma se realiza un análisis de la robustez del sistema.

Los ensayos se han realizado con un periodo de muestreo de 0.001 segundos y 40 segundos de duración.

9.4.1. Resultados del ensayo N^o1 (Controlador tipo P)

En el primer ensayo realizado se ha buscado la estabilización de la planta modificando de forma manual la constante de proporcionalidad de ambos controladores, uno para la señal de error del carro y otro para la señal de error del péndulo. Este ajuste se ha realizado mientras el sistema de control se encontraba en funcionamiento, por lo que se podía observar en tiempo real cómo afectaba la modificación de estos parámetros a la respuesta del sistema real.

Los valores de los parámetros de los controladores tipo P ajustados son los que se muestran en la siguiente tabla.

	K_P	K_I	K_D	N
Controlador Péndulo	15	0	0	0
Controlador Carro	30	0	0	0

Tabla 9.1: Constantes ajustadas para el controlador tipo P

La respuesta que presenta el sistema ante una perturbación de magnitud 24 voltios (suficiente para saturar la señal de control) y con una duración de 0.05 segundos es la que se muestra en la siguiente figura.

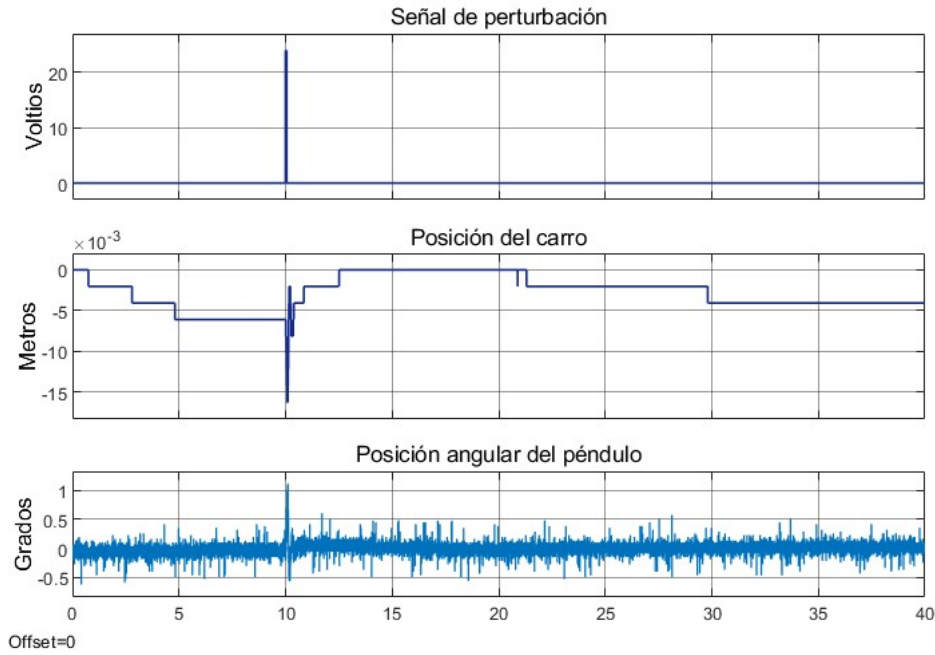


Figura 9.3: Respuesta del sistema de control tipo P ante una perturbación de duración 0.05 segundos a los 10 segundos de simulación

La perturbación introducida en el sistema ha tenido como consecuencia un desplazamiento angular del péndulo de aproximadamente 1 grado y una desviación de aproximadamente 15 milímetros en el desplazamiento del carro. Como se puede observar, el sistema de control ha conseguido estabilizar la planta ante dicha perturbación. El carro se ha mantenido prácticamente en su posición inicial sin llegar a presentar oscilaciones.

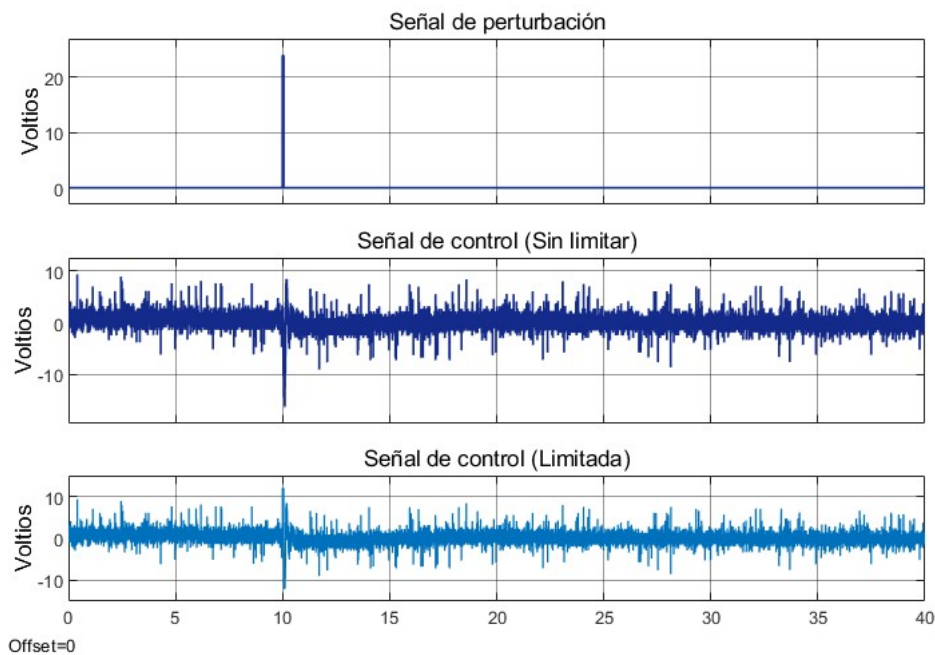


Figura 9.4: Señal de control obtenida de la respuesta de la [Figura 9.3](#)

En la [Figura 9.4](#) se observa cómo la señal de control solo llegó a saturarse en el momento de la perturbación y durante un periodo de tiempo muy corto.

Por lo tanto, se puede decir que se ha conseguido ajustar un controlador que cumple con los requisitos de diseño.

Para buscar una mejora en la robustez del sistema de control se aumenta la duración de la señal de perturbación a 0.1 segundos. La respuesta que se obtiene es la mostrada en la siguiente figura.

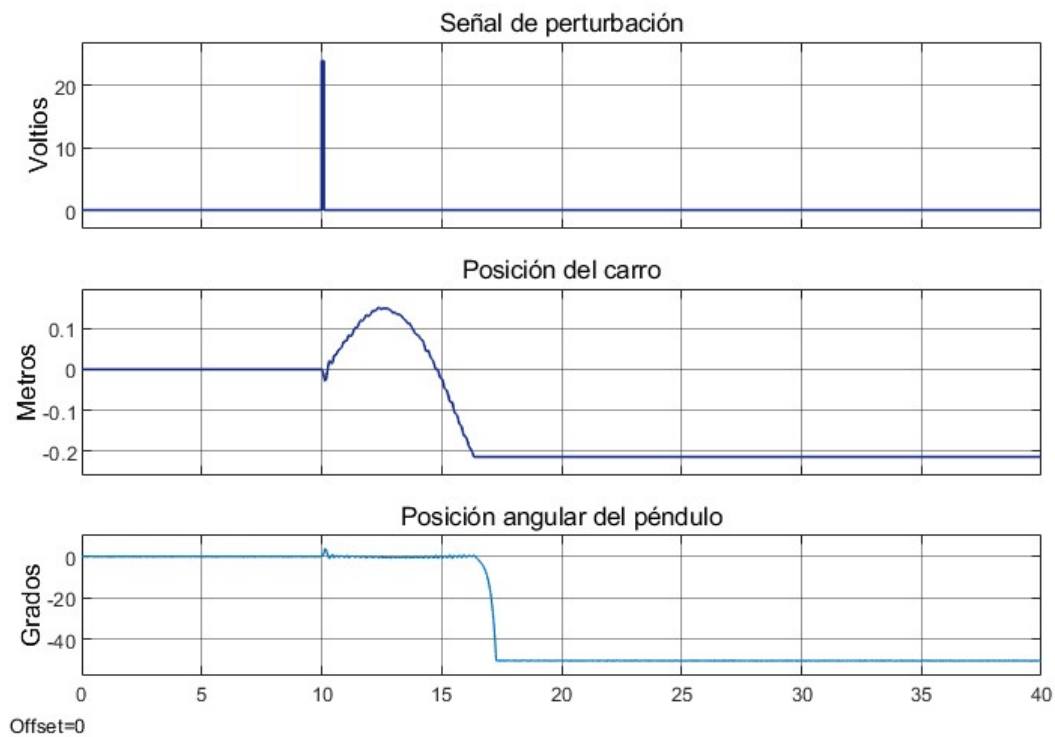


Figura 9.5: Respuesta del sistema de control tipo P ante una perturbación de duración 0.1 segundos a los 10 segundos de simulación

En este caso el controlador ajustado no ha sido capaz de controlar el sistema.

9.4.2. Resultados del ensayo N^o2 (Controlador tipo PI)

En este caso se intentan mejorar los resultados del ensayo anterior, por lo tanto, el sistema se perturbará desde un inicio con una señal de duración 0.1 segundos, para la cual el sistema de control anterior presentó una respuesta inestable.

Los parámetros de los controladores PI ajustados son los que se muestran en la siguiente tabla.

	K_P	K_I	K_D	N
Controlador Péndulo	25	5	0	0
Controlador Carro	30	2	0	0

Tabla 9.2: Constantes ajustadas para el controlador tipo PI

Para este ensayo se ha incrementado la constante de proporcionalidad del péndulo y se han añadido las constantes de integración en ambos controladores. Tras una serie de pruebas se ha logrado la respuesta de la siguiente figura con los parámetros de la [Tabla 9.2](#).

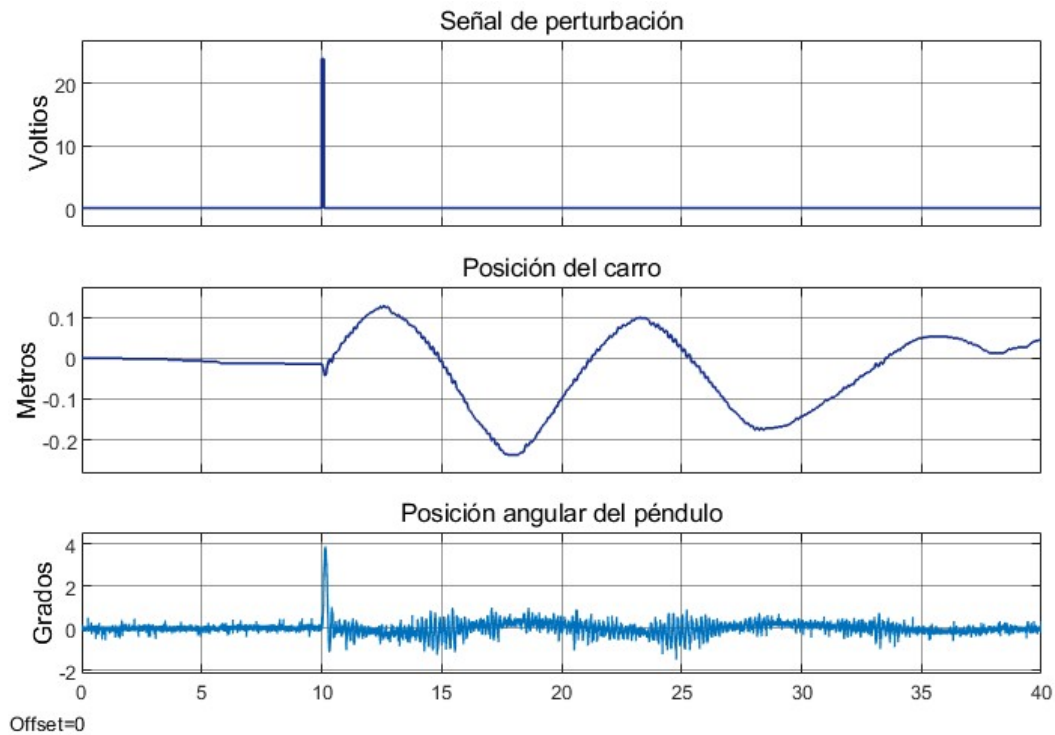


Figura 9.6: Respuesta del sistema de control tipo PI ante una perturbación de duración 0.1 segundos a los 10 segundos de simulación

En la respuesta representada en la [Figura 9.6](#) se muestra cómo el péndulo llegó a estabilizarse ante la perturbación dada. Aunque no se muestra la serie temporal completa, se puede observar cómo las oscilaciones, en la señal de desplazamiento del carro, convergen. En este ensayo el carro llegó a desplazarse hasta un total de 10 centímetros en cada dirección, y el péndulo en el momento de la perturbación llegó a desviarse hasta, aproximadamente, 4 grados.

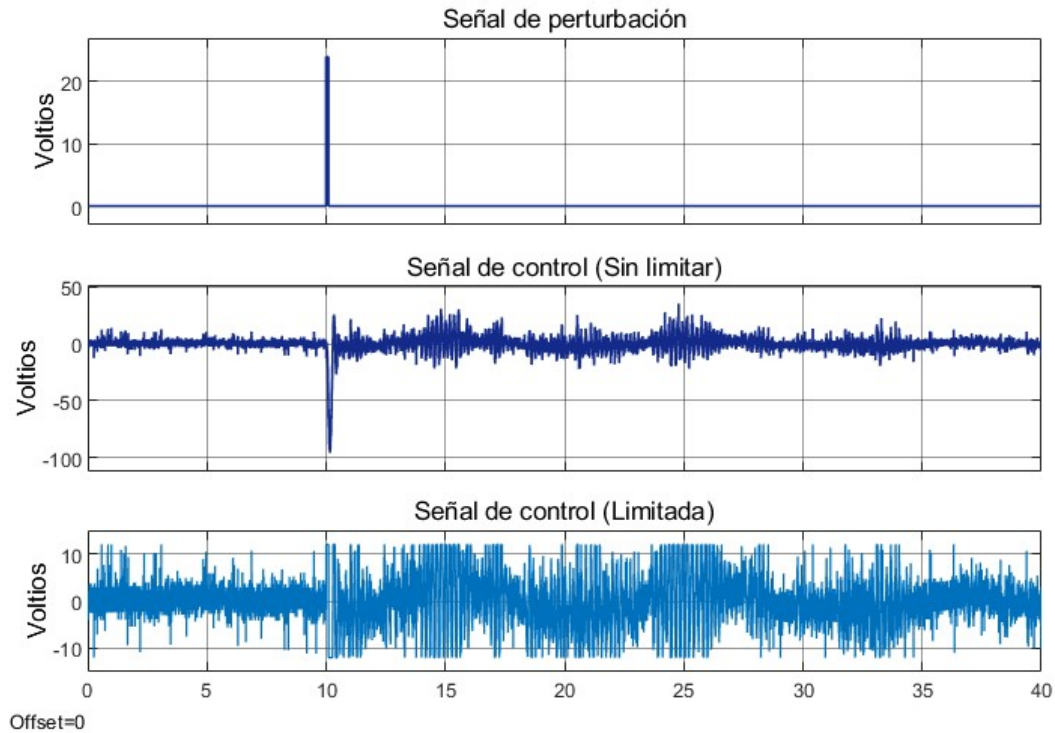


Figura 9.7: Señal de control obtenida en la respuesta de la [Figura 9.6](#)

Respecto a la señal de control esta llega a saturarse durante más tiempo que en el caso del controlador tipo P debido a que se encuentra mayor tiempo en régimen transitorio.

Se ha conseguido ajustar los dos controladores tipo PI para que el sistema consiga estabilizarse ante una perturbación que llega a saturar el controlador durante 0.1 segundos. Sin embargo, la respuesta presenta cierta oscilaciones que hacen que el tiempo de estabilización sea demasiado largo.

9.4.3. Resultados del ensayo N^o3 (Controlador tipo PID)

Para mejorar la respuesta oscilatoria que presenta el controlador ajustado en el caso anterior, se decide añadir una acción derivativa con filtro que suavice la señal de control y mejore la respuesta transitoria del sistema.

Los parámetros de los controladores PID ajustados son los que se muestran en la siguiente tabla.

	K_P	K_I	K_D	N
Controlador Péndulo	25	5	20	0
Controlador Carro	30	2	100	10

Tabla 9.3: Constantes ajustadas para el controlador tipo PID

Tras el ajuste manual de dichos parámetros para obtener una respuesta estable y reducir las oscilaciones del caso anterior, se obtiene la siguiente respuesta.

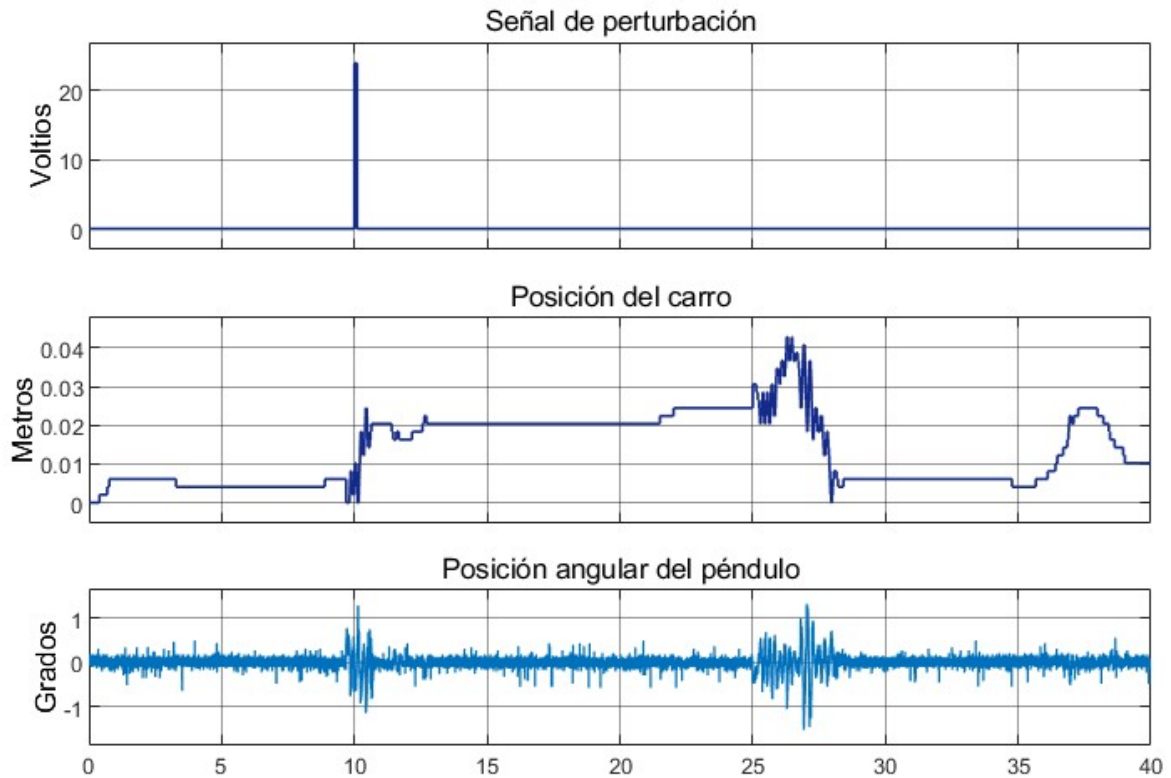


Figura 9.8: Respuesta del sistema de control tipo PID ante una perturbación de duración 0.1 segundos a los 10 segundos de simulación

Como se puede observar en la [Figura 9.8](#) el sistema llegó a estabilizarse presentando un régimen transitorio mucho más corto que en el caso del controlador PI. Se eliminaron las oscilaciones que presentaba el control tipo PI y el pico de desplazamiento del carro se redujo de 10 centímetros a aproximadamente 2 en el instante de la perturbación. Cabe destacar que la respuesta del carro presenta cierto error estacionario que no se consiguió eliminar.

Como puede ocurrir en cualquier sistema real, en este ensayo documentado se produjo una perturbación no controlada ni programada aproximadamente a los 27 segundos de simulación. Se desconoce qué produjo esta perturbación, pero el sistema de control logró de nuevo estabilizar la planta.

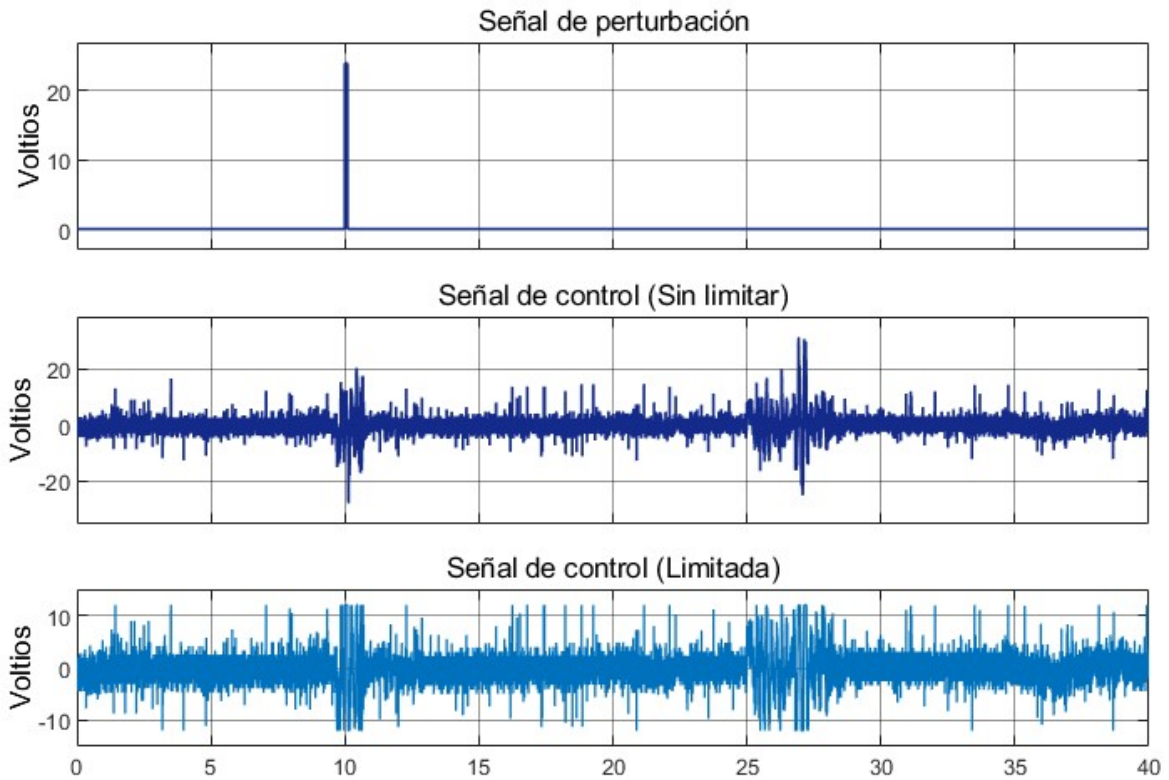


Figura 9.9: Señal de control obtenida en la respuesta de la [Figura 9.8](#)

Exceptuando los instantes en los que el sistema recibe la perturbación y la señal de control se satura por la necesidad de estabilizar la planta con una respuesta rápida, en régimen permanente el controlador se encuentra dentro de los límites del actuador.

Por lo tanto, se ha obtenido un controlador que cumple con los requisitos de diseño. Se ha conseguido estabilizar al sistema en un periodo de tiempo muy corto y sin presentar oscilaciones en la respuesta del carro.

Este mismo sistema de control se probó a excitar con una perturbación de mayor duración, en concreto 0.15 segundos. Como resultado se obtuvo la siguiente respuesta.

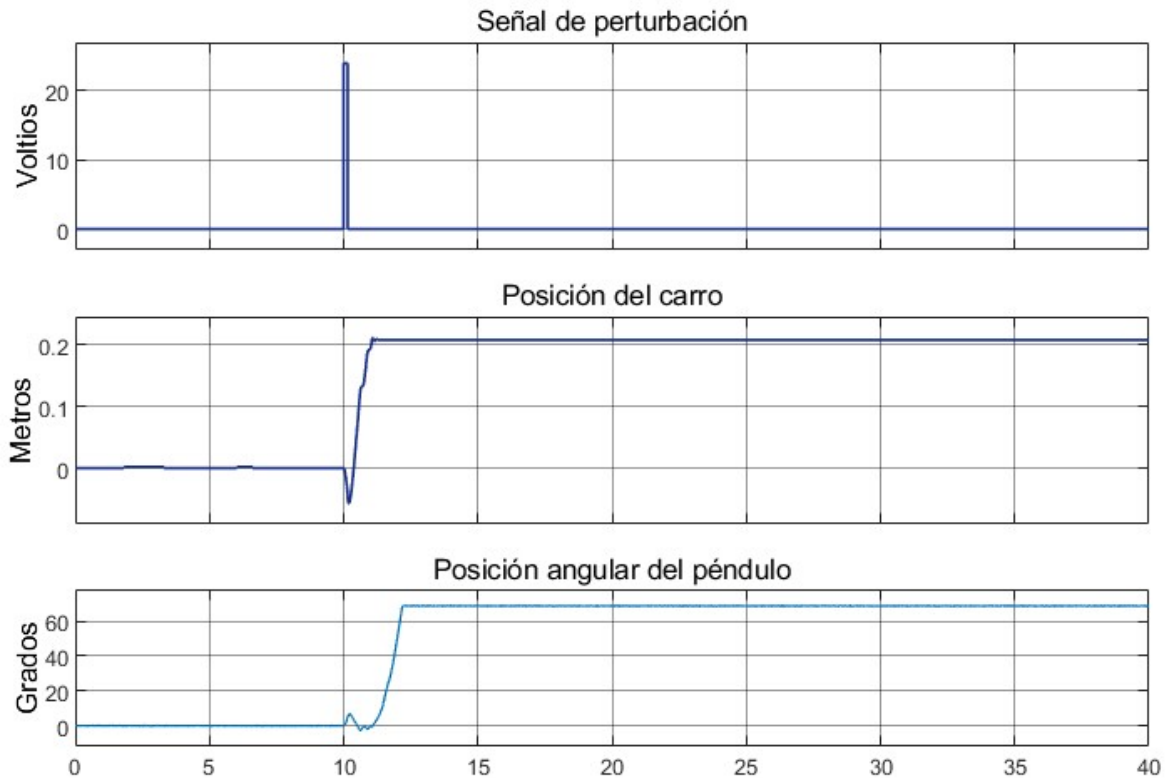


Figura 9.10: Respuesta del sistema de control tipo PID ante una perturbación de duración 0.15 segundos a los 10 segundos de simulación

Se intentó ajustar de nuevo los parámetros del controlador sin llegar a tener éxito en los resultados ante una perturbación de 0.15 segundos.

9.5. Comparación de resultados

Con el fin de poder visualizar la mejora obtenida en el controlador tipo PID, se procede a comparar los resultados obtenidos para dicho controlador con los que se obtuvieron para el controlador tipo PI.

Para ello, se almacenó en una variable cada resultado de las simulaciones, para posteriormente poder representar las respuestas que se obtuvieron en una misma gráfica mediante el Software Matlab.

No se compara el controlador tipo P debido que produjo una respuesta inestable ante la perturbación de 0.1 segundos.

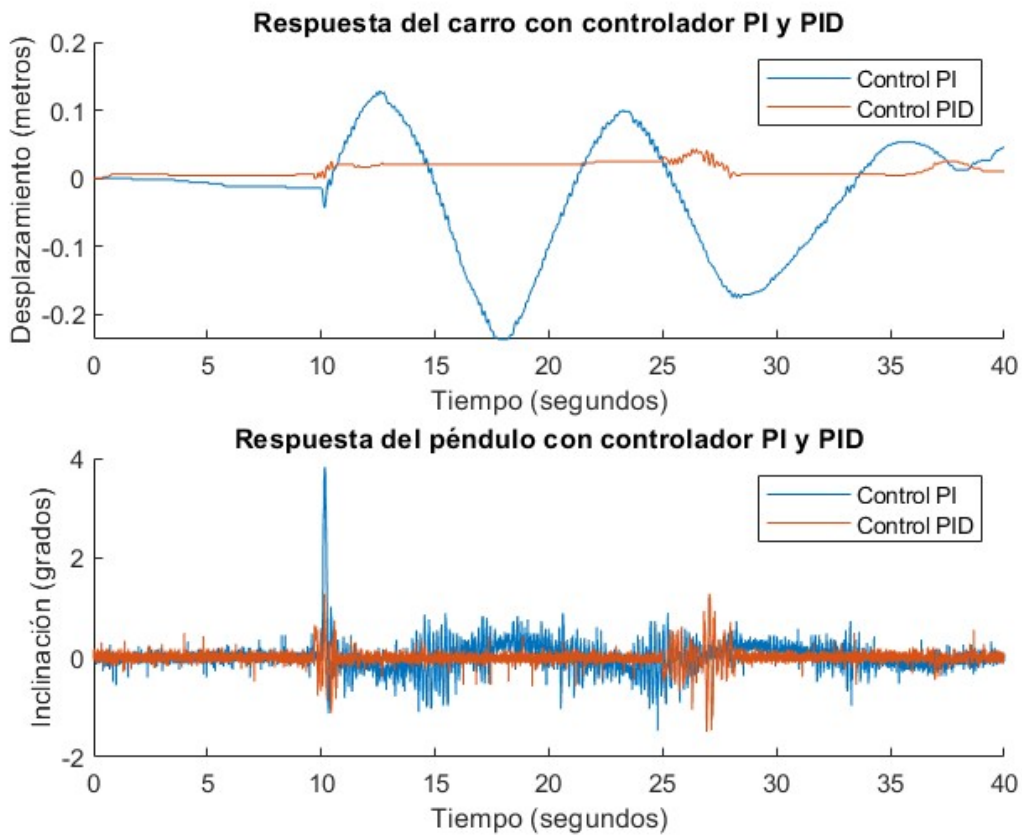


Figura 9.11: Comparación de los sistemas de control tipo PI y PID para una perturbación de 0.1 segundos a los 10 segundos de ensayo con una duración total de 40 segundos

En la comparación de resultados de la [Figura 9.11](#) se observan las respuestas que presentaron los sistemas de control tipo PI y PID para un ensayo de 40 segundos.

Se muestra gráficamente lo que ya se anunció en el ensayo del controlador PID. El controlador tipo PI muestra unas oscilaciones que llegan a converger en un periodo de tiempo largo. Por el contrario, en el controlador tipo PID se obtiene una respuesta sin oscilaciones en la respuesta del carro. Como diferencia en la respuesta del péndulo, se puede observar cómo en el sistema de control tipo PI el péndulo llegó a alcanzar un valor de inclinación mayor al del controlador tipo PID.

En resumen, el controlador PID presenta mejores resultados en todos los aspectos frente al controlador tipo PI.

Por último, se realizaron dos ensayos para los dos controladores con una duración de 120 segundos. Durante el ensayo cada 20 segundos se producía una perturbación que hacía desestabilizar el sistema. El objetivo de este ensayo fue observar cómo respondía el controlador tipo PI que presentaba un periodo de establecimiento alto con oscilaciones.

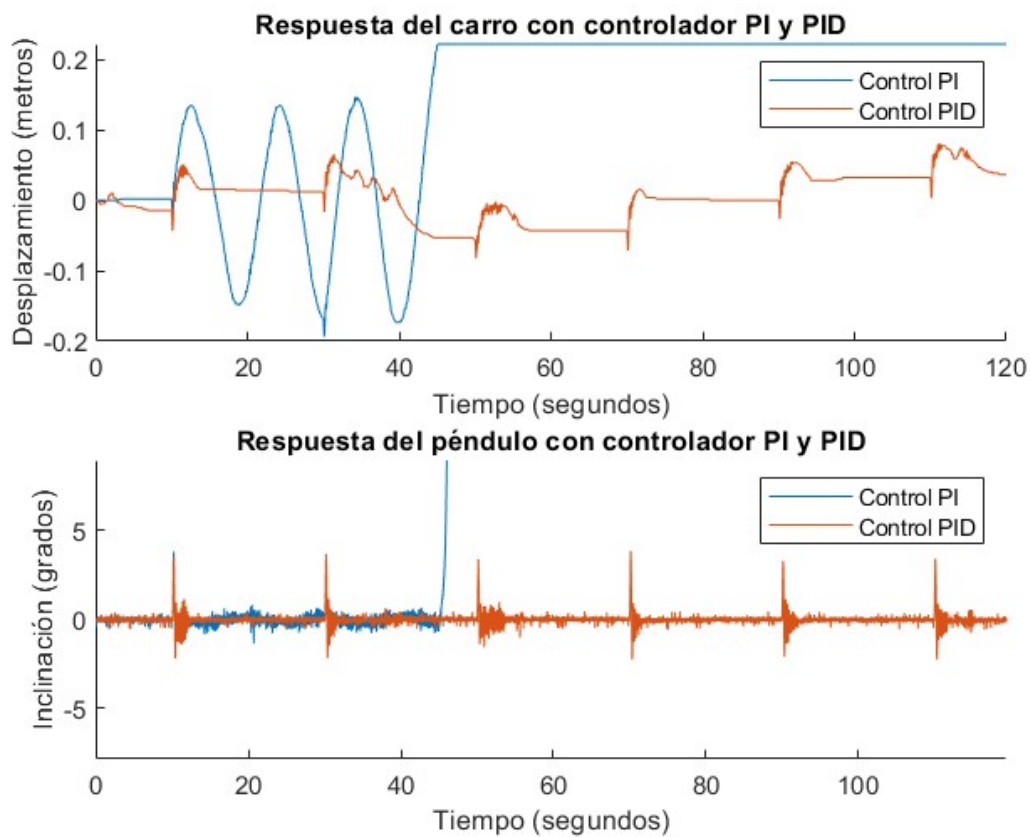


Figura 9.12: Comparación de los sistemas de control tipo PI y PID para una perturbación de 0.1 segundos cada 20 segundos de ensayo con una duración total de 120 segundos

En la primera perturbación la respuesta fue la esperada, el controlador PI mostró las mismas oscilaciones que en el ensayo de 40 segundos de duración. Al no conseguir estabilizarse antes de la segunda perturbación, el sistema se volvió inestable haciendo caer el péndulo y desplazar el carro hasta el límite del recorrido permitido. En el caso del controlador PID, este tuvo una respuesta estable, con periodos de establecimiento cortos y sin oscilaciones.

Capítulo 10

Conclusiones

Durante el periodo de estudio del sistema péndulo invertido se pudo comprobar que en la mayoría de los trabajos realizados, como por ejemplo [5], [3], [13] o [18] entre otros, no se tenía en cuenta la dinámica del actuador en el análisis del sistema. En el proceso de diseño del sistema de control que se presenta en dichos trabajos, se supone que la variable de control es una fuerza que puede ser directamente modificada. Sin embargo, si se desea implementar el sistema de control diseñado por algún método analítico en una planta real, se debe conocer el actuador, modelarlo y diseñar el controlador teniendo en cuenta sus ecuaciones. Y, por lo general, la señal de control generada por el sistema de control será una señal eléctrica.

Es por ello, que una parte del trabajo se ha centrado en obtener el modelo matemático del péndulo invertido y ampliarlo con el modelo matemático del actuador, en este caso un motor de corriente continua. Se obtuvo así un sistema de ecuaciones que se aproximaba al sistema real, y sobre este se diseñó el controlador analítico LQR. Aun habiendo centrado parte del esfuerzo en obtener un modelo preciso del sistema real, no se pudo implementar el controlador diseñado en el sistema construido. El principal motivo fue que no se disponía de los sensores necesarios para medir todas las variables de estados. Una solución analítica hubiese sido diseñar un observador de estados que estimase las variables no medidas, sin embargo esta solución implica un modelo matemático de la planta muy preciso, y no es el caso.

Tras la construcción del sistema péndulo invertido con motor de corriente continua como actuador, se pudo comprobar que la planta contaba con perturbaciones e incertidumbres no modeladas que dificultaban la implementación de un sistema de control diseñado por algún método analítico como, por ejemplo, el controlador LQR.

Fue por este motivo por el que se tomó la decisión de implementar un controlador experimental en el sistema real. Esta es una técnica muy utilizada en la industria, ya que por lo general se suele desconocer el modelo matemático de la planta, o el esfuerzo que conlleva obtenerlo no compensa en los resultados. El controlador utilizado finalmente en el diseño experimental fue un PID. Es el controlador experimental recomendado en [13] cuando se desconoce el modelo matemático de la planta.

El ajuste del controlador PID fue realizado manualmente mientras el sistema de control se encontraba en funcionamiento. Por lo tanto, en el proceso de sintonización se podía ir observando cómo se comportaba la planta mientras se ajustaban las constantes del controlador en cuestión. El procedimiento fue intentar controlar la planta con un controlador P en cada lazo de control e intentar mejorar la respuesta que tenía ante una perturbación añadiéndole a los controladores acción integral y/o derivativa.

Como resultado se obtuvo un controlador tipo PI en cada lazo de control que, ante una perturbación de 0.1 segundos de duración que saturaba la acción de control, tenía

una respuesta oscilatoria que convergía en un periodo de tiempo largo. Por lo tanto, el sistema lograba estabilizarse con éxito. Sin embargo, cuando esta perturbación ocurría cada 20 segundos al sistema no le daba tiempo de estabilizarse antes de que volviese a ocurrir la perturbación, dando como resultado la pérdida del control de la planta.

Finalmente, la implementación y ajuste de un controlador PID logró cumplir con los requisitos de diseño. Ante una perturbación de 0.1 segundos en la señal de control, el sistema lograba volver a su punto de estabilidad en un periodo de tiempo corto (aproximadamente 4 segundos). Por lo tanto, a diferencia del controlador tipo PI, el sistema de control PID no llegaba a descontrolarse en el caso de que las perturbaciones fuesen repetitivas.

Por lo tanto, el desarrollo del presente proyecto ha llevado a obtener un sistema real péndulo invertido con un motor de corriente continua como actuador, que al implementarle y ajustar un controlador tipo PID dicho sistema presenta una respuesta estable. Al analizar la robustez del sistema de control con una perturbación en la señal de control que hacía que se saturase durante 0.1 segundos, se ha obtenido que el sistema responde con una dinámica rápida logrando estabilizarse en aproximadamente 4 segundos.

En una continuación del trabajo realizado se podría agregar otro tipo de sensores, como por ejemplo una IMU, para realizar una fusión de sensores y poder medir todas las variables de estados. De esta forma, se podría implementar el controlador obtenido de forma analítica (LQR) y comparar los resultados con el controlador PID que se ha ajustado en el presente trabajo. Por lo tanto, el proceso llevado a cabo de modelado de la planta y diseño del controlador en el desarrollo del proyecto, aunque no ha tenido aplicación en el sistema real, pueden resultar las bases para nuevos proyectos.

Capítulo 11

Bibliografía

- [1] Cerón Morales, I., Vázquez Espinosa, J. O., and Aquino Díaz, E. (2017). Modelo matemático del motor de corriente directa. *Research Gate*.
- [2] Cortés Cardona, G. A. (2017). *Diseño, simulación y control de un péndulo invertido doble lineal para el laboratorio de automática de la EIEE de Univalle*.
- [3] Danilo, O., Giraldo, M., Valenzuela Hernández, J. G., and Buitrago, G. (2013). Balanceo y Estabilización del Péndulo Invertido Empleando Redes Neuronales Artificiales y un Regulador Lineal Óptimo con Criterio Cuadrático (LQR). *Scientia et Technica Año XVIII*, 18(1).
- [4] Ferrante, R. (2017). *A Robust Control Approach for Rocket Landing*.
- [5] García, J., Ramírez, L., Siordia, X., and Martínez, T. (2016). Las leyes de Newton en el modelado y control del péndulo invertido sobre un carro. *Revista Tecnología e Innovación*, 3(9):11–19.
- [6] Guerrero, J. M. (2000). *Captación de Parámetros Físicos y Tratamiento de Señal*. Dpto. de Ingeniería de Sistemas y Automática, Tecnología Electrónica y Electrónica y Electrónica de la UCA, Cádiz.
- [7] Hernández Santamaría, V. and Rojo Ramírez, J. R. (2011). *Péndulo triple invertido*. Trabajo final de la asignatura de Control Adaptativo y Robusto.
- [8] López Sánchez, M. J. (2021). *Apuntes de la asignatura de Técnicas Avanzadas de Simulación y Procesos Industriales*. Universidad de Cádiz.
- [9] López Sánchez, M. J. and García González, L. (2020). *Apuntes de la asignatura de Regulación Automática*. Universidad de Cádiz.
- [10] Mathworks[®] (2022). Programación de la Raspberry Pi con Matlab y Simulink. <https://es.mathworks.com/discovery/raspberry-pi-programming-matlab-simulink.html>. Accedido en Noviembre de 2022.
- [11] Morell Mirasol, F. J. (2019). *Diseño, implementación y control de un péndulo invertido rotacional*. Universidad Politécnica de Valencia, Trabajo fin de grado.
- [12] Ogata, K. (1996). *Sistemas de control en tiempo discreto*. Pearson, 2 edition.
- [13] Ogata, K. (2010). *Ingeniería de control moderna*. Pearson, 5 edition.
- [14] Pincin, S. J. (2016). *Diseño, construcción y control de un robot balancín*.

- [15] Raspberry[®] (2022). Raspberry Pi 4B. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. Accedido en Noviembre de 2022.
- [16] Rodríguez Rubio, F. and López Sánchez, M. J. (1996). *Control adaptativo y robusto*. Servicio de publicaciones de la Universidad de Sevilla.
- [17] SpaceX[®] (2022). Starship. <https://www.spacex.com/vehicles/starship/>. Accedido en Noviembre de 2022.
- [18] Triviño Macías, L. G. (2020). *Modelado, simulación y control de un péndulo invertido*. Universidad de Barcelona, Trabajo fin de grado.

Anexos

Anexo I: Código de Matlab para el análisis del sistema y diseño del controlador analítico

```
1 %Codigo de Matlab utilizado para obtener:
2 % Modelo matematico del sistema pendulo invertido
3 % Funciones de transferencia del sistema en lazo abierto y lazo cerrado
4 % Controlador LQR
5 % Autor: Jose Barroso Varo
6 % Universidad de Cadiz
7 % Noviembre de 2022
8 % Trabajo fin de grado
9
10 close all
11 clear
12 clc
13 %% Calculos pendulo-carro
14 % Calculos simbolicos para obtener expresiones
15 syms U b X m theta g M l s
16
17 % Relaciona ambas ecuaciones para obtener la relacion theta/U
18 eqn1 = ((U - b*X*s - m*theta*g)/(M*(s^2)))-X == 0;
19 X1 = solve(eqn1,X);
20 X1 = simplify(X1);
21
22 eqn2 = ((theta*g*(M+m) - U + b*X1*s)/(M*l*(s^2)))-theta == 0;
23 theta_U = solve(eqn2,theta);
24 theta_U = simplify(theta_U);
25
26 % Relaciona ambas ecuaciones para obtener la relacion X/U
27 eqn1 = ((theta*g*(M+m) - U + b*X*s)/(M*l*(s^2)))-theta == 0;
28 theta1 = solve(eqn1,theta);
29 theta1 = simplify(theta1);
30
31 eqn2 = ((U - b*X*s - m*theta1*g)/(M*(s^2)))-X == 0;
32 X_U = solve(eqn2,X);
33 X_U = simplify(X_U);
34
35 % Constantes del pendulo
36 M = 0.56; % Masa del carro
37 m = 0.35; % Masa del pendulo
38 b = 0.05; % Coeficiente de rozamiento
39 l = 0.547; % Longitud de la barra
40 g = 9.81; % Constante gravitacional terrestre
41 I = 0.0136; % Momento de inercia
42
43 s = tf('s');
44
45 % Las siguientes expresiones son las obtenidas en los calculos anteriores
46 % tras despejar la accion de control (U) para buscar la forma G=theta/U y
47 % G=X/U
```

```

48 Gla_theta = s/(b*g + M*g*s + g*m*s - M*l*s^3 - b*l*s^2);
49 Gla_X = (-l*s^2 + g)/(s*(b*g + M*g*s + g*m*s - M*l*s^3 - b*l*s^2));
50
51 zpk(Gla_theta);
52 damp(Gla_theta);
53 pzmap(Gla_theta);
54
55 % Conversion a tiempo discreto
56 ts = 0.001;
57 Gla_theta_d = c2d(Gla_theta,ts);
58 Gla_X_d = c2d(Gla_X,ts);
59
60 % Representacion grafica
61 figure(1)
62 subplot(2,1,1);
63 t = 0:0.01:1;
64 x1 = impulse(Gla_theta,t);
65 plot(t,x1);
66 grid;
67 ylabel('\theta (rad)');
68 xlabel('t (s)');
69 title('Respuesta del pendulo a una entrada tipo impulso');
70
71 subplot(2,1,2);
72 x2 = impulse(Gla_X,t);
73 plot(t,x2);
74 grid;
75 ylabel('x (m)');
76 xlabel('t (s)');
77 title('Respuesta del carro a una entrada tipo impulso');
78
79 % Conversion a tiempo discreto
80 ts = 0.001;
81 Gla_theta_d = c2d(Gla_theta,ts);
82 Gla_X_d = c2d(Gla_X,ts);
83
84 % Representacion grafica
85 figure(2)
86 t = 0:ts:1;
87 subplot(2,1,1);
88 x3 = impulse(Gla_theta_d,t);
89 plot(t,x3);
90 grid;
91 ylabel('\theta (rad)');
92 xlabel('t (s)');
93 title('Respuesta del pendulo a una entrada tipo impulso en tiempo
          discreto');
94
95 subplot(2,1,2);
96 x4 = impulse(Gla_X_d,t);
97 plot(t,x4);
98 grid;
99 ylabel('x (m)');
100 xlabel('t (s)');
101 title('Respuesta del carro a una entrada tipo impulso en tiempo discreto');
102

```

```

103 %% Analisis en lazo cerrado
104 Glc_theta = Gla_theta/(1 + Gla_theta);
105 Glc_X = Gla_X/(1 + Gla_X);
106
107 % Representacion grafica
108 figure(3)
109 subplot(2,1,1);
110 t = 0:0.01:1;
111 x5 = impulse(Glc_theta,t);
112 plot(t,x5);
113 grid;
114 ylabel('\theta (rad)');
115 xlabel('t (s)');
116 title('Respuesta del pendulo en lazo cerrado a una entrada tipo impulso')
    ;
117
118 subplot(2,1,2);
119 x6 = impulse(Glc_X,t);
120 plot(t,x6);
121 grid;
122 ylabel('x (m)');
123 xlabel('t (s)');
124 title('Respuesta del carro en lazo cerrado a una entrada tipo impulso');
125 %% Calculos motor
126 % Constantes motor
127 J = 0.01;      % Momento de inercia [Kg*m^2]
128 B = 0.5;      % Coeficiente de friccion [N*m/s]
129 Km = 0.1;     % Constante de torque [N*m/A]
130 Ka = 0.1;     % Constante contraelectromotriz [V/rad/s]
131 Ra = 3;       % Resistencia de armadura [Ohmios]
132 La = 0.5;     % Inductancia de armadura [H]
133
134 Gla_motor = (Km*(J*s + B))/(La*J*s^2 + (Ra*J + La*B)*s + Ra*B + Km*Ka);
135
136 num_Gla_motor = cell2mat(Gla_motor.Numerator);
137 den_Gla_motor = cell2mat(Gla_motor.Denominator);
138
139 figure(4)
140 t = 0:ts:1;
141 x7 = impulse(Gla_motor,t);
142 plot(t,x7);
143 grid;
144 ylabel('Torque (Nm)');
145 xlabel('t (s)');
146 title('Respuesta a una entrada impulso (Motor)');
147
148 %% Espacio de estados ampliado
149 % Rueda dentada
150 Radio = 0.005; % Radio de la rueda dentada [m]
151
152 % Matrices espacio de estados con actuador
153 A_ap = [0, 1, 0, 0, 0 ;
154         g*(M+m)/(M*1), 0, 0, b/(M*1), -Km/(Radio*M*1) ;
155         0, 0, 0, 1, 0;
156         -(m*g)/M, 0, 0, -b/M, Km/(Radio*M);
157         -(Ka*J*m*g*Radio)/(B*La*M), 0, 0, -(Ka*J*b*Radio)/(B*La*M) ,...
158         ((Ka*J*Km)/(B*La*M))-(Ra/La)-((Ka*Km)/(B*La))];

```

```

159 B_ap = [0 ; 0 ; 0 ; 0 ; 1/La];
160 C_ap = [1, 0, 0, 0, 0 ; 0, 0, 1, 0, 0 ; 0, 0, 0, 0, 1];
161 D_ap = 0;
162
163 % Representacion espacio de estado
164 Gla_ap_ss = ss(A_ap, B_ap, C_ap, D_ap);
165 damp(Gla_ap_ss);
166
167 % Respuesta del sistema pendulo-motor en lazo abierto
168 figure(5)
169
170 subplot(3,1,1);
171 x6 = impulse(Gla_ap_ss(1), t);
172 plot(t, x6);
173 grid;
174 title('Respuesta de la posicion angular del pendulo');
175 ylabel('\theta (rad)');
176 xlabel('t (s)');
177
178 subplot(3,1,2);
179 x7 = impulse(Gla_ap_ss(2), t);
180 plot(t, x7);
181 grid;
182 title('Respuesta de la posicion lineal del carro');
183 ylabel('x (m)');
184 xlabel('t (s)');
185
186 subplot(3,1,3);
187 x8 = impulse(Gla_ap_ss(3), t);
188 plot(t, x8);
189 grid;
190 title('Respuesta de la corriente del actuador');
191 ylabel('ia (A)');
192 xlabel('t (s)');
193
194 %% Control LQR
195 % Valores para la matriz Q y Rca
196 % Estos valores son arbitrarios
197 q1 = 100; % Peso asociado a la inclinacion del pendulo
198 q2 = 0;
199 q3 = 10; % Pero asociado a la posicion del carro
200 q4 = 0;
201 q5 = 0;
202
203 % Matrices de ponderacion del sistema
204 Qc = diag([q1, q2, q3, q4, q5]); % Peso asociado a cada estado
205 Rc = 0.1; % Este valor tambien es arbitrario
206 % Peso asociado al esfuerzo de control
207
208 % Calcula la matriz de realimentacion del vector de estado
209 Kc_lqr = lqr(A_ap, B_ap, Qc, Rc);
210
211 % Sistema en lazo cerrado para LQR
212 % Gyr
213 Ayr_lqr = A_ap - B_ap * Kc_lqr;
214 Cyr_lqr = C_ap - D_ap * Kc_lqr;
215

```

```
216 PreC = 0; % Porque se trata de un problema regulacion
217 Byr_lqr = B_ap*PreC;
218 Dyr_lqr = D_ap*PreC;
219
220 Glc_ap_ss = ss(Ayr_lqr ,Byr_lqr ,Cyr_lqr ,Dyr_lqr );
221 damp( Glc_ap_ss );
```

Anexo II: Procedimiento de encendido, puesta en marcha y parada

En el presente Anexo se van a enumerar los pasos a seguir para hacer un correcto manejo del sistema. Es importante seguir este procedimiento para evitar averías debido a un mal uso.

1. Comprobar que el controlador de motor no se encuentra alimentado por una fuente externa, o esta se encuentra apagada.
2. Conectar la Raspberry Pi al la red y al PC Host por puerto Ethernet.
3. Una vez la Raspberry Pi se encuentre iniciada, iniciar la comunicación entre el PC Host y la Raspberry Pi (Ver [Capítulo 7](#)).

En el desarrollo de este proyecto la Raspberry Pi utilizada contaba con la IP: 169.254.243.51

Por lo tanto, el comando para iniciar la comunicación era:

```
1 rpi=raspberrypi( '169.254.243.51 ')
```

4. Ejecutar el 'Script' de Matlab para cargar los parámetros *ts*, *limit*, *Vmin* y *radio*.
5. Situar el carro en la mitad del recorrido
6. Colocar el péndulo en posición vertical de forma manual hasta que este mantenga el equilibrio (sin control).
7. En Simulink, comentar el subsistema 'PID Péndulo-Carro' y abrir el subsistema 'Ajuste Offset'. En la pestaña *Hardware* ejecutar el esquema con *Monitor & Tune* durante 10 segundos.

Al finalizar se obtendrá una serie de medidas del sensor de posición angular del péndulo en voltios. Con el siguiente comando se calcula la media de estos valores, dando como resultado el Offset del sensor para establecer la posición 0:

```
1 offset = mean(signal_pend);
```

Además las salidas de la Raspberry Pi que controlan el motor se establecen a nivel bajo para poder alimentar el controlador de motor con seguridad.

8. Encender la fuente externa que alimenta el controlador de motor. La tensión de alimentación son 12 voltios.
9. En Simulink, comentar el subsistema 'Ajuste Offset' y des-comentar el subsistema 'PID Péndulo-Carro'.
10. Dentro de este bloque se tiene un *Switch* para activar o desactivar el motor mientras se esté ejecutando el algoritmo, otro *Switch* para activar o desactivar

las perturbaciones, un potenciómetro para ajustar el offset, si fuese necesario, en simulación, y una serie de potenciómetros para realizar los ajustes de los PID.

Con el *Switch* del motor activado iniciar el sistema de control en la pestaña *Hardware* y pinchando en *Monitor & Tune*. Utilizar el tiempo de simulación que se desee.

El algoritmo tarda unos segundos en iniciarse. Si no se ha realizado bien el procedimiento la respuesta puede ser imprevisible. Por lo tanto, se recomienda estar atento para des-energizar la planta si fuese necesario.

11. Antes de que finalice el tiempo de muestreo, desactivar manualmente el motor con el *Switch* correspondiente que se encuentra en Simulink. A continuación, se puede detener o dejar que finalice el experimento.
12. Si se desea volver a iniciar, volver a activar el *Switch* y ejecutar.
13. Si en este punto diese un fallo de espacio en la Raspberry Pi, realizar el siguiente procedimiento:
 - a) En Matlab ejecutar el comando: *openShell(rpi)*
 - b) En la ventana de comandos de la Raspberry Pi dirigirse a la carpeta que se indica en el error y eliminarla
14. Para apagar el sistema, desconectar la fuente de alimentación del controlador de motor.
15. A continuación, apagar la Raspberry Pi desde su ventana de comandos.
16. Una vez la Raspberry Pi haya terminado su proceso de apagado, se puede desconectar del PC Host y de la red eléctrica.

Anexo III: Diagramas de Simulink para las pruebas experimentales

Diagrama de Simulink para el ajuste del Offset del sensor de posicionamiento angular.

