



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA
ITINERARIO ELECTRÓNICO

**CONTROL DE UN PÉNDULO INVERTIDO
MEDIANTE VARIACIÓN DEL MOMENTO
ANGULAR**

Autor: Santiago García Serrano

Director: Juan Luis Zamora Macho

Director: José Porras Galán

Madrid

Julio 2016

Santiago
García
Serrano

CONTROL DE U PÉNDULO INVERTIDO MEDIANTE VARIACIÓN DEL MOMENTO ANGULAR



AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. SANTIAGO GARCÍA SERRANO

DECLARA ser el titular de los derechos de propiedad intelectual de la obra:

CONTROL DE UN PÉNDULO INVERTIDO MEDIANTE VARIACIÓN DEL MOMENTO ANGULAR,

que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor CEDE a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

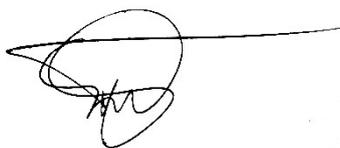
6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 18 de Julio de 2016

ACEPTA



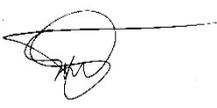
Fdo.....

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
CONTROL DE UN PÉNDULO INVERTIDO MEDIANTE VARIACIÓN DEL
MOMENTO ANGULAR

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2015/16 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni
total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Santiago García Serrano

Fdo: 

Fecha: 18 / Julio / 2016

Autorizada la entrega del proyecto cuya información no es de carácter confidencial

Directores del proyecto:

Juan Luis Zamora Macho

Fdo: Fecha:/...../.....

José Porrás Galán

Fdo: 
.....

Fecha: 18 / Julio / 2016

VºBº del Coordinador de Proyectos

Álvaro Sánchez Miralles

Fdo: Fecha:/...../.....

(Esta página se ha dejado en blanco a propósito)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA ELECTROMECÁNICA

ITINERARIO ELECTRÓNICO

**CONTROL DE UN PÉNDULO INVERTIDO
MEDIANTE VARIACIÓN DEL MOMENTO
ANGULAR**

Autor: Santiago García Serrano

Director: Juan Luis Zamora Macho

Director: José Porras Galán

Madrid

Julio 2016

Santiago
García
Serrano

CONTROL DE U PÉNDULO INVERTIDO MEDIANTE VARIACIÓN DEL MOMENTO ANGULAR



Resumen

Introducción y objetivos

Uno de los mecanismos más utilizados en los últimos años para investigar sobre la robótica, en especial las aplicaciones de la electrónica de control a la robótica es el péndulo invertido. Esto se debe a que es un tipo de mecanismo de los denominados sistemas mecánicos subactuados, cuyo gran interés se debe en gran parte a que pueden ofrecer un considerable ahorro de energía y de esfuerzos de control respecto a los sistemas completamente actuados. Esto se consigue gracias a que tienen menos actuadores que grados de libertad y por lo tanto se simplifican los controles.

Este proyecto trata diseñar los primeros pasos de un cubo que pueda saltar y equilibrarse sobre una arista o vértice mediante el diseño de un sistema de control de una de las caras del cubo. La actuación, en este caso, se realiza modificando el momento angular de un volante de inercia acoplado a dicha cara utilizando control digital.

Metodología

Para cumplir los objetivos del proyecto se han de seguir una serie de pasos:

1. Diseño y fabricación de las piezas
2. Diseño de un control en simulaciones
3. Aplicación del control al sistema real

Diseño y fabricación de las piezas

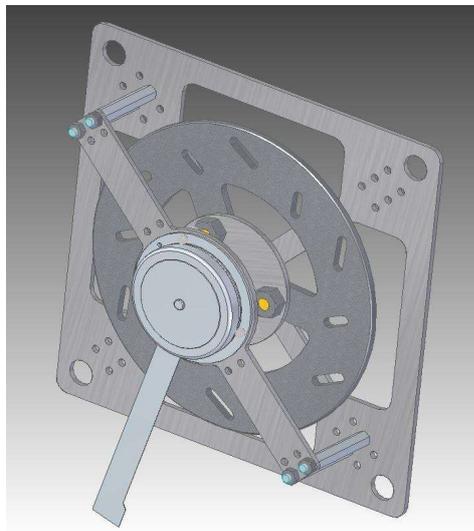
El diseño y la fabricación de las piezas es vital para el correcto funcionamiento del péndulo invertido controlado por el volante de inercia. Lo más importante a tener en cuenta es la ligereza del material, para necesitar menos potencia de motor en las fases de salto y equilibrio, y la resistencia, puesto que en la fase de salto se somete al péndulo, en especial al sistema de frenado y el eje del motor a una carga de fuerzas elevada.

El diseño de las piezas se hace utilizando la herramienta de diseño de CAD en 3D Solid Edge. Solid Edge no solo permite la visualización de las piezas por separado, sino que tiene un formato de archivo en el cual se pueden unir las diferentes piezas para comprobar el resultado final. Lo primero que se diseña es el péndulo invertido sin ningún

JULIO 2016

sistema de salto, que sea solamente capaz de equilibrarse liberándolo en un ángulo diferente de 0° , y lo segundo es el sistema de frenado para realizar la fase de salto.

Las piezas del péndulo se fabrican todas de aluminio excepto el volante de inercia, que es de acero. Debido a la dificultad de fabricar un volante de inercia perfectamente equilibrado sobre su eje con los recursos disponibles, se adquiere un disco de freno de una minimoto de acero, que cumple con las características necesarias de estar equilibrado, aportar un par suficiente para equilibrar el péndulo, y ser lo suficientemente resistente como para aguantar el impacto para saltar.



Diseño del péndulo invertido con volante de inercia

El sistema de frenado consta de un servomotor, sujeto a la pieza que soporta el motor, cuyo brazo golpea a un tornillo anclado a una de las ranuras del volante de inercia. Para evitar que los engranajes se rompan, se posiciona el servo de tal manera que el impacto lo reciba en la dirección de sus ejes. La pieza que sujeta el servo, se fabrica de plástico utilizando una impresora 3D debido a la dificultad de fabricarla en aluminio.



Diseño del péndulo invertido con volante de inercia y sistema de frenado

Diseño del control

Para controlar el sistema, es necesario separar las dos fases, la de equilibrio y la de salto. El diseño de la fase de equilibrio se lleva a cabo mediante un control LQR. El control LQR es un algoritmo de control que trata de minimizar la función de coste de un sistema dinámico definido mediante ecuaciones diferenciales lineales para hallar las ganancias de realimentación de estado. Para hallar los parámetros óptimos que minimizan la función de coste, hay que dar unos pesos a las variables de estado del sistema para tratar de optimizar el control centrándose más en unas variables u otras.

Las variables de estado del sistema son el ángulo del péndulo (θ_b), la velocidad angular del péndulo (ω_b), y la velocidad angular del volante de inercia (ω_w).

Estas fases se unen mediante la utilización de una máquina de estados de cuatro estados:

1. Estado inicial: Calibración de las medidas de las variables de estado para eliminar offsets.
2. Aceleración: Aceleración del volante de inercia hasta alcanzar la velocidad suficiente para realizar el salto.
3. Salto: Activación del sistema de frenado para efectuar el salto.
4. Balanceo: Activación del control LQR para equilibrar el péndulo sobre el ángulo $\theta_b = 0$.

JULIO 2016

La fase de salto, se controla haciendo un ensayo de prueba y error. Este ensayo consiste en acelerar el volante hasta una velocidad determinada, y activar el sistema de frenado. La velocidad se aumenta poco a poco hasta que se consiga que el péndulo alcance un ángulo suficiente como para que al activar el control de balanceo sea capaz de equilibrarse.

Implementación en el sistema real

El microcontrolador seleccionado para realizar la tarea de controlar el sistema es el STM32F407 Discovery. Este procesador tiene una gran capacidad de cálculo que sirve para poder realizar las tareas explicadas en la máquina de estados.

Para la implementación del control en el microcontrolador se utiliza los bloques Waijung de Simulink. Los bloques Waijung sirven para generar automáticamente código en C a partir de Matlab/Simulink para procesadores de la familia STM32F4.

Las variables de estado se miden de diferentes métodos. El ángulo y la velocidad angular del péndulo se estiman mediante la utilización de dos IMU colocadas en la diagonal del péndulo. Utilizando las medidas de los acelerómetros se obtiene θ_b , y las de los giróscopos se obtiene ω_b . Para ω_w se utilizan los sensores hall del motor. El mando del sistema, la corriente del motor, se envía utilizando una de las salidas analógicas de la placa.

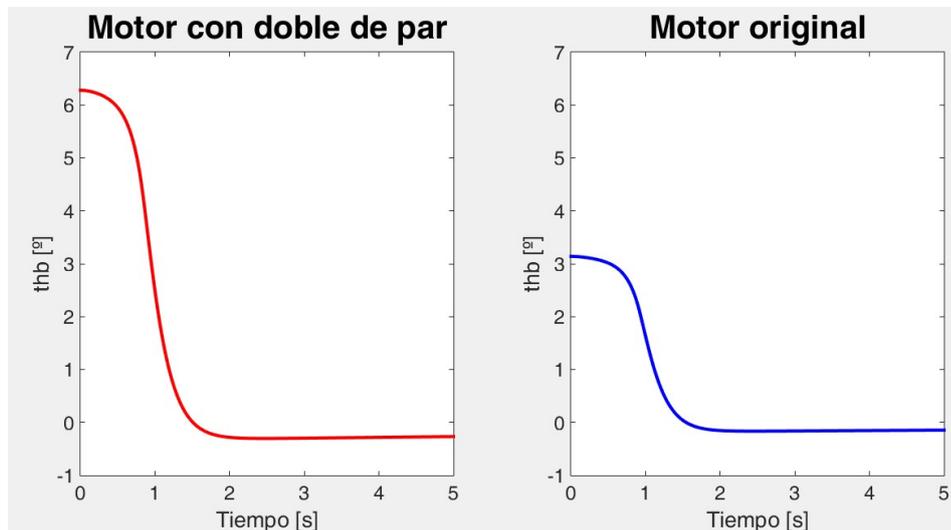
Resultados y conclusiones

Se ha diseñado un prototipo de péndulo invertido partiendo de cero. En cuanto al diseño mecánico del péndulo invertido, para equilibrar el péndulo sobre su esquina, es correcto en su mayoría. Se puede mejorar el eje sobre el que se equilibra para disminuir las vibraciones que surgen al girar el volante de inercia. En la fase del salto, el diseño no ha sido correcto. Cuando se acelera a la velocidad necesaria para poder realizar el salto, el brazo del servo no es capaz de resistir el impacto, si es de plástico se dobla, y si es metálico se rompe el tornillo que lo sujeta.

Después de lidiar con los problemas de tener que programar y configurar toda la arquitectura de control y comunicaciones del sistema, en algunos casos registro a registro, se ha conseguido realizar una identificación de parámetros y lectura de variables capaz de simular y controlar el sistema correctamente.

El diseño del control se ha optimizado lo máximo posible. El ángulo de límite que es capaz de equilibrar en simulaciones es de 3.14° . Este ángulo es muy pequeño, y a la hora de implantar el control utilizado en el sistema real, no es capaz de equilibrarse debido al ruido en las medidas provocado por las vibraciones en el eje sobre el que está fijo el péndulo, aunque se ha encontrado una solución para ese problema, mejorar el eje inferior instalando algún tipo de amortiguamiento.

Se ha comprobado en las simulaciones que el motor escogido no es lo suficientemente potente para proporcionar el par necesario. De esta manera, implantando un motor más potente, el sistema acelerará lo suficientemente rápido para contrarrestar la energía potencial y estabilizarse.



Comparativa resultados en simulación utilizando el motor original y un motor con el doble de par de inercia

En conclusión, modificando el sistema de frenado, amortiguando la sujeción del eje, y utilizando un motor más potente, se puede aplicar el control sobre el sistema real sin la necesidad de realizar grandes modificaciones en el diseño.

JULIO 2016

Abstract

Introduction and objectives

One of the devices that has been used the most in the last few years in robotics investigations, specifically in the applications of control electronics applied this field, is the inverted pendulum. This is because it is a special type of mechanism called underactuated mechanical system. The great interest on them mainly comes from the considerable amount of energy and effort they save to control compared to other type of systems. This is achieved thanks to having a smaller amount of actuators than degrees of freedom, which leads to a simpler control.

This project consists in trying to design the first steps of a cube that can jump and balance on its edge or corner by designing a digital control system for one of the cube's faces. The actuator, in this case, is made up of an inertia wheel that gives angular momentum to the pendulum.

Methodology

To be able to achieve the goals of the project, the following steps need to be accomplished:

1. Design and manufacturing of the components of the pendulum.
2. Design of a digital control for simulations.
3. Applying the control to the actual system

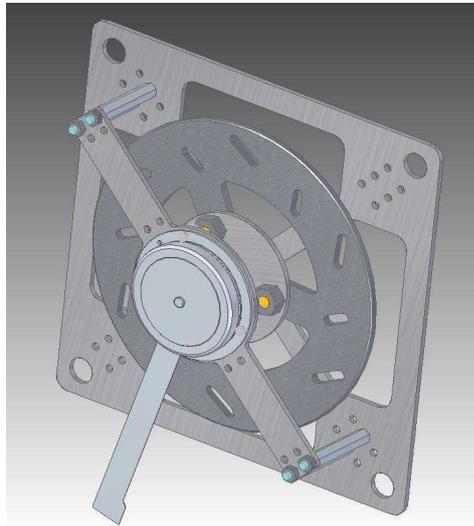
Design and manufacturing of the components

The design and manufacturing of components is vital to the inverted pendulum controlled by an inertia wheel to work correctly. The most important parts to take into account are the material's weight and resistance. The first one so that less motor power is needed to jump and balance, and the second one, because the pendulum, especially the braking system and the motor axis, receive a great force during the jump up action.

The design of the components is done by using a 3D CAD tool design in Solid Edge. Solid Edge does not only allow to visualize the separate parts on their own, it has a special type of file format in which all the different components can be joined together to check the final result. The first thing to design is the inverted pendulum without any

braking system attached, it only needs to be able to balance itself when let go from a small angle different from 0° . The last thing is the braking system to carry out the jumping action

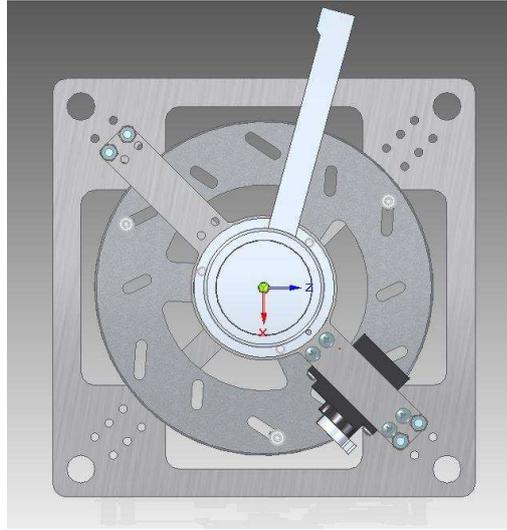
The different parts of the pendulum are all manufactured in aluminium except from the inertia wheel, which is made out of steel. Due to the difficulty of manufacturing a perfectly balanced on its axis inertia wheel with the available resources, a small motorbike disc brake is purchased. The advantage of using a motorbike braking disc is that it possesses the required features of being completely balanced, provides enough moment to balance the pendulum, and is resistant enough to tolerate the impact received when braking to jump.



Design of the inverted pendulum with an inertia wheel

The braking system is made up of a servo fixed to the piece that holds the motor, which. The servo's arm hits a screw secured to one of the slots near the edge of the inertia wheel and causes it to brake instantly. To avoid the servo gears from breaking, the servo is pointed in a way so that the impact is received in the direction of its edges. The piece that holds the servo is manufactured in plastic using a 3D printer due to the difficulty of making it out of aluminium.

JULIO 2016



Design of the inverted pendulum with an inertia wheel and braking system

Control design

To control the system, the separation of the two main phases, jumping and balancing action, is required. The balancing action design is accomplished by the use of a LQR control. The LQR control is a control algorithm that tries to minimize the cost function of a dynamic system defined by differential equations to find the state feedback gain matrix. To find the parameters that minimize the cost function, different weights are given to the state variables of the system so that the control tries to optimize the result focusing on some variables or others.

The state variables of the system are the pendulum angle (θ_b), the angular velocity of the pendulum (ω_b), and the angular velocity of the inertia wheel (ω_w).

Both phases are joined together by using a four states finite-state machine:

1. Initial State: Calibration of the measurements of the state variables to eliminate any possible offsets.
2. Acceleration: Acceleration of the inertia wheel until the speed needed to make the jump is achieved.
3. Jumping action: Activation of the braking system to jump.
4. Balancing action: Activation of the LQR control to balance the pendulum on an angle $\theta_b = 0$.

The jump up action phase is controlled by carrying out a trial and error test. This test consists on accelerating the wheel up to a determined speed and then

activating the braking system. The speed is raised by small amounts until the pendulum achieves an angle small enough so that when activating the control it manages to balance on its corner.

Implementation on the physical system

The microcontroller selected to execute the task of controlling the whole system is the STM32F407 Discovery. This microprocessor has a great calculations capacity which allows to accomplish the jobs explained in the finite-state machine.

Simulink Waijung's Blockset is used for the implementation of the control into the microcontroller. Waijung Blockset is used to automatically generate C code from Matlab/Simulink simulation models for many kinds of microcontrollers, including those of the STM32F4 family.

The state variables are measured using different methods. The angle and angular velocity of the pendulum are estimated by using two Inertial Measurement Units positioned in the diagonal of the pendulum. Using the measurements of the accelerometers, θ_b is obtained, and the gyroscopes give ω_b . The motor's hall sensors are used to obtain the last state variable, ω_w . The input to the system, the motor's current, is sent by using one of the analogue pins of the board.

Results and conclusions

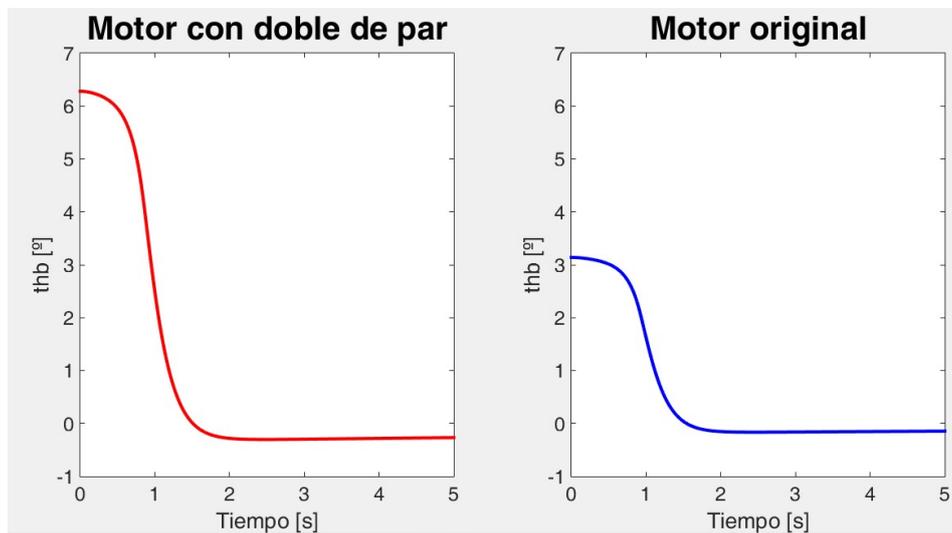
A prototype of an inverted pendulum has been designed from scratch. The mechanical design of the inverted pendulum, of which the main purpose is for it to balance on its corner, is correct. The method used to hold the pendulum vertically, the axis, can be improved by trying to reduce the vibrations that appear when the inertia wheel spins. On the other hand, the jumping action the design was not good enough. When accelerating to the necessary speed to start jumping off the ground to achieve an angle where the balancing control is able to work correctly, the servo's arm cannot resist the impact. If the material is plastic, it bends, if it is metallic, the screw that holds it in place is the one that breaks.

After dealing with the trouble of having to configure the whole architecture of the control and communications of the system, in some cases even register by register, an identification of parameters and variable readings that is able to simulate and control the whole system correctly was achieved.

JULIO 2016

The control design has been optimized as much as possible. The maximum angle that is able to recover balance from in simulations is 3.14° . This is a very small angle, and when trying to load the control y the physical system, it is not able to balance due to the noise in the measurements caused by the vibrations on the axis that holds it vertically, although a solution for that problem was proposed, using a damping system to reduce vibrations.

On the other hand, it can be shown that using a motor that is able to supply a greater moment of inertia to the wheel, the control can be improved significantly if the rest of the parameters are kept constant, as the more powerful motor will have a greater acceleration to counteract the effects of gravity and stabilize better.



Comparison of the simulation results using the original motor and a motor with double as much moment of inertia

In conclusion, by modifying the braking system, installing a damping system to the axis, and using a more powerful motor, the control used in this project can be applied to the physical system without the need of making a great deal of modifications.

Índice de la memoria

Índice de figuras	v
Índice de ecuaciones.....	ix
Parte 1: Memoria	- 1 -
Capítulo 1: Introducción.....	- 3 -
1.1. Estado del arte	- 3 -
1.1.1 Sistema de Péndulos	- 4 -
1.1.2. Proyectos basados en péndulos invertidos.....	- 7 -
1.2. Motivación	- 9 -
1.3. Objetivos	- 10 -
1.4. Metodología	- 10 -
1.4.1. Tareas.....	- 10 -
1.5. Recursos	- 11 -
1.5.1. Software.....	- 11 -
1.5.2. Hardware.....	- 11 -
Capítulo 2: Diseño de péndulo invertido con volante de inercia.....	- 13 -
2.1. Diseño general del péndulo invertido.....	- 13 -
2.2. Motor y volante de inercia.....	- 14 -
2.3. Sistema de frenado	- 18 -
2.4. Sistema total	- 19 -
Capítulo 3: Fabricación de piezas.....	- 21 -
3.1. Impresora 3D.....	- 21 -
3.2. Fresadora automática.....	- 21 -
Capítulo 4: Modelado y dinámica del péndulo invertido	- 25 -
4.1. Dinámicas del sistema.....	- 25 -

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

4.2.	Estimación de las variables de estado	- 28 -
4.2.1.	Estimación del ángulo (θ_b).....	- 28 -
4.2.2.	Estimación de la velocidad de rotación de la cara del péndulo (ω_b).....	- 29 -
4.2.3.	Estimación de la velocidad de rotación del volante de inercia (ω_w).....	- 29 -
Capítulo 5:	Protocolos de comunicación.....	- 31 -
5.1.	Puerto serie (UART)	- 31 -
5.2.	Protocolo I2C	- 32 -
Capítulo 6:	Identificación de parámetros	- 35 -
6.1.	Parámetros del motor	- 35 -
6.2.	Masas.....	- 35 -
6.3.	Distancias	- 35 -
6.4.	Momento de inercia y coeficiente de rozamiento del volante.....	- 36 -
6.5.	Momento de inercia y coeficiente de rozamiento del péndulo.....	- 39 -
Capítulo 7:	Diseño del control.....	- 43 -
7.1.	Control de balanceo.....	- 43 -
7.2.	Control de salto	- 44 -
7.3.	Puesta en conjunto.....	- 45 -
Capítulo 8:	Implementación en Simulink.....	- 47 -
8.1.	Simulaciones	- 47 -
8.2.	Bloques Waijung.....	- 48 -
8.3.	Pruebas en sistema real	- 52 -
8.3.1.	Servomotor	- 52 -
8.3.2.	Motor DC.....	- 54 -
8.3.3.	Sensores (IMU).....	- 58 -
8.3.4.	Máquina de estados.....	- 61 -
8.4.	Prueba del control LQR.....	- 65 -
Capítulo 9:	Resultados.....	- 69 -
9.1.	Resultados de las simulaciones	- 69 -

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

ÍNDICES

9.1.1. Equilibrio desde ángulo inicial pequeño	- 69 -
9.1.2. Equilibrio en recepción de salto	- 74 -
9.1.3. Aumento del par motor.....	- 77 -
9.2. Resultados de las pruebas.....	- 79 -
9.2.1. Servomotor	- 81 -
9.2.2. Motor DC.....	- 82 -
9.2.3. Sensores (IMU).....	- 84 -
9.2.4. Máquina de estados.....	- 90 -
9.3. Resultados del control sobre el sistema real.....	- 91 -
Capítulo 10: Conclusiones.....	- 95 -
Bibliografía.....	- 97 -
Anexo A: Código fuente.....	- 99 -
Código del diseño del control LQR	- 99 -
Identificación parámetros péndulo.....	- 101 -
Ejemplo de tratamiento de datos	- 105 -
Anexo B: Datasheets	- 107 -
Parte 2: Planos	- 109 -
Capítulo 1: Planos de piezas.....	- 111 -
1.1. Acoplamiento del motor al volante de inercia.....	- 111 -
1.2. Apoyo del motor.....	- 112 -
1.3. Cara del cubo.....	- 113 -
1.4. Sujeción del servomotor.....	- 114 -
1.5. Volante de inercia.....	- 115 -
Capítulo 2: Esquemas electrónicos.....	- 117 -
2.1. Conexionado IMU y servomotor	- 117 -
2.2. Conexionado ESCON Module 24/2.....	- 118 -
2.3. Conexionado Motor EC	- 119 -
Parte 3: Presupuesto.....	- 121 -

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

Capítulo 1: Mediciones	- 123 -
1.1. Componentes principales	- 123 -
1.2. Componentes del montaje	- 123 -
1.3. Equipo y herramientas.....	- 124 -
1.4. Software	- 124 -
1.5. Mano de obra.....	- 124 -
Capítulo 2: Precios unitarios	- 125 -
2.1. Componentes principales	- 125 -
2.2. Componentes del montaje.....	- 125 -
2.3. Equipo y herramientas	- 126 -
2.4. Software	- 126 -
2.5. Mano de obra.....	- 126 -
Capítulo 3: Sumas parciales	- 127 -
3.1. Componentes principales	- 127 -
3.2. Componentes del montaje	- 127 -
3.3. Equipo y herramientas.....	- 128 -
3.4. Software	- 128 -
3.5. Mano de obra.....	- 129 -
Capítulo 4: Presupuesto general	- 131 -

Índice de figuras

Figura 1: Dibujo CAD del Cubli sin cubiertas	- 4 -
Figura 2: Péndulo de Furuta	- 5 -
Figura 3: Reaction Wheel Pendulum	- 6 -
Figura 4: Péndulo invertido subactuado	- 7 -
Figura 5: Péndulo sujeto a carrito	- 7 -
Figura 6: M-Block	- 8 -
Figura 7: Segway	- 8 -
Figura 8: ASIMO, robot humanoide [16]	- 9 -
Figura 9: Cara Péndulo Invertido	- 13 -
Figura 10: Volante de inercia	- 14 -
Figura 11: Maxon EC45 flat 12W	- 15 -
Figura 12: Sistema motor + volante de inercia	- 16 -
Figura 13: Apoyo del motor	- 17 -
Figura 14: Diseño de péndulo y volante de inercia sin sistema de frenado	- 17 -
Figura 15: Sistema de frenado	- 18 -
Figura 16: Sujeción servomotor	- 19 -
Figura 17: Diseño del montaje del péndulo invertido con volante de inercia (planta y alzado)-	20
-	
Figura 18: Vectorización y previsualización de fabricado de la cara del péndulo	- 22 -
Figura 19: Montaje final (planta y alzado)	- 23 -
Figura 20: Prototipo del péndulo invertido indicando ángulos y distancias [3]	- 26 -
Figura 21: Estimación del ángulo [3]	- 28 -
Figura 22: Comunicación serie síncrona	- 31 -
Figura 23: Comunicación serie asíncrona	- 32 -
Figura 24: I2C maestro-esclavo	- 32 -
Figura 25: Ejemplo de comunicación I2C	- 33 -
Figura 26: Identificación del centro de gravedad	- 36 -
Figura 27: Serie temporal de corriente (naranja) y velocidad en rpm (azul)	- 37 -

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

Figura 28: Serie temporal de corriente (naranja) y velocidad en rad/s (azul) filtrados	38 -
Figura 29: Velocidad real (negro) vs velocidad estimada (verde)	38 -
Figura 30: Medidas del ángulo y velocidad angular sin filtrar y con un filtro de medias móviles -	39 -
Figura 31: Zonas de interés de ángulo y velocidad angular filtradas por medias móviles.....	40 -
Figura 32: θ_b (verde) vs ω_b (rojo) vs α_b (azul) después de filtro de tercer orden	41 -
Figura 33: Máquina de estados de funcionamiento del Cubli	45 -
Figura 34: Simulador sistema no lineal.....	47 -
Figura 35: Target Setup (bloque Waijung).....	49 -
Figura 36: Configuración de comunicaciones (bloque Waijung).....	49 -
Figura 37: Convertidor AD (bloque Waijung).....	50 -
Figura 38: Convertidor DA (bloque Waijung).....	50 -
Figura 39: I2C leer/escribir (bloque Waijung).....	51 -
Figura 40: Entrada digital (bloque Waijung)	51 -
Figura 41: Salida digital (bloque Waijung).....	51 -
Figura 42: Salida PWM (bloque Waijung).....	52 -
Figura 43: Transmisión UART (bloque Waijung)	52 -
Figura 44: Funcionamiento servomotor	53 -
Figura 45: Prueba del servomotor y pulsador	53 -
Figura 46: ESCON Studio (Monitor de control).....	55 -
Figura 47: Configuración del offset de la velocidad del motor.....	56 -
Figura 48: Temporizador de calibración de la lectura de la ω_w	57 -
Figura 49: Calibración de la ω_w	57 -
Figura 50: Diagrama de Simulink para la comparación referencia velocidad angular del disco con la velocidad real	58 -
Figura 51: Prueba de IMU (Who Am I).....	59 -
Figura 52: Prueba de IMU (lectura de ángulo).....	60 -
Figura 53: Prueba de IMU (lectura de ángulo y velocidad angular)	61 -
Figura 54: Implantación de máquina de estados	62 -
Figura 55: Implantación de la lectura de entradas.....	63 -
Figura 56: Implementacion del subsistema de salidas	64 -

Figura 57: Implementación del sistema total	- 65 -
Figura 58: Lecturas del control LQR	- 66 -
Figura 59: Salidas del control LQR.....	- 66 -
Figura 60: Resultado de la simulación utilizando mismo peso para las variables de estado y mando	- 69 -
Figura 61: Resultado de la simulación dando peso 1000 a θ_b y a ω_b	- 70 -
Figura 62: Resultado de la simulación dando peso 1000 a θ_b y a ω_b y peso 50 a la I.....	- 70 -
Figura 63: Resultado de la simulación con peso 1000 en θ_b y ω_w y peso 50 en la I.....	- 71 -
Figura 64: Resultado de la simulación con peso 1000 en θ_b , ω_w e I	- 71 -
Figura 65: Resultado de simulación eliminando el peso de ω_w	- 72 -
Figura 66: Resultado de simulación para máximo ángulo con la matriz Klqr6	- 73 -
Figura 67: Resultado de la simulación para el ángulo máximo del control 5	- 74 -
Figura 68: Resultado de simulación de salto para Klqr6 (izquierda $\omega_b = -1.4$ rad/s, derecha $\omega_b = -0.74$)	- 75 -
Figura 69: Resultado de simulación de salto para Klqr5 (izquierda $\omega_b = -1.4$ rad/s, derecha $\omega_b = -0.74$).....	- 75 -
Figura 70: Comparación del efecto de la modificación del peso del control sobre el ángulo...-	76 -
Figura 71: Comparación del efecto de la modificación del peso del control a la velocidad angular del péndulo	- 76 -
Figura 72: Comparación del efecto de la modificación del peso del control a la velocidad angular del disco	- 77 -
Figura 73: Comparación del efecto de la modificación del peso del control a la corriente del motor	- 77 -
Figura 74: Resultados de simulación aumentando la constante del par motor respecto a la del control número 6	- 78 -
Figura 75: Resultados de la simulación con el par aumentado maximizando el ángulo hasta 6.28°	- 79 -
Figura 76: Lectura de variables de interés mediante la comunicación a través de UART	- 80 -
Figura 77: Posición de reposo (izquierda) y de frenado (derecha) del servomotor.....	- 82 -
Figura 78: Calibración del offset del motor	- 83 -
Figura 79: Comparación de la referencia de velocidad angular y la velocidad angular real del motor DC (ω_w).....	- 83 -

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

Figura 80: Lectura de la prueba de "Who Am I" de las IMU.....- 85 -

Figura 81: Lectura de ángulo 1 (θ_b).....- 86 -

Figura 82: Lectura de ángulo 2 (θ_b).....- 86 -

Figura 83: Lectura de ángulo 3 (θ_b).....- 87 -

Figura 84: Gráfico de las lecturas de los ángulos filtrando el ruido de la UART- 88 -

Figura 85: Superposición de las lecturas de los dos giróscopos.....- 89 -

Figura 86: Comprobación del desfase entre las lecturas del ángulo y la velocidad angular ...- 90 -

Figura 87: Brazos de servomotor inservibles- 91 -

Figura 88: Resultados del control en el sistema real- 92 -

Figura 89: Resultados del control en el sistema real filtrando la lectura del ángulo antes del control
.....- 93 -

Índice de ecuaciones

<i>Ecuación 1: Ecuación general de Lagrange</i>	25 -
<i>Ecuación 2: Ecuación general de Euler-Lagrange</i>	25 -
<i>Ecuación 3: Lagrangiano del sistema</i>	25 -
<i>Ecuación 4: Derivada parcial del lagrangiano respecto ω_b</i>	26 -
<i>Ecuación 5: Derivada parcial del lagrangiano respecto ω_w</i>	26 -
<i>Ecuación 6: Derivada del lagrangiano respecto θ_b</i>	27 -
<i>Ecuación 7: Derivada del lagrangiano respecto θ_w</i>	27 -
<i>Ecuación 8: Ecuación del par motor</i>	27 -
<i>Ecuación 9: Rozamiento del péndulo sobre su eje</i>	27 -
<i>Ecuación 10: Rozamiento del disco sobre su eje</i>	27 -
<i>Ecuación 11: Ecuación no lineal de las dinámicas del péndulo</i>	27 -
<i>Ecuación 12: Ecuación no lineal de las dinámicas del volante de inercia</i>	27 -
<i>Ecuación 13: Aceleraciones del acelerómetro respecto de sus ejes</i>	29 -
<i>Ecuación 14: Eliminación de los términos dinámicos de los acelerómetros</i>	29 -
<i>Ecuación 15: Estimación de θ_b</i>	29 -
<i>Ecuación 16: Estimación de ω_b</i>	29 -
<i>Ecuación 17: Estimación de ω_w</i>	29 -
<i>Ecuación 18: Respuesta temporal del disco a impulsos de corriente</i>	36 -
<i>Ecuación 19: Función de transferencia de la velocidad del disco con la corriente</i> ..	37 -
<i>Ecuación 20: Respuesta temporal a la caída libre del péndulo</i>	39 -
<i>Ecuación 21: Ecuación de coste de un sistema en tiempo continuo</i>	43 -
<i>Ecuación 22: Ecuaciones de estado en tiempo continuo</i>	43 -
<i>Ecuación 23: Solución de la matriz de ganancias del control LQR</i>	44 -
<i>Ecuación 24: Ecuación de Riccati</i>	44 -
<i>Ecuación 25: Conversión a voltios de la referencia de velocidad del volante</i>	56 -
<i>Ecuación 26: Conversión a rpm de la velocidad del volante</i>	56 -
<i>Ecuación 27: Conversión a rpm de la velocidad del volante ajustada con offsets</i>	56 -
<i>Ecuación 28: Conversión de corriente a voltios</i>	67 -

Parte 1: Memoria

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 1: Introducción

En los últimos años los dispositivos electrónicos se han ido adentrando cada vez más en nuestras vidas y por lo tanto nuevas especialidades de la electrónica han comenzado a aparecer. Debido al gran aumento de este tipo de dispositivos, hay dos ramas de la electrónica en las cuales ha aumentado el interés general, y estas son la electrónica de control y la robótica.

La electrónica de control es la rama de la electrónica que se especializa en la manipulación de las señales de entrada en un sistema para obtener las salidas deseadas, salvando las perturbaciones que puedan interferir en la respuesta. Mientras que la robótica es una rama de la ingeniería electrónica que va unida a la construcción de artefactos que intentan imitar al ser humano y realizar más eficientemente sus tareas. Estas dos especialidades están muy ligadas, ya que la robótica utiliza las técnicas de control para manejar y automatizar los robots.

Uno de los mecanismos más utilizados en los últimos años para investigar sobre estos temas es el péndulo invertido. Esto se debe a que es un tipo de mecanismo de los denominados sistemas mecánicos subactuados [1], cuyo gran interés se debe en gran parte a que pueden ofrecer un considerable ahorro de energía y de esfuerzos de control respecto a los sistemas completamente actuados. Esto se consigue gracias a que tienen menos actuadores que grados de libertad y por lo tanto se simplifican los controles.

Este proyecto trata diseñar los primeros pasos de un cubo que pueda saltar y equilibrarse sobre una arista o vértice mediante el diseño de un sistema de control de una de las caras del cubo. La actuación, en este caso, se realiza modificando el momento angular de un volante de inercia acoplado a dicha cara utilizando control digital.

1.1. Estado del arte

Sobre los péndulos invertidos se han realizado ya numerosas investigaciones ya que son un excelente punto de referencia para muchos problemas de control [2], tales como:

- Modelado
- Identificación de parámetros
- Control de posición y velocidad
- Control no lineal
- Compensación por fricción
- Análisis del ciclo límite

JULIO 2016

- Control híbrido

Existen dos versiones de péndulo invertido, dependiendo del mecanismo de actuación que se utilice para su movimiento:

- Actuador giratorio
- Actuador lineal

El proyecto que se va a realizar está basado en los péndulos con un actuador giratorio. Se ancla una esquina de una cara de un cubo y se acopla un disco a la misma permitiendo solamente este tipo de movimiento.

Previamente se han realizado ya experimentos sobre este tipo de mecanismos, algunos de los cuales se presentan a continuación.

1.1.1 Sistema de Péndulos

1.1.1.1. “Cubli”

El sistema que se va a diseñar está basado en un proyecto desarrollado por el ETH de Zurich llamado “Cubli” [3]-[5]. El “Cubli” es un cubo que puede saltar y balancearse sobre una esquina mediante la aplicación de momentos en tres volantes de inercia montados en tres caras del cubo. Esto se consigue girando la rueda a altas velocidades y frenándola bruscamente logrando que el “Cubli” salte. En los artículos de referencia se muestra el uso de controles tanto lineales como no lineales, además de la estrategia de análisis del sistema dinámico no lineal. El objetivo del proyecto presentado en esta memoria es conseguir imitar esa idea únicamente en una cara de un cubo.

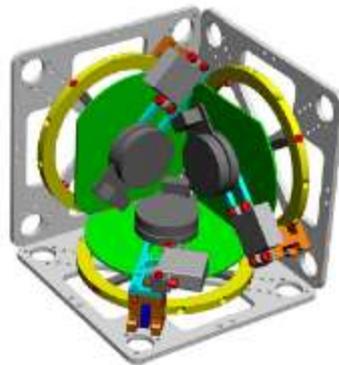


Figura 1: Dibujo CAD del Cubli sin cubiertas

1.1.1.2. “Furuta’s Pendulum”

Un punto de referencia internacional sobre los péndulos invertidos para la comunidad del control automático en los últimos años es el “péndulo de Furuta” [6]. El péndulo de Furuta ofrece una mejor comprensión de las Técnicas de Control Lineal. Puesto que la mayoría de estas técnicas están basadas en los invariantes del movimiento, el modelo dinámico juega un rol muy importante. El artículo describe en detalle el modelo dinámico desarrollado en el laboratorio. Su éxito se debe a un modelo dinámico básico derivado de la mecánica clásica, el cual se modifica para compensar los momentos no conservativos. De esta manera, el modelo práctico cuasi-conservativo permite diseñar los controles digitales como si el sistema fuera estrictamente conservativo. También se muestra los resultados del estudio de diferentes controles no lineales en este tipo de sistemas, como por ejemplo el LQR [7].

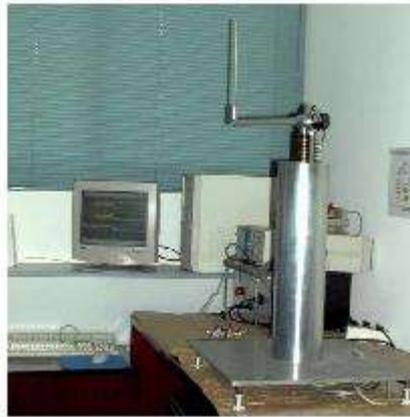


Figura 2: Péndulo de Furuta

1.1.1.3. “Reaction Wheel Pendulum”

Este proyecto consiste en un péndulo mecánico en cuyo extremo se engancha un peso rotativo [8]. Desde un punto de vista pedagógico, es un sistema con unas dinámicas muy simples para un sistema no lineal. Se utiliza para investigar técnicas de control como la realimentación de estado y los controles basados en pasividad [9]. Se plantea el problema desde un punto de vista de conservación de energías y se utiliza un sistema de control híbrido que cambia en el momento de balancear, en este caso se utiliza un control lineal, y el de mover el péndulo, que consistirá en el uso de un control no lineal.

JULIO 2016



Figura 3: Reaction Wheel Pendulum

1.1.1.4. “Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom”

Lo que tratan de conseguir en este proyecto [10] es desarrollar unos algoritmos de estimación de la inclinación de un cuerpo rígido sobre un punto de anclaje utilizando múltiples acelerómetros. La estimación es independiente de las dinámicas del objeto. Este método es aplicable en condiciones estáticas y dinámicas independientemente. Lo único que se necesita conocer es la posición de los sensores en el espacio. En el artículo se muestran resultados experimentales sobre un cubo.

1.1.1.5. “Controlling an inverted pendulum with bounded controls”

Se analiza el comportamiento de un sistema mecánico subactuado simple utilizando leyes de control continuo [11]. El sistema consiste en un péndulo con un disco de inercia sujeto en el extremo libre. Se aplican momentos al disco de inercia utilizando un motor de corriente continua. El sistema en lazo cerrado muestra una dinámica compleja ante la variación de parámetros de control. Se consigue estabilizar el péndulo en posición vertical con velocidad cero si la matriz de ganancias del control es elegida correctamente. En este caso se aplica una estrategia de control híbrido que consiste en el uso de controles lineales y no lineales dependiendo del momento de utilización.

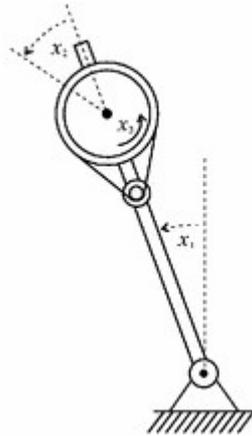


Figura 4: Péndulo invertido subactuado

1.1.1.6. “Standup and Stabilization of the Inverted Pendulum”

Esta tesis investiga la rutina de levantamiento y estabilización de un péndulo invertido mediante el uso de un carrito sobre el cual está sujeto el péndulo [12]. El péndulo se levanta moviendo estratégicamente el carrito de tal manera que se da suficiente energía para que el péndulo alcance una posición vertical. La estabilización se consigue utilizando un control de realimentación de estado lineal. Estos métodos y algoritmos se implementan utilizando herramientas de Matlab/ Simulink.

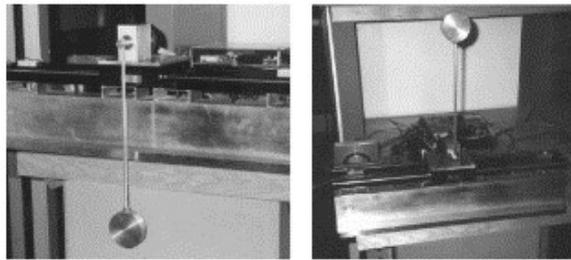


Figura 5: Péndulo sujeto a carrito

1.1.2. Proyectos basados en péndulos invertidos

1.1.2.1. “M-Blocks”

El MIT ha creado un prototipo de robots modulares [13] que son capaces de auto-ensamblarse. Los robots son cubos capaces de rodar y hasta subirse unos encima de otros adquiriendo la forma que deseen. Esto lo consiguen gracias a que en su interior tienen unos discos de inercias que al ser acelerados y frenados bruscamente provocan un momento de inercia al cubo

JULIO 2016

que le permite moverse en cualquier dirección rotando sobre sus aristas. Este sistema funciona con el mismo principio que el “Cubli“, ya mencionado anteriormente.



Figura 6: M-Block

1.1.2.2. “Segway”

El “Segway” [14] es un vehículo de transporte eléctrico de dos ruedas con auto-equilibrio que usa una tecnología patentada de estabilización dinámica. Esta tecnología llamada “Smart Motion™” consiste en una red inteligente de sensores, componentes mecánicos y sistemas de control, que permiten al “Segway” mantenerse en equilibrio y desplazarse sobre dos ruedas. En el momento en que una persona se sube, cinco giroscopios y dos sensores de inclinación detectan los cambios en el terreno y la posición del cuerpo a una velocidad de 100 veces por segundo.

El funcionamiento es similar al de un péndulo invertido anclado sobre un carrito, está basado en movimiento lineal y por lo tanto utiliza técnicas de control similares.



Figura 7: Segway

1.1.2.3. *Humanoides*

Los humanoides son robots cuya estructura intenta imitar la del ser humano. El sistema de movimiento del robot y de mantenerse de pie en términos dinámicos se interpreta como si fuera un péndulo invertido apoyado sobre el suelo. Esto significa que el sistema de control [15] que tiene el robot para mantener una posición estable sobre su centro de gravedad es similar a la de un péndulo invertido.

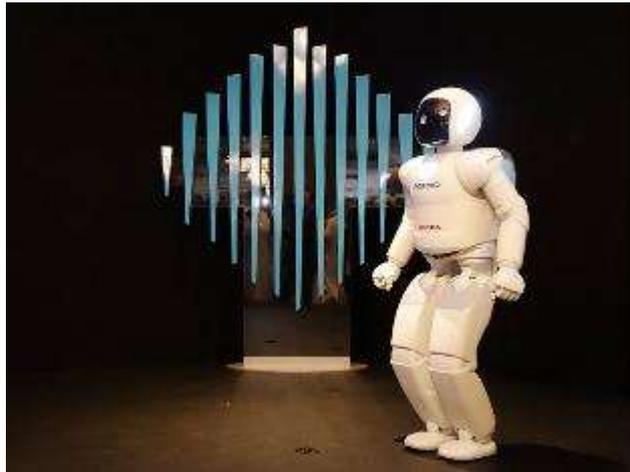


Figura 8: ASIMO, robot humanoide [16]

1.2. *Motivación*

Aunque a primera vista, el sistema de péndulo invertido puede parecer un mecanismo que no tiene mucha utilidad, a pesar de su simplicidad ofrece una gran variedad de aplicaciones desde la investigación hasta aplicaciones de robótica, transporte y aeroespacial.

A nivel de investigación, permite analizar diferentes técnicas de control lineal o no lineal dependiendo del mecanismo que se utilice de actuador en el sistema o el tipo de tarea que queremos ejecutar, ya sea lineal o giratorio, o estabilización o movimiento respectivamente.

Como ya se ha mencionado previamente, estas técnicas de control tienen múltiples aplicaciones como por ejemplo los robots que se autoensamblan formando sistemas más complejos, equilibrar medios de transporte sobre dos ruedas o incluso ayudar al sistema de movimiento de robots bípedos. No obstante tiene mucho interés seguir investigando sobre ello y sobre su posible utilidad para infinidad de aplicaciones más como equilibrar un cohete en su despegue, ayudar a personas con equilibrio, o incluso utilizar algún tipo de volante de inercia en edificios en zonas de alto riesgo de actividad sísmica para que no se caigan al suelo.

JULIO 2016

Es por ello por lo que se propone en este proyecto desarrollar el control de un péndulo invertido mediante la variación angular y así contribuir a la investigación y la mejora del conocimiento de este tipo de sistemas.

1.3. Objetivos

Los objetivos principales del proyecto son:

1. Diseñar y construir un péndulo invertido el cual tenga unido un disco de inercia, y enganches para motor de corriente continua y servomotor.
2. Diseñar un modelo digital del sistema para crear un simulador y así poder realizar pruebas de diferentes sistemas de control.
3. Diseñar un control que consiga estabilizar la cara del sistema sobre una esquina e implantarlo en el sistema real.
4. Conseguir que el sistema salte desde la posición de reposo en la arista de la cara a la esquina y mantenerse estabilizado sobre la misma.

1.4. Metodología

1.4.1. Tareas

Para cumplir los objetivos del proyecto habrá que realizar las siguientes tareas:

- Diseño de las piezas necesarias en Solid Edge.
- Impresión en 3D de piezas.
- Mecanizado en aluminio de piezas.
- Montaje del sistema incluyendo todos los componentes electrónicos.
- Análisis del sistema dinámico del prototipo e identificación de parámetros
- Creación de la planta del sistema para simular en Matlab utilizando los modelos digitales en 3D.
- Diseño de controles que funcionen en la simulación en Simulink.
- Ensayo de controles en la simulación.
- Ensayo y ajuste de parámetros de control en sistema real.
- Redacción de la memoria

1.5. Recursos

Las herramientas que se emplearán para la realización del proyecto incluirán el siguiente equipamiento de software y hardware:

1.5.1. Software

- Solid Edge ST7: Programa de diseño en 3D donde se diseñará el sistema.
- Matlab: Herramienta de software matemático en el cual se diseñarán los controles del sistema.
- Simulink: Módulo de simulación de Matlab que se utilizará para realizar simulaciones antes de probar el sistema real.
- ESCON Studio: Programa de configuración de controladora de motores Maxon.

1.5.2. Hardware

- Motor de corriente continua Maxon EC45 flat 12W que servirá para acelerar el volante de inercia.
- Controladora de motor de corriente continua ESCON Module 24/2
- Servomotor MG 90S que actuará como sistema de frenado.
- Microcontrolador STM32F407 Discovery.
- IMU (Unidad de Medición Inercial) para medir la orientación del sistema.

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 2: Diseño de péndulo invertido con volante de inercia

2.1. Diseño general del péndulo invertido

El diseño del péndulo invertido, ya que más adelante el objetivo sería convertirlo en un cubo que se balanceara sobre sus aristas y esquinas, tiene que ser cuadrado. El diseño se intenta que además se pueda aprovechar para al fabricar seis caras iguales que se puedan unir y de esta manera ya se tendría el cubo. Lo primero que hay que tener en cuenta es que tiene que ser ligero. Para conseguir minimizar el peso se fabricará en aluminio de 2mm de grosor, y de 15x15cm. Sobre la cara se realiza vaciados para utilizar el mínimo material necesario para poder mantener todos los componentes correctamente fijados sin que sean un riesgo para la salud en caso de piezas sueltas.

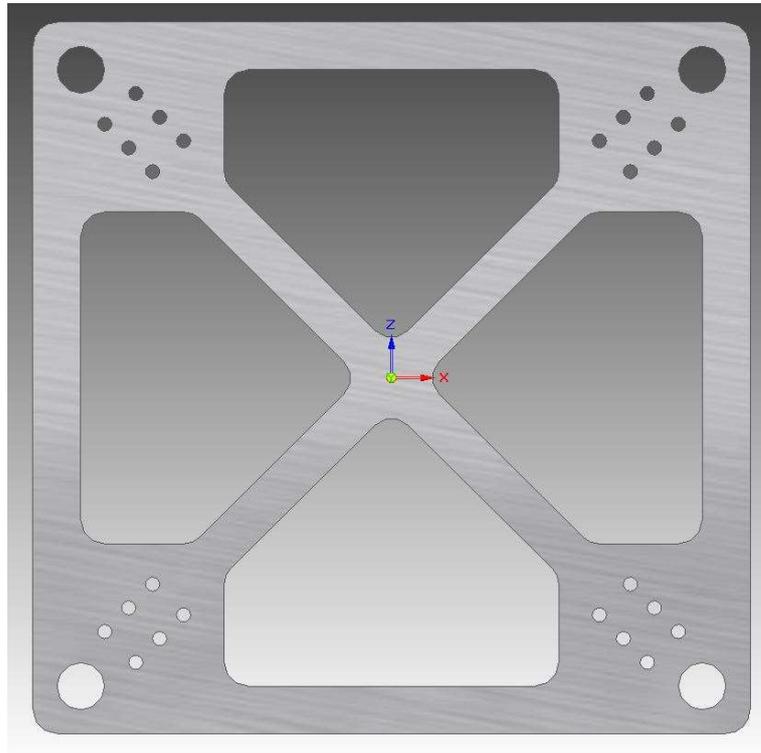


Figura 9: Cara Péndulo Invertido

2.2. Motor y volante de inercia

El eje del motor y el volante de inercia son las piezas, cuya masa en rotación, van a proveer la inercia al sistema para poder saltar, balancearse y equilibrarse sobre el ángulo que se requiera. Para poder traspasar la máxima cantidad de energía de rotación a energía cinética y potencial y poder mover el péndulo más fácilmente, se tienen que intentar cumplir los siguientes requisitos:

1. Máximas revoluciones por minuto.
2. El peso debe estar lo más alejado del centro de rotación manteniendo el centro de gravedad lo más centrado posible.
3. El volante de inercia debe pesar lo mismo o más que el cuerpo del péndulo.

Para ello, ya que diseñar la pieza de acero no es posible con los recursos disponibles, se utiliza un disco de freno de minimoto. Utilizando un disco de freno, ya que es para motos de competición, se asegura que va a estar correctamente equilibrado además de tener la mayoría de su peso en el exterior. Otra ventaja de utilizar un disco de freno, es que las ranuras que tiene en su exterior, se pueden aprovechar para montar parte del sistema de frenado utilizado para la acción del salto. Esto se explica más adelante.

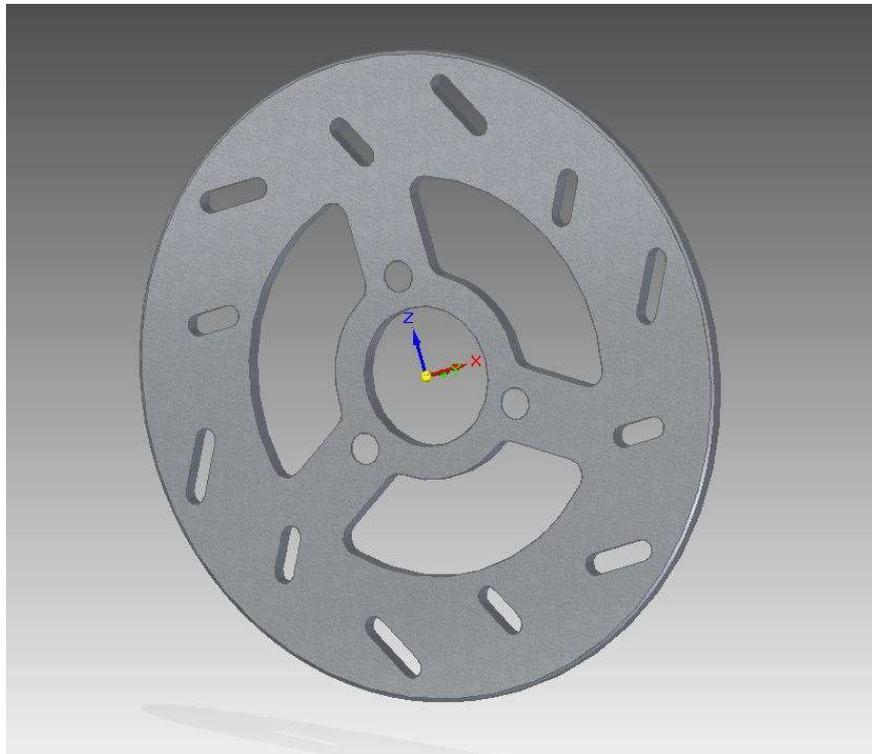


Figura 10: Volante de inercia

El motor es un “brushless” de corriente continua (Maxon EC45 flat 12W) cuya velocidad máxima es de aproximadamente 10000 rpm. En un principio, el motor puede parecer que tiene poca potencia, pero lo importante para realizar el salto es que consiga que el volante de inercia gire a la velocidad suficiente, aunque acelere más despacio. Aun así, la potencia del motor podría ser útil para ser capaz de acelerar más rápido para la acción de balanceo, ya que si no alcanza la velocidad adecuada antes de cruzar el ángulo límite, se caerá y no será capaz de equilibrarse.

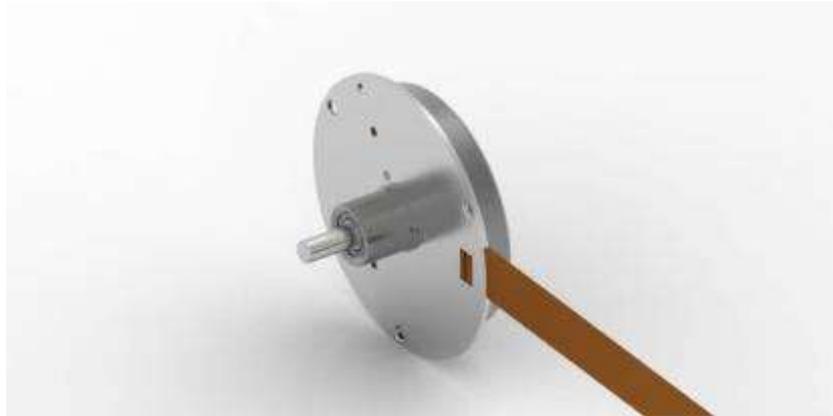


Figura 11: Maxon EC45 flat 12W

Para poder sujetar el eje del motor al disco de freno es necesario fabricar dos piezas que funcionen de adaptador. Esto se debe a que el eje del motor es muy fino, y los agujeros del disco de freno para acoplarse al motor tienen un radio mayor.

Primero se diseña un pequeño disco de aluminio en el cual se realizan tres taladros que coincidan con los agujeros para los tornillos del disco de freno, y otros 3 taladros para acoplarlo a la segunda pieza.

La segunda pieza es un acoplamiento de tipo “Oldham”. Estos acoplamientos se utilizan para transmitir rotación de un motor a otro eje. Para poder aprovecharlo, se tienen que primero eliminar las pestañas de acoplamiento habitual, y realizar 3 taladros para ser acoplado al disco de aluminio.

Con el motor, el acoplamiento tipo “Oldham”, el disco de aluminio y el disco de freno de minimoto se obtiene el sistema de volante de inercia que funcionará de actuador para realizar el salto y equilibrio del péndulo invertido.

JULIO 2016

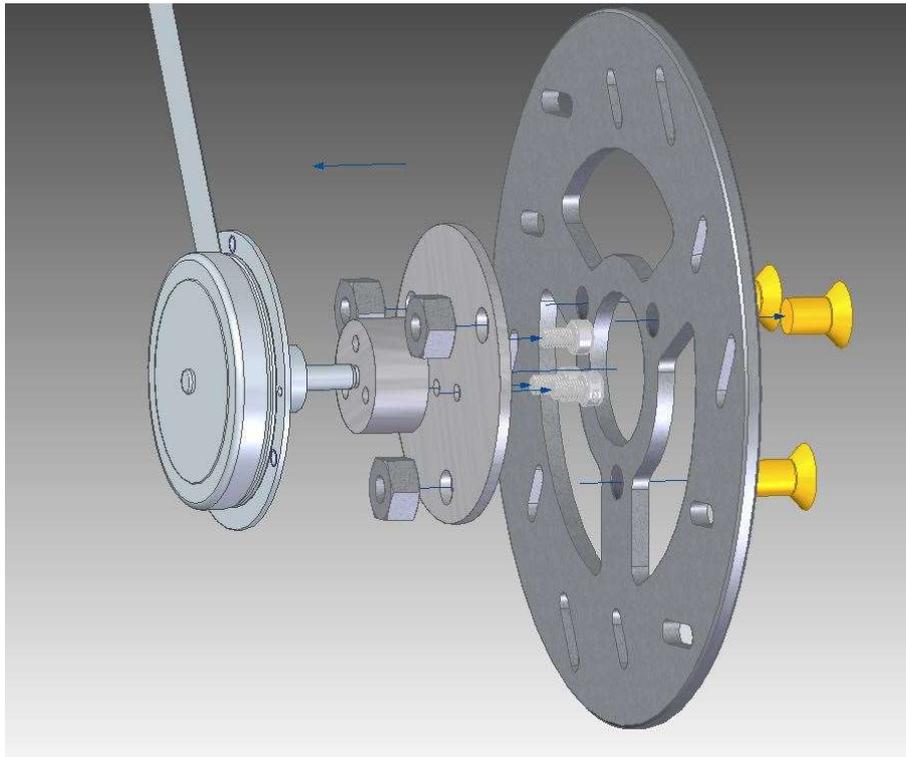


Figura 12: Sistema motor + volante de inercia

Llegados a este punto, ya sería posible equilibrar el péndulo sobre su esquina si la posición de inicio estuviera dentro del rango de balanceo. Solo falta la pieza que une la cara del péndulo con el volante de inercia y el motor.

La pieza de apoyo del motor, se diseña de tal manera que su ubicación sea encima de la diagonal de la cara del péndulo. Esta pieza, tiene que ser capaz de sujetar el motor y el volante de inercia, además de intentar proteger al usuario. Para cumplir estos requisitos, se hace un agujero en el medio lo suficientemente grande como para que el eje del motor pase, y además se taladran unos agujeros pequeños ya pensando en la sujeción del sistema de frenado. Esto obliga a que el volante de inercia quede situado entre la cara y la pieza de apoyo del motor, haciendo que en caso de que el disco se soltara, solamente tuviera una posible dirección de escape.

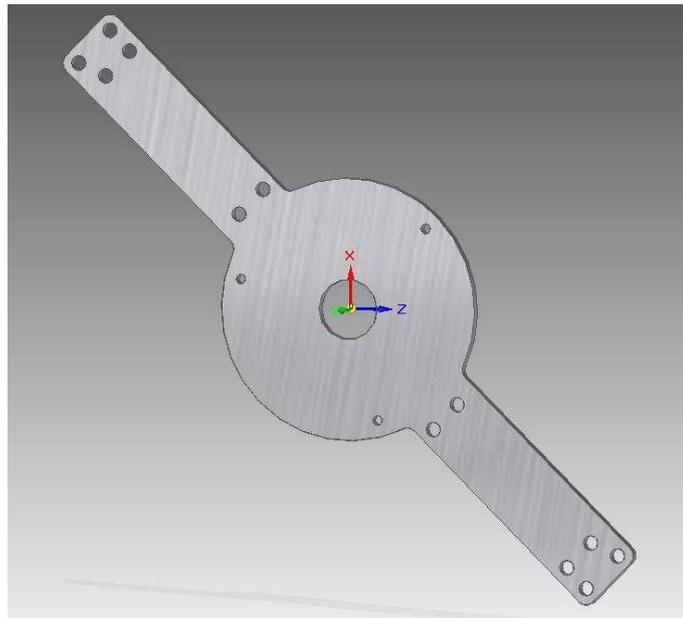


Figura 13: Apoyo del motor

Para unir la cara con la pieza de apoyo del motor, se utilizan separadores de placas electrónicas de aluminio, para minimizar su peso pero no comprometer el agarre y la integridad estructural del sistema.

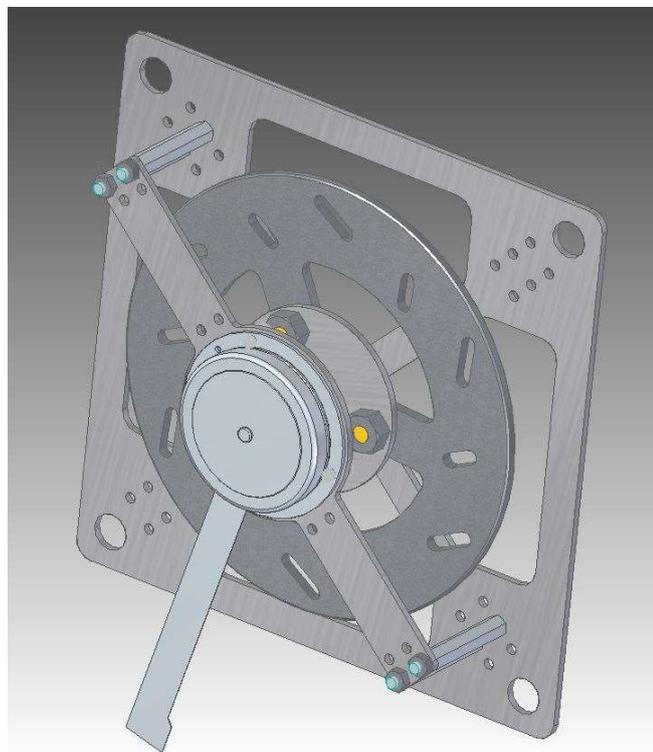


Figura 14: Diseño de péndulo y volante de inercia sin sistema de frenado

JULIO 2016

2.3. Sistema de frenado

En un principio se plantean 2 posibles sistemas de frenado, mediante pastillas de freno similar a los coches, o mediante un sistema de palanca que choque contra un tope frenando en seco.

El uso de pastillas freno parece una opción obvia ya que se utiliza un disco de freno como volante de inercia. El problema surge de la necesidad de una deceleración muy elevada. Para poder utilizar pastillas de freno haría falta aplicar mucha energía para poder presionar con suficiente fuerza el disco como para ser capaz de convertir suficiente energía rotacional en potencial para realizar el salto.

Por esta razón, el sistema de freno usado es mediante un golpe seco contra un tope. Los topes que se utilizan son 3 tornillos sujetos a las ranuras del exterior del volante de inercia. La palanca consiste en un servomotor que utiliza su brazo para frenar en seco el volante chocando contra uno de los 3 tornillos.

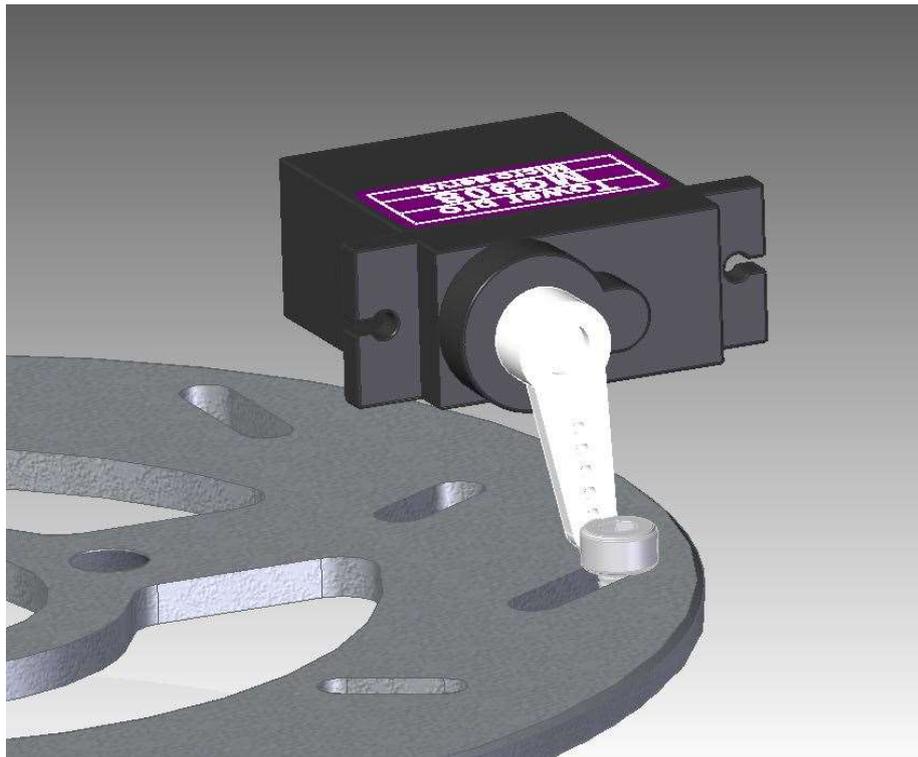


Figura 15: Sistema de frenado

Para sujetar el servo, se diseña una pequeña pieza de plástico que va atornillada a los agujeros restantes de la pieza de apoyo del motor. Esta pieza está diseñada para que el sentido del servo sea tal que cuando el brazo reciba el golpe, la fuerza vaya en la dirección del eje de los

engranajes del servo. De esta manera se consigue que los dientes de los engranajes no se rompan a los dos golpes.

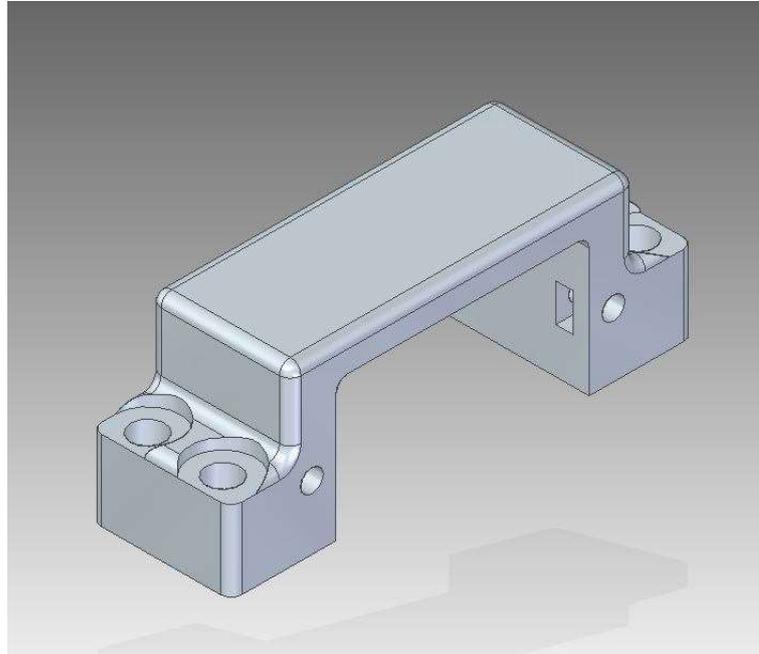


Figura 16: Sujeción servomotor

Debido a que el choque debe transmitir una elevada cantidad de energía al sistema, el lugar de la sujeción del servo donde van ubicados los tornillos se hace relativamente grueso. Esto se debe a que así se reparte la fuerza en una mayor superficie, y ya que la pieza será construida en plástico con una impresora 3D, no se rompe tan fácilmente.

2.4. Sistema total

Uniando todas las diferentes piezas se forma el conjunto del péndulo invertido con el volante de inercia. La siguiente figura muestra el resultado:

JULIO 2016

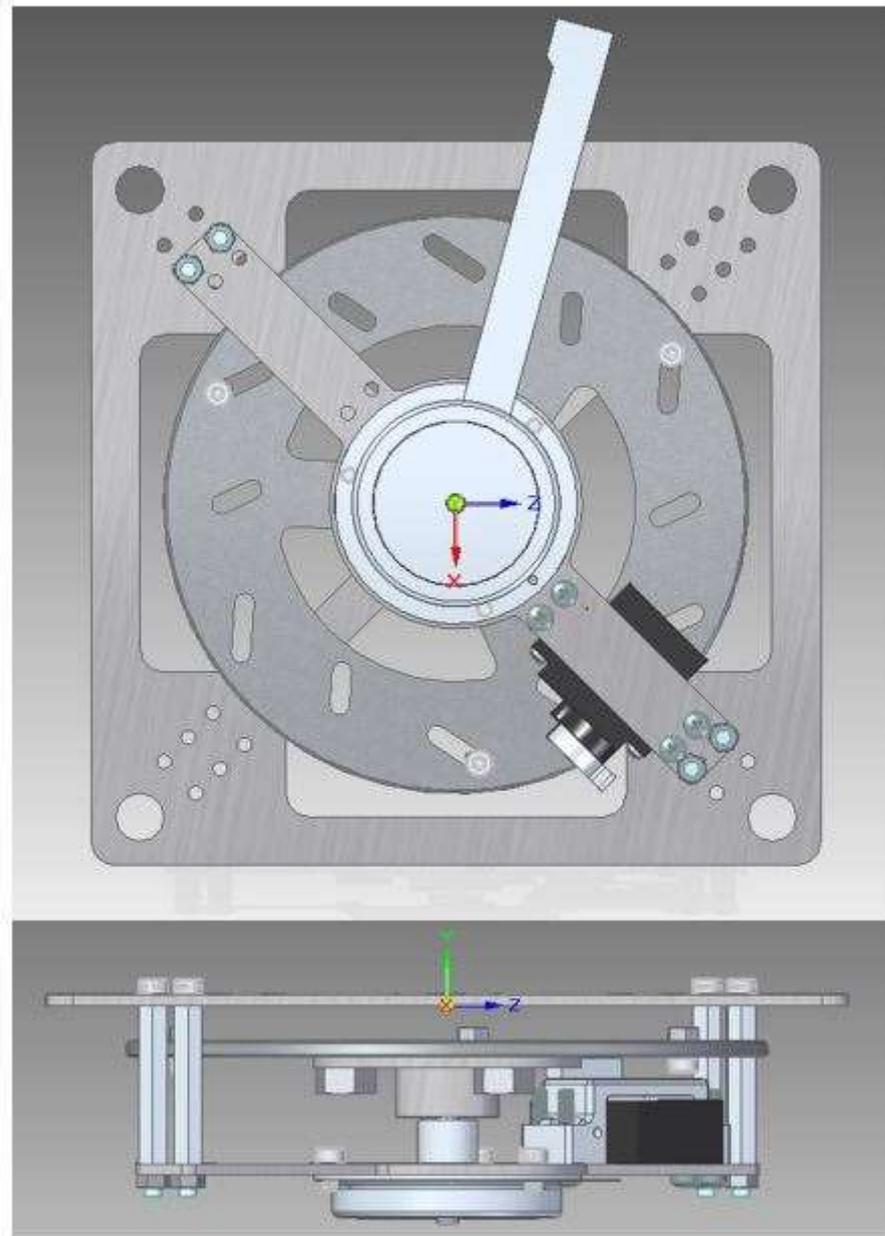


Figura 17: Diseño del montaje del péndulo invertido con volante de inercia (planta y alzado)

Capítulo 3: Fabricación de piezas

La forma de las piezas que se diseñan es en general bastante sencilla. Gracias a ello, se puede utilizar la fresadora automática para poder fabricarlas en aluminio con bastante facilidad. Solamente una de las piezas, la sujeción del servo, resulta más sencillo producirla en la impresora 3D.

3.1. Impresora 3D

Para imprimir la pieza en 3D, es necesario primero convertir el diseño en 3D del formato Solid Edge a formato “STereo Lithography”. Este formato está concebido para su uso en la industria del prototipado rápido y la fabricación asistida por ordenador, en especial para impresoras 3D de extrusión de plástico termofusible.

La única pieza que se fabrica de esta manera es la sujeción del servo. Esto es debido a que su forma es complicada de fabricar mediante otros métodos, y además, al ser de plástico, disminuye su peso.

3.2. Fresadora automática

La fresadora automática, es una herramienta para realizar trabajos mecanizados mediante arranque de viruta que se utiliza para fabricar piezas de una forma determinada a partir de una placa de un determinado material. Además de realizar los cortes necesarios, si se cambia la herramienta de fresado por una broca, realiza los agujeros necesarios.

La mayor ventaja de una fresadora automática es que utiliza control numérico computarizado para lograr movimientos imposibles de efectuar manualmente. Esto se traduce en una mayor rapidez y seguridad de fabricación de piezas. El CNC controla la posición y velocidad de los motores que accionan los ejes de la máquina. Para poder realizar estos movimientos complejos, es capaz de mover la herramienta en los tres ejes a la vez. Se denomina control numérico, debido a que las órdenes se indican mediante código de números, se dan las coordenadas iniciales, las finales, y como moverse de un lado a otro.

Para que el uso de maquinaria de este tipo sea posible, es necesario primero vectorizar los modelos de las piezas. Las que se fabrican mediante este método son la cara del péndulo, el

JULIO 2016

disco de aluminio que sirve de adaptador para el volante de inercia, y la pieza de apoyo del motor. Para vectorizar la pieza, se utiliza un programa llamado VCarve, y hay que seguir los siguientes pasos:

- Seleccionar tamaño de la placa sobre la que se fabrica la pieza (ancho=x, largo=y grueso=z).
- Elegir una esquina como coordenadas (x=0, y=0, z=0).
- Juntar todos los vectores abiertos.
- Programar los agujeros.
- Programar los cortes internos de la pieza.
- Programar los cortes externos de la pieza.
- Elegir herramienta.
- Guardar vectores.

Los vectores se guardan en formato NC. Este formato incluye las coordenadas iniciales y finales de cada operación y como moverse entre ellas. Los programas guardados de cada pieza se cargan en la fresadora, y esta se encarga de fabricarlas.

VCarve ofrece la opción de previsualizar el recorrido de la máquina y el resultado de la pieza como se muestra en la siguiente figura:

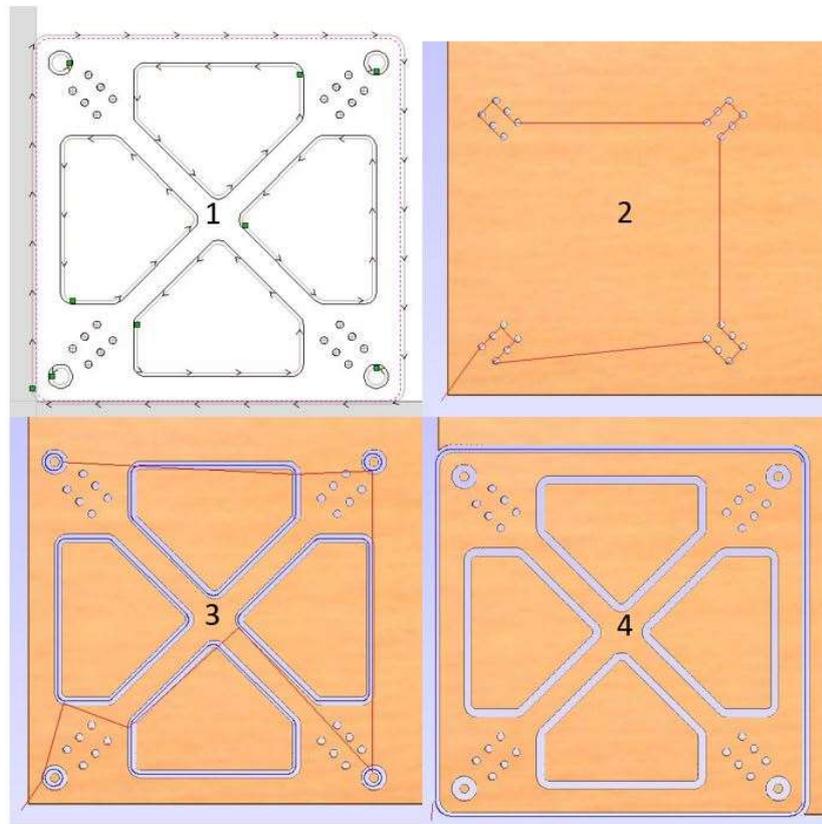


Figura 18: Vectorización y previsualización de fabricación de la cara del péndulo

Donde:

1. Visualización de todas las coordenadas vectorizadas.
2. Previsualización de recorrido para realizar agujeros.
3. Previsualización de recorrido para cortes internos.
4. Previsualización de recorrido para cortes externos.

Para asegurarse de que la fresadora realiza los cortes y agujeros correctamente es necesario programar la profundidad al menos medio milímetro mayor que el grosor de la plancha de aluminio.

El resultado de la fabricación de las piezas unidas formando el sistema péndulo invertido y volante de inercia se muestra en la siguiente figura:

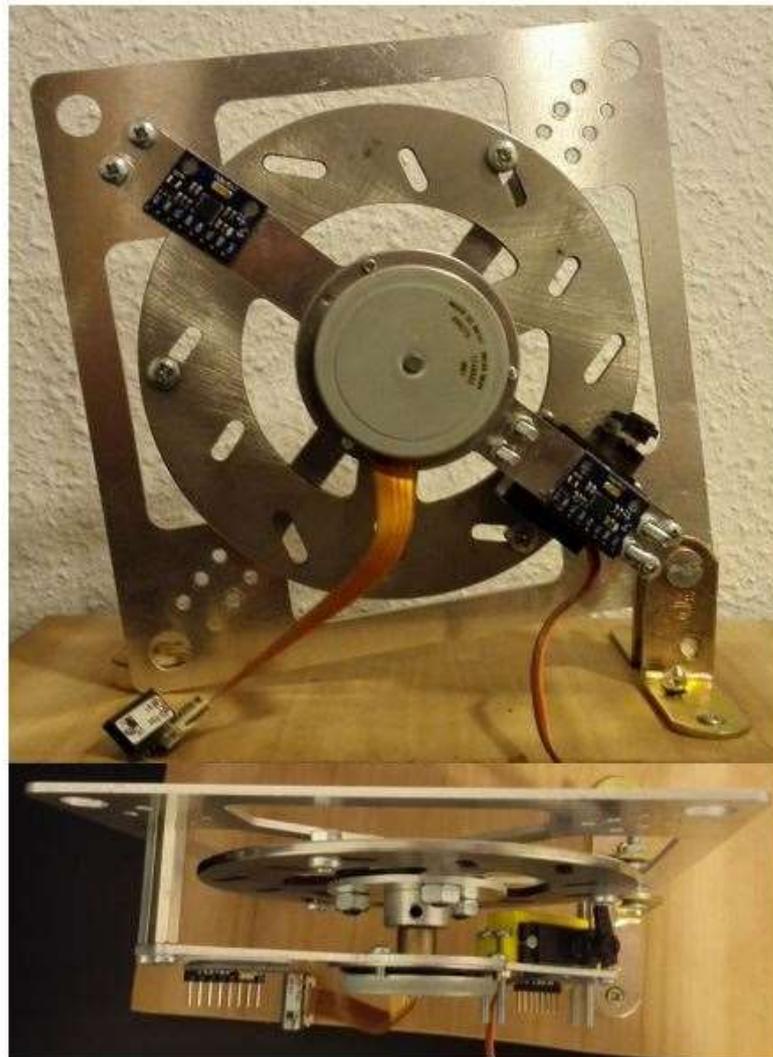


Figura 19: Montaje final (planta y alzado)

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 4: Modelado y dinámica del péndulo invertido

4.1. Dinámicas del sistema

Para el planteamiento de las ecuaciones dinámicas del sistema se hace uso de la mecánica lagrangiana [17]. La mecánica lagrangiana es ideal para sistemas con fuerzas conservativas en cualquier sistema de coordenadas, y se plantea en forma de energías potencial (V) y cinética (T) dando la siguiente ecuación:

$$L = T - V \quad (1)$$

Las fuerzas disipativas e impulsos se pueden tener en cuenta separando las fuerzas externas en la suma de fuerzas potenciales y no potenciales, lo cual lleva a las ecuaciones de Euler-Lagrange [18]. El planteamiento de estas ecuaciones se basa en el usado por el departamento de señales y sistemas de la “Chalmers University of Technology” para su proyecto de Cubli [19].

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} + \frac{\partial R}{\partial \dot{q}_k} = \tau_k \quad (2)$$

Donde τ_k es los pares de inercia y $\frac{\partial R}{\partial \dot{q}_k}$ modela las fuerzas disipativas.

Sustituyendo los parámetros de las energías potencial y cinética, el lagrangiano se modela como:

$$L = \frac{1}{2} I_b \dot{\theta}_b^2 + \frac{1}{2} I_w (\dot{\theta}_b + \dot{\theta}_w)^2 + \frac{1}{2} m_b (\dot{\theta}_b l)^2 - (l_b m_b + m_w l) g \cos \theta_b \quad (3)$$

Donde:

- θ_b = ángulo de la diagonal de la cara del péndulo respecto el eje vertical
- θ_w = ángulo del volante de inercia respecto la diagonal del péndulo

JULIO 2016

- m_w = masa del volante de inercia y rotor del motor
- m_b = masa del péndulo
- I_b = momento de inercia del péndulo sobre su eje de rotación
- I_w = momento de inercia del volante de inercia y rotor del motor sobre el eje del motor
- l = distancia del eje de rotación del motor al eje de rotación del péndulo
- l_b = distancia del eje de rotación del péndulo a su centro de gravedad
- g = aceleración de la fuerza de gravedad (9.81 ms^{-2})

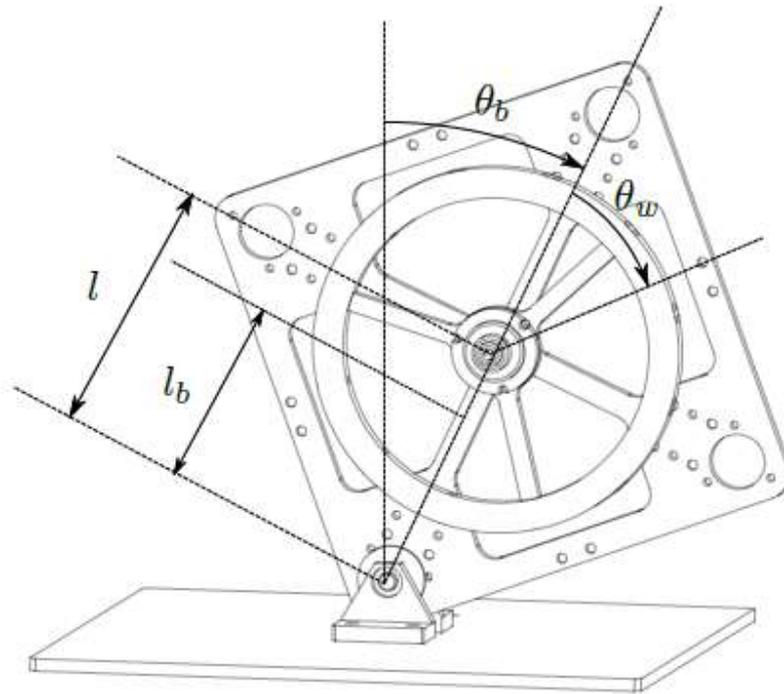


Figura 20: Prototipo del péndulo invertido indicando ángulos y distancias [3]

De las derivadas parciales del lagrangiano respecto las velocidades de giro se obtienen los siguientes resultados, los cuales son el primer término de la ecuación de Euler-Lagrange:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_b} \right) = I_b \ddot{\theta}_b + m_b l^2 \ddot{\theta}_b + I_w (\ddot{\theta}_b + \ddot{\theta}_w) \quad (4)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_w} \right) = I_w \ddot{\theta}_b + I_w \ddot{\theta}_w \quad (5)$$

Por otro lado, derivando el lagrangiano respecto de los ángulos de giro resulta en:

$$\frac{\partial L}{\partial \theta_b} = (l_b m_b + m_w l) g \sin \theta_b \quad (6)$$

$$\frac{\partial L}{\partial \theta_w} = 0 \quad (7)$$

Por último, se incluyen los rozamientos y el par motor:

$$\tau_w = T_m \quad (8)$$

$$\frac{\partial R}{\partial \dot{\theta}_b} = C_b \dot{\theta}_b \quad (9)$$

$$\frac{\partial R}{\partial \dot{\theta}_w} = -C_w \dot{\theta}_w \quad (10)$$

Donde:

- C_b = Coeficiente de rozamiento del péndulo con el eje de rotación
- C_w = Coeficiente de rozamiento del motor
- T_m = Par del motor ($T_m = K_m \cdot i$) ($K_m = 5.24 \cdot 10^{-3} \text{ NmA}^{-1}$)

Combinando todas las ecuaciones anteriores y resolviendo para hallar las ecuaciones diferenciales que definan el sistema de manera no lineal, se despeja las aceleraciones angulares del ángulo del péndulo y del volante de inercia obteniendo las ecuaciones principales que modelan el sistema del péndulo invertido controlado por volante de inercia:

$$\ddot{\theta}_b = \frac{(m_b l_b + m_w l) g \sin \theta_b - T_m - C_b \dot{\theta}_b + C_w \dot{\theta}_w}{I_b + m_w l^2} \quad (11)$$

$$\ddot{\theta}_w = \frac{(I_b + I_w + m_w l^2)(T_m - C_w \dot{\theta}_w)}{I_w(I_b + m_w l^2)} - \frac{(m_b l_b + m_w l) g \sin \theta_b - C_b \dot{\theta}_b}{I_b + m_w l^2} \quad (12)$$

En este sistema las variables de estado son:

$$X = (\theta_b \dot{\theta}_b \dot{\theta}_w)$$

JULIO 2016

Siendo:

- $\theta_b =$ Ángulo de la diagonal del péndulo.
- $\dot{\theta}_b =$ Velocidad angular del péndulo, a partir de ahora se denominará ω_b .
- $\dot{\theta}_w =$ Velocidad angular del volante de inercia, a partir de ahora se denominará ω_w .

4.2. Estimación de las variables de estado

Las variables de estado son el conjunto de variables de un sistema cuyo conocimiento en cualquier instante nos permite determinar la evolución futura del sistema sin más que conocer la entrada aplicada a partir de ese instante [26]. Es por eso, que la correcta estimación de las variables de estado del péndulo y volante de inercia resulta esencial para que el sistema realice su función correctamente.

4.2.1. Estimación del ángulo (ϑ_b)

Para la estimación del ángulo se sigue el método utilizado por el ETH de Zurich en su artículo “The Cubli: A Cube that can Jump up and Balance” [3]. El método se basa en el uso de dos acelerómetros posicionados en la diagonal de la cara del péndulo a unas distancias r_i del eje de rotación, $i=1,2$.

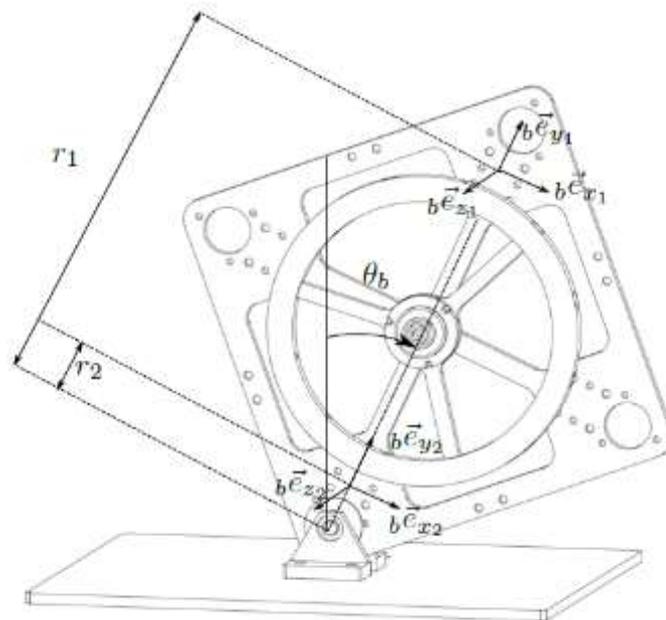


Figura 21: Estimación del ángulo [3]

$${}_{a_i}m := (r_i\ddot{\theta}_b + g \sin \theta_b, -r_i\dot{\theta}_b - g \cos \theta_b, 0), \quad i = 1,2 \quad (13)$$

Los términos dinámicos se eliminan resolviendo la siguiente ecuación donde $\mu = \frac{r_1}{r_2}$:

$${}_{a_1}m - \mu {}_{a_2}m = ((1 - \mu)g \sin \theta_b, -(1 - \mu)g \cos \theta_b, 0) =: (m_x, m_y, 0) \quad (14)$$

De las cuales se puede despejar la siguiente estimación del ángulo:

$$\hat{\theta}_b = \tan^{-1}\left(-\frac{m_x}{m_y}\right) \quad (15)$$

4.2.2. *Estimación de la velocidad de rotación de la cara del péndulo (ω_b)*

La estimación de la velocidad de rotación de la cara del péndulo se realiza utilizando las lecturas de los giróscopos de la IMU posicionados en la diagonal de la cara del péndulo. Las medidas se pasan a radianes por segundo y se calcula la media para tratar de minimizar el error.

$$\hat{\omega}_b = \frac{\omega_{z1} + \omega_{z2}}{2} \quad (16)$$

4.2.3. *Estimación de la velocidad de rotación del volante de inercia (ω_w)*

Para la estimación de la velocidad de rotación del volante de inercia, se utilizan los sensores hall del motor. Los sensores hall del motor envían seis combinaciones diferentes de impulsos por vuelta que son contados para determinar la velocidad. Además mandan señales desfasadas para determinar el sentido de rotación.

La señal que se mide con el microprocesador, es la velocidad actual promediada mediante un filtro digital paso-bajo de frecuencia 5 Hz. Este promedio lo calcula la controladora, y la señal se envía a través de una de las salidas analógicas de la controladora del motor. La señal analógica es proporcional a la velocidad del motor siguiendo la siguiente ecuación:

$$V = \frac{2 * \omega_{wmax}}{VCC} \omega_w + \omega_{wmax} \quad (17)$$

JULIO 2016

Donde:

- $\omega_{w_{max}} = 10000 \text{ rpm}$
- $V_{cc} = 3.3 \text{ V}$

Esta señal, una vez en la placa, vuelve a ser transformada a velocidad y con esto se obtiene la velocidad angular del volante de inercia.

Capítulo 5: Protocolos de comunicación

5.1. Puerto serie (UART)

UART (Universal Asynchronous Receiver/Transformer) es un controlador cuya función es controlar la interfaz para la comunicación serie. La UART coge bytes de los datos y los transmite bit a bit de manera secuencial. El puerto de recepción, convierte de nuevo estos bits en bytes de información. Este método de comunicación se denomina comunicación serie, y puede haber síncrona o asíncrona.

- Transmisión síncrona: El emisor y el receptor comparten un reloj. El emisor de la señal envía una señal que indica al receptor cuando debe comenzar a leer la información, si no hay información, se envía un carácter vacío.

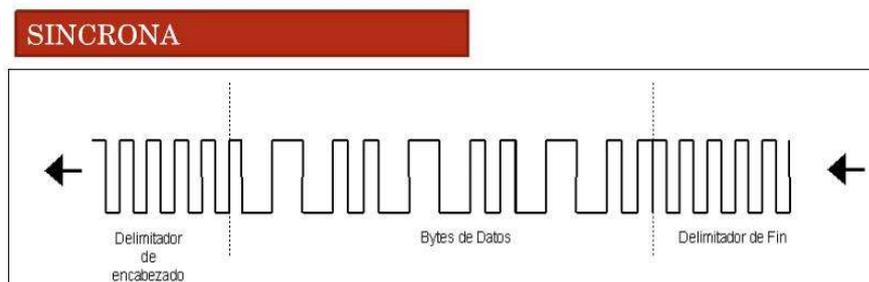


Figura 22: Comunicación serie síncrona

- Transmisión asíncrona: No es necesario una señal de reloj, los emisores y receptores deben estar previamente configurados con la misma velocidad para poder entenderse. En este caso, para saber cuándo se envían datos, se manda un bit de start, y al final de la comunicación un bit de stop. Para comprobar si la información se ha enviado correctamente, también se puede incluir un bit de paridad.



Figura 23: Comunicación serie asíncrona

El bit de start hace que los relojes se sincronicen temporalmente, es necesario que como máximo, estén desfasados en un 10% del periodo de la señal. Después del bit de start, se envían bytes de información, enviando primero el bit menos significativo. Si se ha configurado paridad, este bit iría después de la información. Por último iría el bit o bits de stop, que pueden ser hasta dos. Si se quiere seguir enviando información, se vuelve a recomenzar el proceso.

Las ventajas de UART son que solamente es necesario un cable y además tiene una interfaz muy sencilla de usar. Por otro lado, las desventajas son que necesita una precisión razonable de reloj, y no es excesivamente rápido.

5.2. Protocolo I2C

El protocolo I2C es un protocolo de comunicación serie del tipo maestro-esclavo para una interfaz de 2 cables con dispositivos de baja velocidad [25]. El I2C tiene una ventaja, y es que es muy fácil de usar. Cada esclavo tiene una única dirección, y la información se transmite en serie en paquetes de 8 bits.

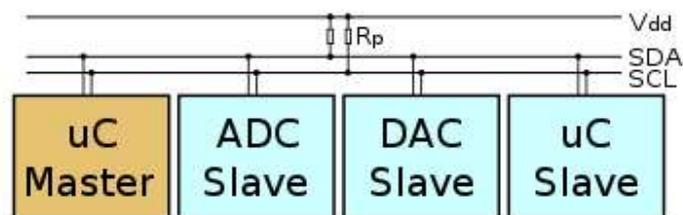


Figura 24: I2C maestro-esclavo

La interfaz de la comunicación I2C es simple, utiliza solo dos cables:

- SCL: Reloj
- SDA: Datos

Para comunicarse con los esclavos, cada uno de ellos tienen una dirección de 7 bits, el bit 8 indica si se lee (1) o se escribe (0) en el registro. El maestro no necesita dirección ya que es él el que genera la señal reloj y direcciona a los esclavos.

El protocolo, en estado normal, tiene las dos líneas de comunicación puestas a 1. Para comenzar a comunicarse, el maestro manda el bit de start (S) seguido de la dirección del esclavo con el que va a hablar (B1). Los siguientes bytes serán la información que va a leer o escribir, y para terminar envía un bit de stop (P), dejando la línea abierta para comunicarse con otros esclavos.

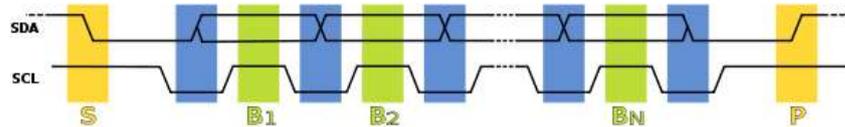


Figura 25: Ejemplo de comunicación I2C

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 6: Identificación de parámetros

Los parámetros del sistema se obtienen de diferentes métodos, algunos de hojas de datos de componentes y otros a través de ensayos. Para realizar los ensayos, se siguen los pasos marcados por “The Cubi: A Cube that can Jump Up and Balance”.

6.1. Parámetros del motor

De la hoja de características del motor se puede sacar la constante del par motor y la constante que define la corriente máxima del motor, cuyos valores son:

- $K_m = 13.5 \cdot 10^{-3} \text{ NmA}^{-1}$
- $K_i = 1.37 \text{ A}$
-

6.2. Masas

Las masas se miden en dos conjuntos por separado, estos conjuntos son el volante de inercia y el cuerpo del péndulo:

- $m_b = 207.3 \text{ g}$
- $m_w = 207.75 \text{ g}$
-

6.3. Distancias

Las distancias necesarias son las que van desde el eje de rotación inferior del péndulo al centro de gravedad del péndulo y el volante de inercia, y del eje de rotación inferior del péndulo al eje de rotación del volante de inercia.

La primera se obtiene directamente de las medidas del diseño en Solid Edge:

- $l = 91.925 \text{ mm}$

Para la segunda, lo primero que hay que hacer es calcular el centro de gravedad [\[20\]](#). Para realizar esto, se sujeta el péndulo con todas las piezas montadas desde varios extremos utilizando

JULIO 2016

una cuerda. La cuerda tiene que ser lo suficientemente larga como para, enganchando un peso al extremo que queda suelto, sea más larga que la cara del péndulo. Se cuelga la cuerda y una vez se queda estático, se pinta sobre un papel pegado al cuerpo del péndulo la trayectoria de la cuerda. Una vez se siguen los mismos pasos desde diferentes puntos, el lugar donde se crucen las líneas marca el centro de gravedad. Midiendo la distancia desde ese punto al eje de rotación, se obtiene:

- $l_b = 85 \text{ mm}$

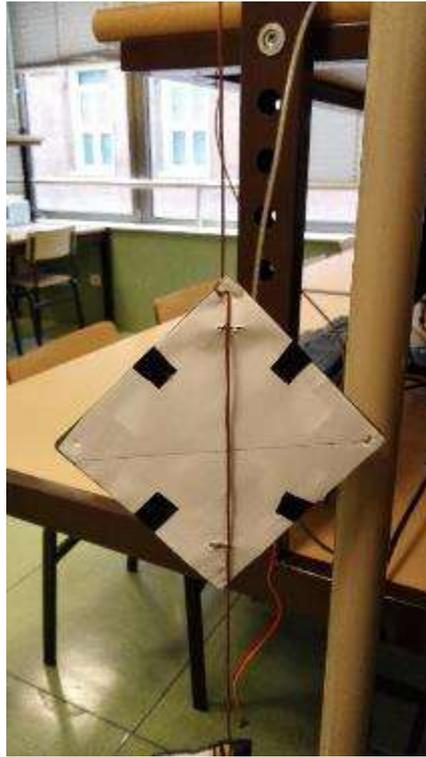


Figura 26: Identificación del centro de gravedad

6.4. Momento de inercia y coeficiente de rozamiento del volante

Para encontrar el momento de inercia y el coeficiente de rozamiento del volante de inercia se realiza un ensayo en el cual se mandan escalón de corriente al motor y se mide la velocidad angular a la que gira y varía manteniendo el cuerpo del péndulo fijo. Esto da la siguiente ecuación:

$$I_w * \ddot{\theta}_w(t) = K_m * u(t) - C_w * \dot{\theta}_w(t) \quad (18)$$

La función de transferencia que relaciona la velocidad angular y la corriente, por lo tanto, es:

$$\frac{\dot{\theta}_w(t)}{u(t)} = \frac{\frac{K_m}{I_w}}{s + \frac{C_w}{I_w}} \quad (19)$$

Para ajustar el modelo, se utiliza el comando “ident” en Matlab. La función de este comando consiste en, metiendo unos datos de entrada, que contengan las series temporales de la corriente y la velocidad, realiza una estimación del tipo de modelo que se le especifique, en este caso una función de transferencia con un polo y ningún cero.

Debido a que las comunicaciones a través de puerto serie contienen ruido, es necesario filtrarlas previamente. Esto se ha realizado mediante un filtro de medias móviles, con el cual se reduce en gran manera el ruido.

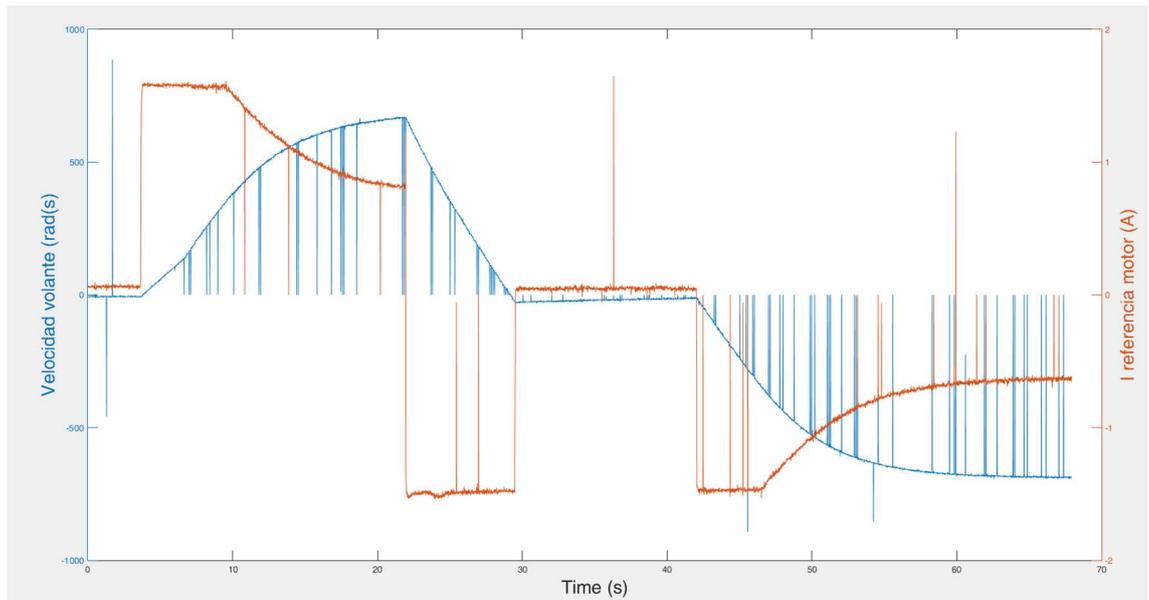


Figura 27: Serie temporal de corriente (naranja) y velocidad en rpm (azul)

JULIO 2016

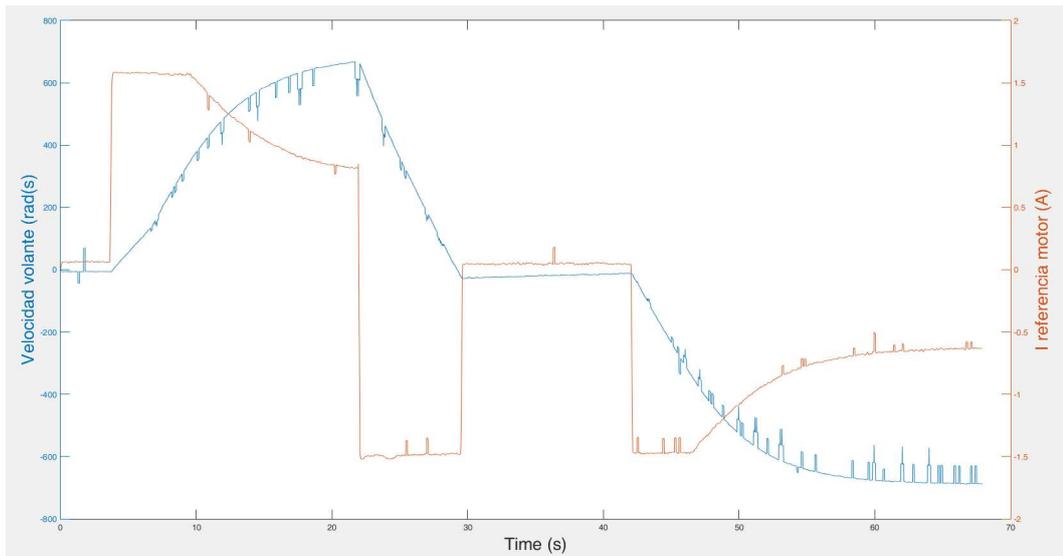


Figura 28: Serie temporal de corriente (naranja) y velocidad en rad/s (azul) filtrados

El resultado de la estimación del modelo, con un ajuste del 90,56% es el siguiente:

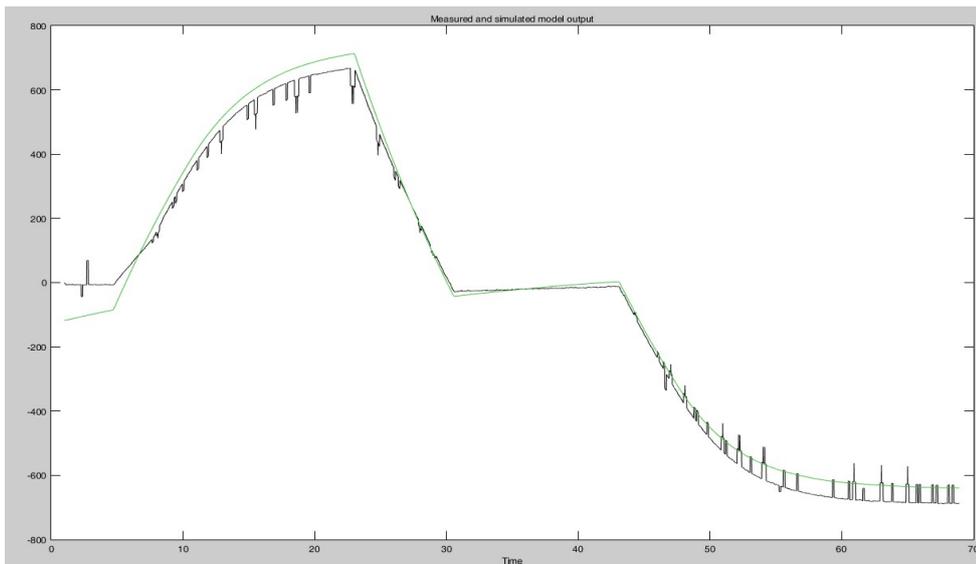


Figura 29: Velocidad real (negro) vs velocidad estimada (verde)

Cuya función de transferencia es:

$$\frac{\theta_w(t)}{u(t)} = \frac{56.17}{s + 0.05396}$$

Donde:

- $I_w = 0.24 \cdot 10^{-3} \text{ kgm}^2$
- $C_w = 0.013 \cdot 10^{-3}$

6.5. Momento de inercia y coeficiente de rozamiento del péndulo

Para hallar el momento de inercia y coeficiente de rozamiento del péndulo, se sigue el mismo método que para los del volante de inercia, lo que cambia es que ahora lo que se mide es el ángulo del péndulo y su velocidad de giro para ajustar la serie temporal a la siguiente ecuación:

$$(I_b + I_w + m_w l^2) \ddot{\theta}_b(t) = -C_b \dot{\theta}_b(t) + (m_b l_b + m_w l) g \sin \theta_b(t) \quad (20)$$

Para medir las variables, lo primero que hay que hacer es fijar el volante de inercia al cuerpo del péndulo, de esta manera no se puede mover y no afecta al movimiento. Una vez está fijado, hay que sujetarlo al revés, dejando que cuelgue. Después se sujeta en una posición de aproximadamente 20 grados y se deja en caída libre hasta que se estabilice. Esto se realiza varias veces para poder obtener varias medidas. En el siguiente gráfico, se ve que la primera vez que se suelta es en aproximadamente el segundo 20 (gráfico en décimas de segundo), y se repite otras 6 veces más.

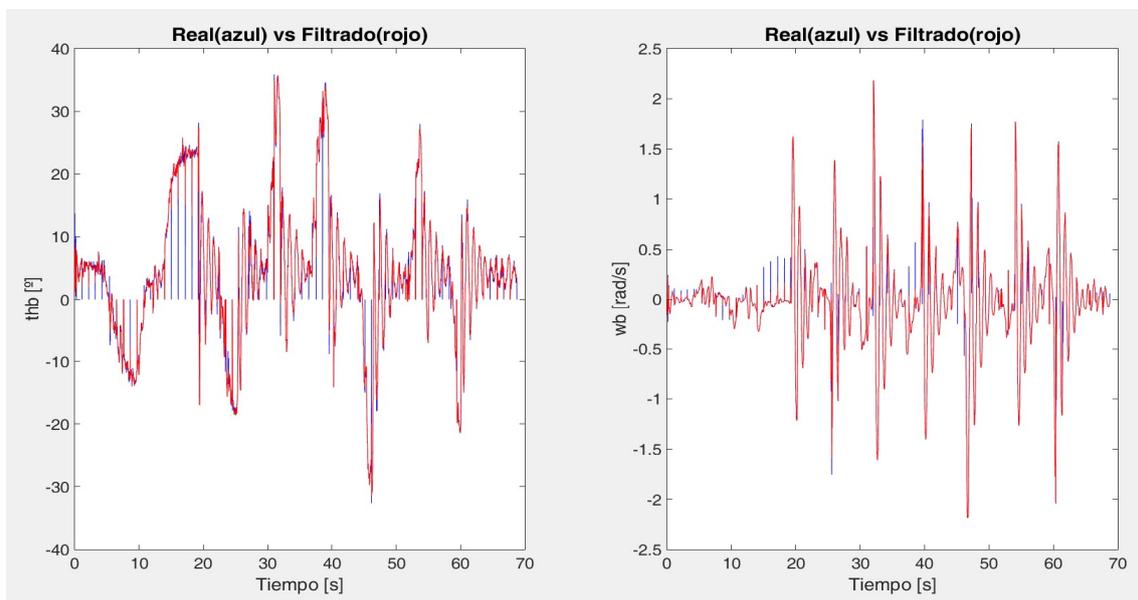


Figura 30: Medidas del ángulo y velocidad angular sin filtrar y con un filtro de medias móviles

Viendo que las medidas tienen bastante ruido, se aplica un filtro de medias móviles haciendo que cada valor sea igual a la media de los últimos 12 valores, de esta manera se consigue eliminar parte del ruido y suavizar las curvas.

JULIO 2016

Analizando las curvas, además se puede comprobar que el péndulo no se estabiliza exactamente en 0 grados, se estabiliza a 2 grados. Esto es importante y se tiene que tener en cuenta a la hora de realizar el control, ya que si queremos que se balancee sobre un punto, será más fácil en el cual el centro de gravedad coincida con el eje de rotación que queriendo tener la diagonal completamente vertical.

Para asegurarse de que el ajuste de la ecuación sea lo más cercano posible a la realidad, se separan las 7 zonas en las cuales el péndulo está en balanceo libre.

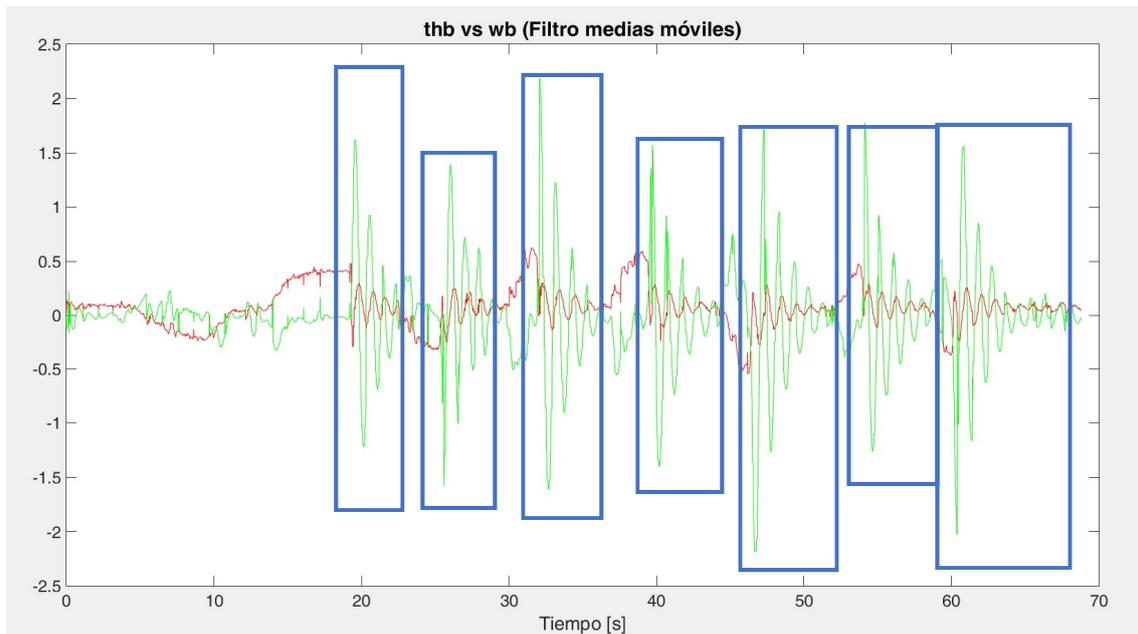


Figura 31: Zonas de interés de ángulo y velocidad angular filtradas por medias móviles

Una vez se ha separado en esas siete zonas de interés, se aplica un filtro de tercer orden a cada una de ellas, y a la velocidad angular, también un filtro derivativo. De esta manera se obtiene también la medida que faltaba, la aceleración angular. Para este filtro, se coge un valor de ω_n igual a $\frac{2\pi}{T}$ siendo $T=0.1$ s el periodo de balanceo del péndulo, y un factor de amortiguamiento $\xi=0.7$. De esta manera se obtiene, tomando de ejemplo la última sección, algo así:

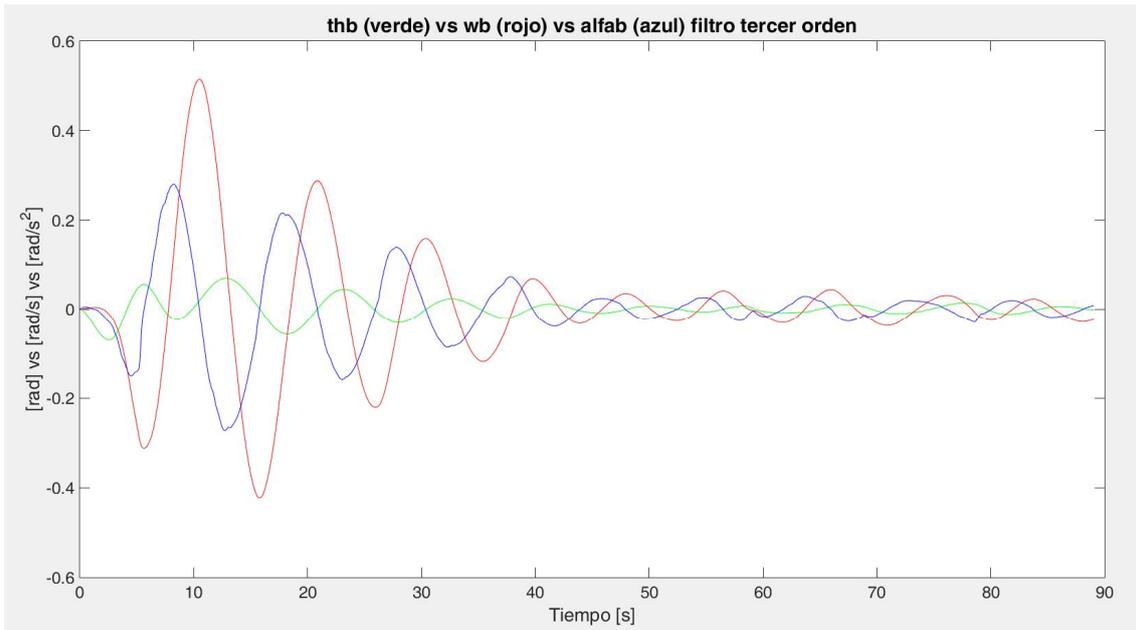


Figura 32: θ_b (verde) vs ω_b (rojo) vs a_b (azul) después de filtro de tercer orden

Una vez se han adquirido todas las medidas correctamente filtradas, se procede a despejar las constantes de las variables. Para ello, se utiliza una aproximación de mínimos cuadrados, y resolviendo, se obtienen los dos parámetros que faltaban:

- $I_b = 8.0 \cdot 10^{-3} \text{ kgm}^2$
- $C_b = 1.0 \cdot 10^{-3}$

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 7: Diseño del control

7.1. Control de balanceo

Mantener el equilibrio en la posición de ángulo $\vartheta_b = 0$ es el objetivo principal del proyecto. Para lograrlo hay que diseñar y aplicar un control de balanceo robusto. Puesto que se está tratando de imitar unos resultados parecidos a los del ETH de Zurich, se diseña el mismo tipo de control, el control LQR [21].

LQR significa Linear-Quadratic Regulator (regulador lineal cuadrático). Esto viene de que la aplicación de este regulador es principalmente para sistemas dinámicos que están definidos mediante ecuaciones diferenciales lineales, y su función de coste es cuadrática. Lo que hace el algoritmo LQR es tratar de minimizar esta función de coste para encontrar las ganancias de un control de realimentación de estado. La dificultad de este regulador es que es un proceso iterativo de prueba y error para encontrar los parámetros óptimos. En este caso se va a hacer uso del controlador en tiempo discreto.

$$J = \sum_{k=1}^{\infty} (x[k]^T Q_x [k] + u[k]^T Q_r u[k] + 2x[k]^T N u[k]) \quad (21)$$

Para poder utilizar este control, antes hay que linealizar el sistema dinámico de manera que se obtenga un espacio de estado de la siguiente forma:

$$\begin{aligned} \frac{dX}{dt} &= AX + BU \\ Y &= CX + DU \end{aligned} \quad (22)$$

Las variables de estado son, como ya se ha mencionado anteriormente $X = (\theta_b \ \dot{\theta}_b \ \dot{\theta}_w)$, y el sistema se linealiza para los valores de $x = (0, 0, 0)$, resultando en las siguientes matrices de estado:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{(m_b l_b + m_w l)g}{I_b + m_w l^2} & \frac{C_b}{I_b + m_w l^2} & \frac{C_w}{I_b + m_w l^2} \\ -\frac{(m_b l_b + m_w l)g}{I_b + m_w l^2} & \frac{C_b}{I_b + m_w l^2} & -\frac{C_w(I_b + I_w + m_w l^2)}{I_w(I_b + m_w l^2)} \end{bmatrix}$$

JULIO 2016

$$B = \begin{bmatrix} 0 \\ \frac{K_m}{I_b + m_w l^2} \\ \frac{K_m(I_b + I_w + m_w l^2)}{I_w(I_b + m_w l^2)} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Puesto que se va a usar un control en tiempo discreto, hay que discretizar las ecuaciones de estado. El tiempo de muestreo T_s que se utiliza para ello es 0.01s. Aplicando el controlador en tiempo discreto al sistema discretizado se obtienen las ganancias de realimentación de estado, una por cada variable de estado del sistema.

$$Klqr = (K_1, K_2, K_3)$$

Estas ganancias vienen de la solución de la siguiente ecuación:

$$Klqr = (B^T S B + Q_r)^{-1} (B^T S A + N^T) \quad (23)$$

Que a su vez viene simplificar por mínimos cuadrados la ecuación de coste cuadrático utilizando la ecuación de Riccati para encontrar S, donde N se suele despreciar para simplificar los cálculos:

$$A^T S A - S - (A^T S B + N)(B^T S B + Q_r)^{-1} (B^T S A + N^T) + Q_u = 0 \quad (24)$$

7.2. Control de salto

El control de salto, técnicamente no se le podría denominar control en este caso. Se calcula de manera experimental. Utilizando el ESCON Studio para medir la corriente y velocidad del motor con mayor exactitud, y la conexión en serie con Simulink para medir el ángulo, se va probando aumentar la corriente de referencia hasta que llega a unas rpm determinadas. Una vez se alcanza esa velocidad, se activa el sistema de frenado, que al parar de golpe el giro del volante de inercia, conseguirá que este salte hasta alcanzar un ángulo. Este proceso se repite hasta que el salto alcanza un ángulo de entre 0 y 10 grados.

7.3. Puesta en conjunto

Para poner en conjunto el salto y el balanceo se utiliza una máquina de estados del tipo Moore [22] de cuatro estados muy simple. Los estados que incluye son los siguientes:

- Inicio: El péndulo está en la posición de reposo a la espera de la pulsación del botón de encendido, cuando este se pulsa, se inicia un temporizador para calibrar el motor y los giróscopos. Cuando este termina, se pasa al siguiente estado.
- Aceleración: Una vez pulsado el botón de encendido, el volante de inercia es acelerado. Cuando alcanza la velocidad adecuada para saltar pasa al siguiente estado.
- Salto: Entra en acción el servomotor actuando de freno. El brazo del servo se mantiene en la posición de frenado el tiempo suficiente para que el péndulo alcance el ángulo apropiado para comenzar la actuación del control de balanceo.
- Balanceo: El control de balanceo trata de estabilizar el péndulo sobre su esquina ($\theta_b = 0$).

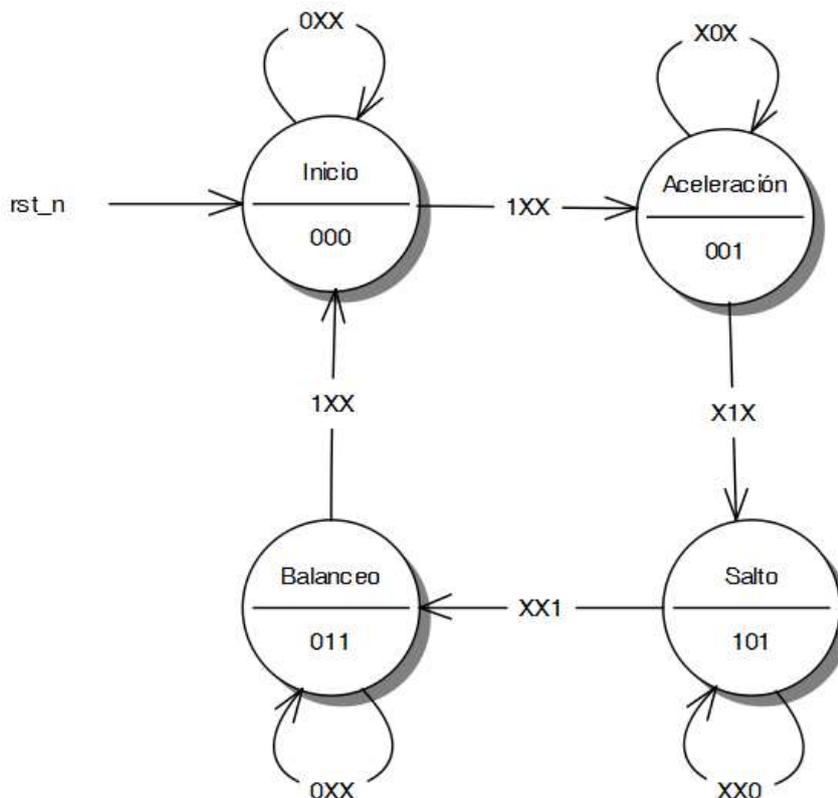


Figura 33: Máquina de estados de funcionamiento del Cubli

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

Las entradas de la máquina de estados son (Temporizador de calibración, ω_w [rpm], Temporizador):

- Temporizador de calibración: 1 si termina de contar, 0 no ha contado tiempo.
- ω_w : 1 si el módulo de la velocidad es igual que la velocidad necesaria para saltar, 0 si es menor.
- Temporizador: 1 Cuando termina de contar el tiempo del salto, 0 mientras cuenta.

Las salidas de la máquina de estados son (Servo, Balanceo, Motor):

- Servo: 1 Posición de frenado del servomotor, 0 posición de reposo.
- Balanceo: 1 Activar el control LQR de balanceo, 0 control no activado.
- Motor: 1 Motor activado, 0 motor desactivado.

Capítulo 8: Implementación en Simulink

8.1. Simulaciones

Para las simulaciones se plantea un único script y un único simulador en los cuales se linealiza la planta y se realiza el control. En el script se pueden cambiar los diferentes parámetros, como las ganancias del control o los valores iniciales de las diferentes variables, y de esta manera podemos imitar las condiciones de balanceo y las de salto.

El diagrama de la figura 34 representa al sistema real. El sistema de péndulo con volante de inercia se modela mediante las ecuaciones no lineales explicadas anteriormente y estas se representan en el subsistema de “planta no lineal”. A este subsistema, de entrada llega la corriente y de salida las variables de estado.

Para ser capaz de comprender e interpretar mejor estas variables y los resultados obtenidos con los diferentes controles, el ángulo se convierte a grados, y la velocidad angular del disco a revoluciones por minuto. Además, como la intensidad se va a enviar al dispositivo mediante una señal analógica cuyo valor será de tensión, se convierte la corriente en intensidad para visualizar el valor del mando, y se vuelve a reconvertir a intensidad antes de entrar de nuevo en la planta.

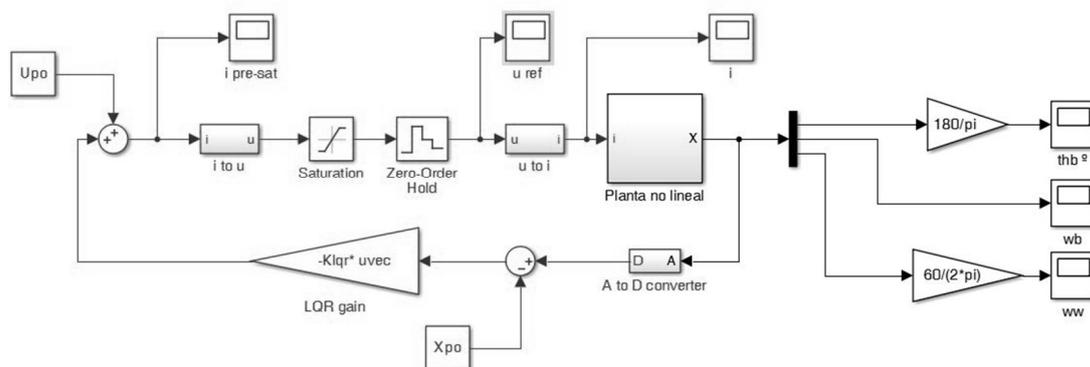


Figura 34: Simulador sistema no lineal

JULIO 2016

Para ajustar el control de balanceo, primero se hace una aproximación desde el simulador. Se prueban diferentes pesos en las matrices R y Q hasta se consigue una respuesta decente. A partir de ese valor, se realizan cambios más pequeños directamente en la matriz de ganancias hasta conseguir un resultado óptimo.

También se plantean diferentes condiciones iniciales intentando imitar posibles casos reales. La función de probar el control en diferentes situaciones es porque podría ocurrir que un control determinado fuera el mejor para equilibrarse, pero también podría haber otro que no fuera tan bueno en esa situación, pero fuera mejor en controlar la recepción de un salto. Entre estas condiciones se pueden encontrar:

- Vuelta a ángulo cero desde un ángulo determinado: Esto serían las condiciones típicas de funcionamiento, simplemente equilibrarse.
- Balanceo desde salto previo: Esta condición intentaría imitar como actúa el control cuando recibe el salto.

8.2. Bloques Waijung

Para la implementación al sistema real, se utilizan los bloques Waijung [23]. Waijung consiste en un set de bloques de Simulink cuya característica principal es que es capaz de generar automáticamente y de manera sencilla código en C a partir del código de Matlab/Simulink para diferentes microprocesadores, específicamente aquellos de la familia STM32F4.

El mayor beneficio que se obtiene de utilizar los bloques Waijung se debe a que, además de generar automáticamente el código en C para los microprocesadores, la interfaz es muy sencilla de utilizar, y viene con demos de la implementación de los bloques.

Otra ventaja de este paquete de Simulink deriva de la posible combinación de los mismos con los bloques de serie de Simulink. Así combinándolos con otros sets se logra realizar programas intuitivos para ser capaces de comprender mejor el funcionamiento del sistema.

Lo primero que hay que hacer, es colocar en el diagrama de Simulink el bloque de “Target Setup”, en el cual se selecciona el microprocesador del que se va a hacer uso, el compilador y el tiempo de muestreo. También se pueden configurar otros parámetros, como por ejemplo borrar la memoria antes de cargar un programa nuevo. Con esto ya se puede comenzar a construir el resto del diagrama de bloques.

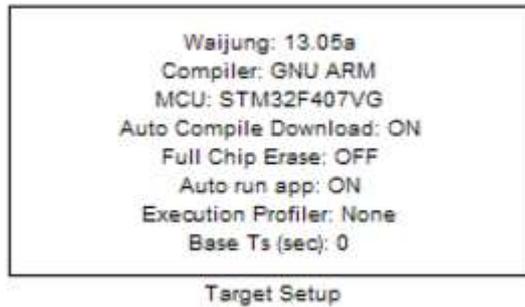


Figura 35: Target Setup (bloque Waijung)

Los siguientes bloques importantes a tener en cuenta, son los de configuración de comunicaciones. Siempre que se va a establecer algún tipo de comunicación entre microprocesador y periféricos u ordenador es necesario colocar estos bloques. En este caso los parámetros que se inicializan son el tiempo de muestreo, las velocidades de la comunicación, el tipo de codificación y los pines que se vayan a utilizar, ya sea para transmisión, recepción o ambas.

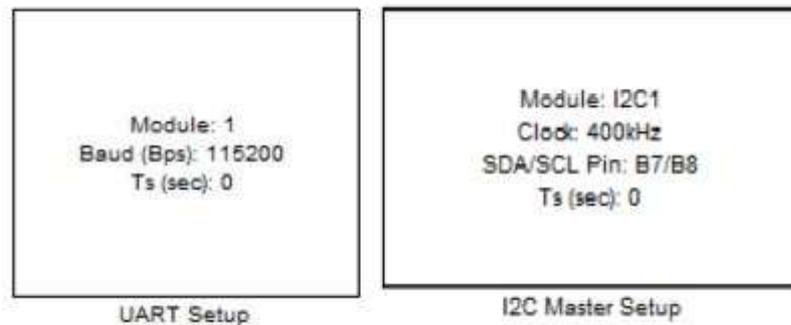


Figura 36: Configuración de comunicaciones (bloque Waijung)

Aparte de estos bloques de configuración se utilizan los siguientes bloques:

- ADC: Convertidor analógico digital. Recibe una señal analógica entre 0 y 3.3 V y transforma esa señal a binario entre 0 y 4095 bits respectivamente. Se usa para leer la velocidad angular del volante de inercia.

JULIO 2016

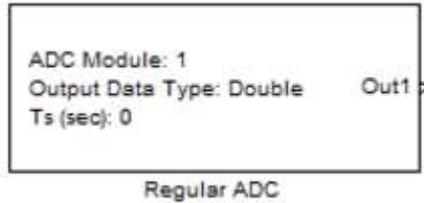


Figura 37: Convertidor AD (bloque Waijung)

- DAC: Convertidor digital analógico. Convierte una señal digital a una señal analógica de entre 0 y 3.3V. Se utiliza para enviar la referencia del mando, la corriente, al motor del volante de inercia.

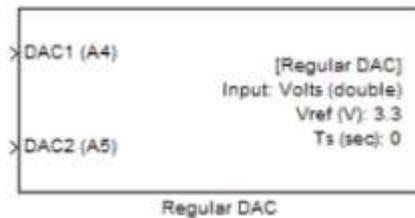


Figura 38: Convertidor DA (bloque Waijung)

- I2C “Read/Write”: Este bloque sirve para escribir o leer registros mediante la comunicación I2C. Siempre hay que indicar la dirección del esclavo en la entrada “SlvAddr”.

Si se va a leer un registro simplemente se escribe la dirección de memoria de ese registro en Wr0 y la lectura de ese registro saldrá por Rd0. Es posible añadir más entradas y salidas de escritura y lectura, ya que cada una de ellas recibe un byte de información.

Para escribir en un registro, se indica en Wr0 la dirección, y lo que se vaya a escribir se indica en Wr1 hasta Wrn, n siendo el número de bytes a escribir.

La salida status indica si la comunicación se establece correctamente, 0 indica éxito, 1 error de comunicación y 2 “timeout”.



Figura 39: I2C leer/escribir (bloque Waijung)

- Digital Input: Recibe entradas digitales binarias. Se usa como interruptor o pulsador.

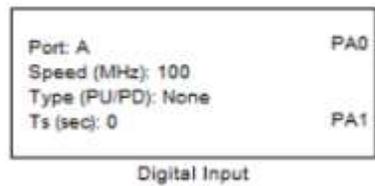


Figura 40: Entrada digital (bloque Waijung)

- Digital Output: Recibe salidas digitales binarias. Se usa para encender y apagar componentes, como por ejemplo un LED.

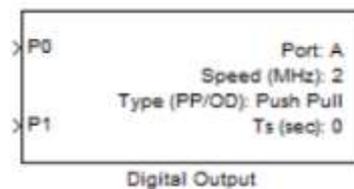


Figura 41: Salida digital (bloque Waijung)

- Basic PWM: Implementa una señal de modulación de ancho de pulso utilizando un temporizador de 16 bits. Dentro del bloque se fija el tiempo de muestreo, el periodo de pulso y la polaridad del mismo. En las entradas se envía una constante con valor de 0 a 100 que indica el porcentaje del periodo del pulso en el que está activo. Se utiliza para controlar el servomotor.

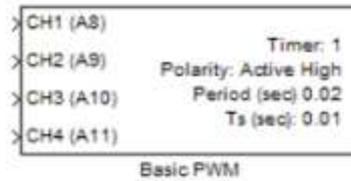


Figura 42: Salida PWM (bloque Waijung)

- UART Tx: Se configura el bloque de transmisión de comunicación UART. Se indican el número de entradas con cada tipo de datos (uint8, int32, doublé...), el formato de los datos (binario o ASCII), la cabecera y la terminación de la información a enviar. (Nota, para la recepción de los datos, ya que solamente se hace para las simulaciones, se utiliza el bloque de comunicación serie standard de Simulink para el ordenador).

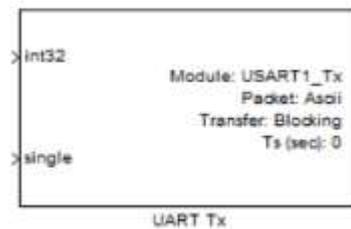


Figura 43: Transmisión UART (bloque Waijung)

8.3. Pruebas en sistema real

Antes de construir todo el sistema en conjunto, es necesario ir realizando pruebas en los diferentes módulos que conforman el sistema del péndulo y volante de inercia y ajustar los diferentes parámetros correctamente. Los módulos que se prueban por separado son, el servomotor, el motor y los sensores de posición y velocidad. Además de para ajustar los parámetros, estas pruebas sirven para ir preparando los módulos y configurarlos, para que una vez se pongan en común, se haya asegurado su correcto funcionamiento.

8.3.1. Servomotor

Los servomotores son un dispositivo electrónico que, mediante un motor eléctrico y un circuito electrónico giran el eje de salida en base a una señal PWM [24]. La señal de control, generalmente, consiste en un pulso cuyo periodo es de 20 ms y el ancho de pulso para controlar

el servo varía entre 0.5 y 2.5 ms. En este caso se utiliza un servomotor de engranajes metálicos (MG90S) para poder resistir la fuerza del golpe para el frenado del volante de inercia. Las equivalencias en ángulo son aproximadamente para -90° , 0.5 ms y para 90° 2.5 ms, como se muestra en la siguiente figura:

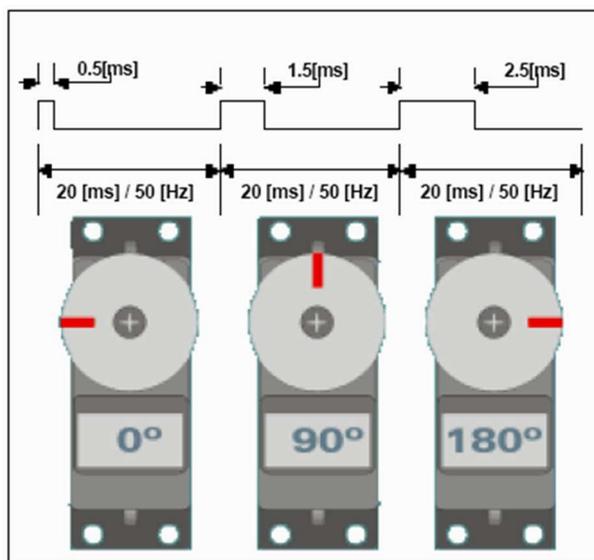


Figura 44: Funcionamiento servomotor

Para comprobar el correcto funcionamiento del servomotor, y ajustar sus parámetros de control para que esté en la posición adecuada, se realiza un diagrama de Simulink muy sencillo, que además sirve también para asegurarse de que el pulsador hace lo que debe.

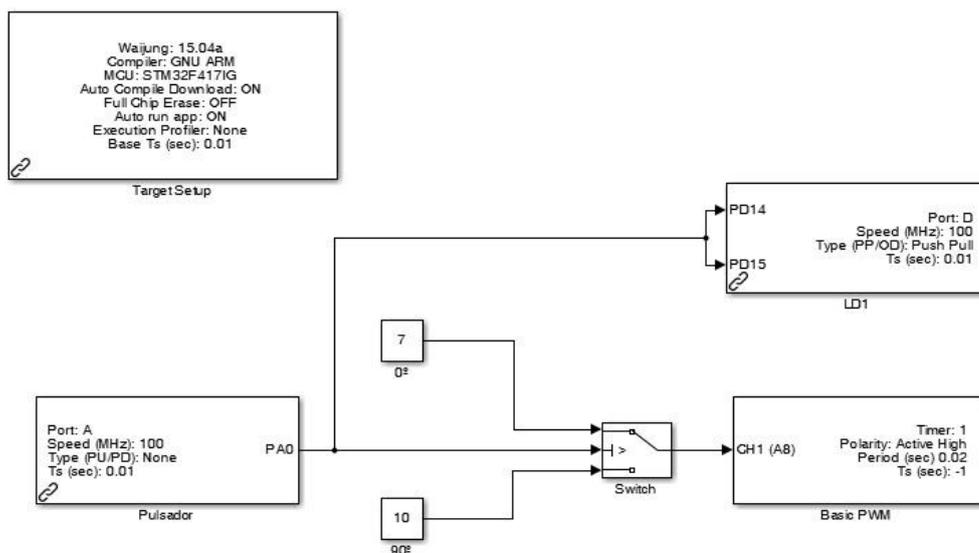


Figura 45: Prueba del servomotor y pulsador

JULIO 2016

En esta prueba, se conectan dos LED al pulsador para que si al pulsarse, se encienden los LEDs, se asegura que cumple su función correctamente. Para el servomotor, se conecta la salida del pulsador con un “switch”. Cuando el pulsador está en 1, llega al bloque de señal PWM un 7, y si está en 0, llega un 10. Las constantes que se están enviando al bloque de PWM son el porcentaje de ancho de pulso de la señal sobre el tiempo de muestreo. Esto significa, que cuando el pulsador está en modo ON, el servo se posiciona en $+90^\circ$ (2 ms), y en OFF, se coloca en 0° (1.4ms).

8.3.2. Motor DC

El motor que se utiliza para girar el volante de inercia es un Maxon EC flat de 12 W controlado por una controladora Maxon ESCON Module 24/2. El uso de esta controladora facilita la configuración del motor ya que viene con un pequeño software que permite ajustar todos los parámetros de control, realizar medidas y visualizar las mismas a tiempo real. Aparte de las conexiones de los devanados del motor, los sensores hall y la tensión de funcionamiento, se configura la controladora con las siguientes entradas y salidas:

- Entrada analógica 1: Esta entrada recibe una señal analógica cuyo valor varía entre 0 y 2.9 V (técnicamente serían 3.3, pero comprobando experimentalmente, el valor máximo que recibe es de 2.9 V). Esta señal es proporcional a la intensidad cuyo rango recorre desde -1.36 A hasta +1.36 A.

La señal analógica sale de uno de los convertidores digital-analógico de la placa. En un principio, la primera placa de prueba no tenía de estos convertidores, y se planteó diseñarlo y montarlos mediante un filtro paso bajo, pero el tiempo requerido para lograr ese objetivo iba a ser demasiado, además de que podría enviar una señal con más ruido que comprar una placa que viniera con ellos. Por lo tanto ese planteamiento se dejó de lado.

- Entrada digital 1: La entrada digital simplemente contiene un habilitador. Si no está activado, no permite el paso de corriente al motor.
- Salida digital 1: Esta salida envía una señal analógica a la placa cuya información es la velocidad promediada del motor. Se decide mandar la promediada puesto que así se elimina ruido posible ruido en la señal. La señal de velocidad del motor tiene un rango de entre 0 y 3.3 V, que llega al microprocesador, y este con un convertidor analógico-digital lo convierte a binario en una variable doble, cuyo significado en velocidad tiene un rango proporcional que cubre desde -10000 a +10000 rpm.

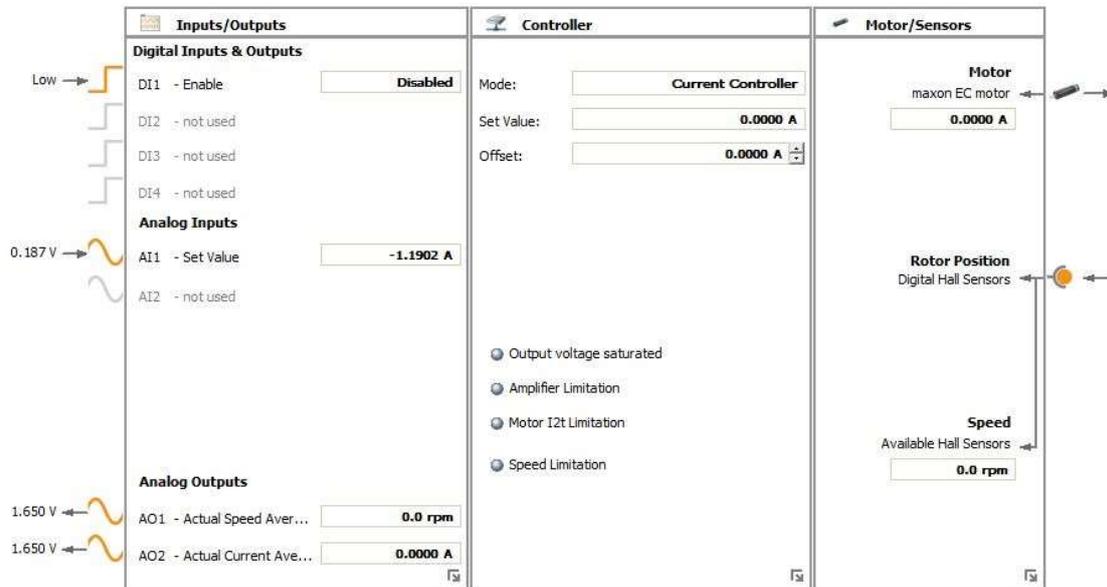


Figura 46: ESCON Studio (Monitor de control)

Para la prueba del motor, en vez de recibir una referencia de velocidad desde el STM32, lo configuramos para obtener esa referencia desde el ESCON Studio, y con la placa simplemente leer el valor de velocidad, que a través de comunicación serie con el ordenador, se muestra en otro diagrama de Simulink.

La comunicación serie se configura con los siguientes parámetros:

- Baudrate: 28800 bps
- Bits: 8
- Parity: None
- Stop bits: 1
- Ts: 0.01 s

JULIO 2016

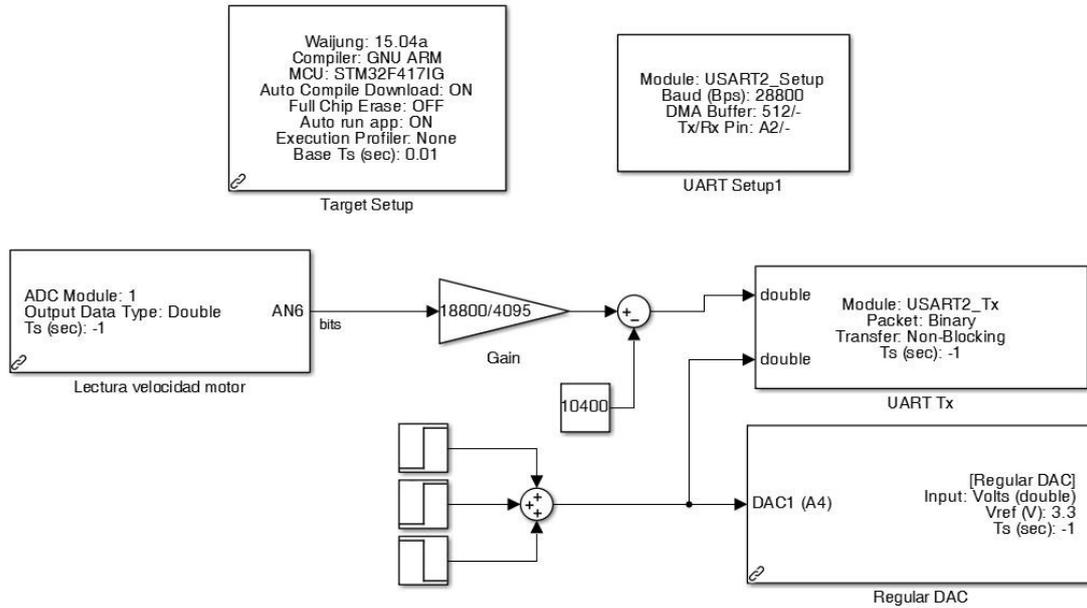


Figura 47: Configuración del offset de la velocidad del motor

Una vez está configurada y activa la comunicación serie, y el programa cargado, se va aumentando la velocidad de referencia al motor, y comparando las lecturas de velocidad en el ordenador provenientes de la controladora de Maxon, y del microprocesador STM32. El objetivo de esta comprobación, es ajustar el offset que provoca la señal analógica al transmitir la velocidad del motor en comparación con la real. La ecuación teórica de equivalencia sería, siguiendo la transformación de voltios a bits y bits a revoluciones por minuto:

$$bits = \frac{4096}{V_{cc}} * V \quad (25)$$

$$\omega_w = \frac{2 * \omega_{wmax}}{4096} * bits - \omega_{wmax} \quad (26)$$

A base de prueba y error, la configuración óptima en la cual existe el mínimo error sería la siguiente:

$$\omega_w = \frac{2 * (\omega_{wmax} * 0.94)}{4096} * bits - (\omega_{wmax} + 40) \quad (27)$$

A pesar de realizar ese ajuste, para valores de velocidad angular pequeños sigue existiendo ruido. Para eliminar este ruido, se implementa un pequeño subsistema de calibración. Este subsistema consta de dos partes:

- Temporizador de calibración:

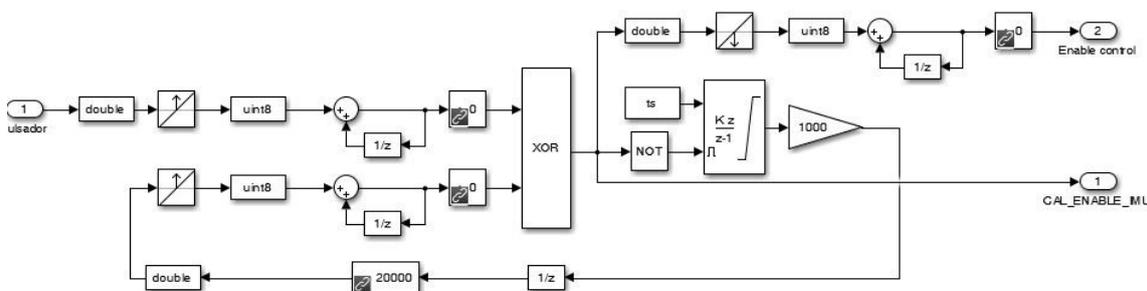


Figura 48: Temporizador de calibración de la lectura de la ω_w

La función del temporizador es, cuando se da una señal de inicio (en este caso el pulsador de la placa), detecta el flanco de esa señal y comienza a contar un tiempo en milisegundos. Mientras está contando, habilita el subsistema de calibración, una vez termina la cuenta, habilita el control. Se utiliza un tiempo de 20 segundos para durar el tiempo suficiente para eliminar cualquier offset que contenga la lectura.

(Nota: la habilitación del control es solo para cuando se implemente la lectura de la velocidad angular en la puesta en conjunto de todos los elementos del sistema).

- Calibración:

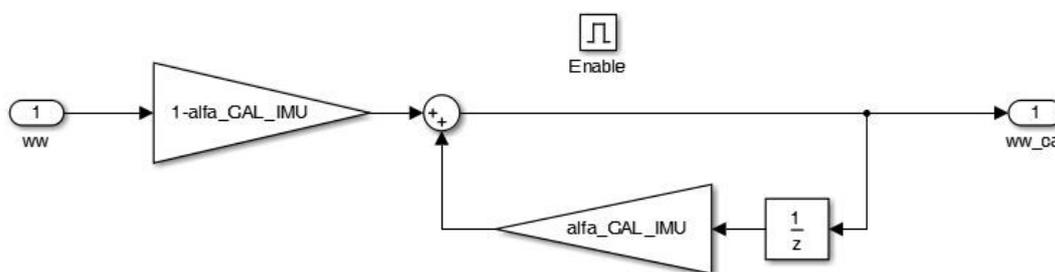


Figura 49: Calibración de la ω_w

La función del sistema de calibración es calcular la media de la lectura mientras el temporizador está activo. Una vez se acaba el tiempo, se resta la media calculada al valor real de la medida, y de esta manera se consigue eliminar el offset. Para hacer esta calibración es necesario asegurarse que la velocidad del motor sea 0 rpm, ya que resta la media del valor actual que se esté

JULIO 2016

leyendo para conseguir 0 rpm sin offset. Si estuviera moviéndose por ejemplo a 200 rpm, ya que la calibración es para 0, se habría calibrado con un offset de 200 rpm.

Además de incluir la calibración, esta vez, en vez de enviar la referencia de corriente para alcanzar una velocidad al motor, se indicará la velocidad utilizando el ESCON Studio. Utilizando la herramienta de Maxon se asegura que la referencia que se pide es más exacta, ya que se envía directamente a la controladora.

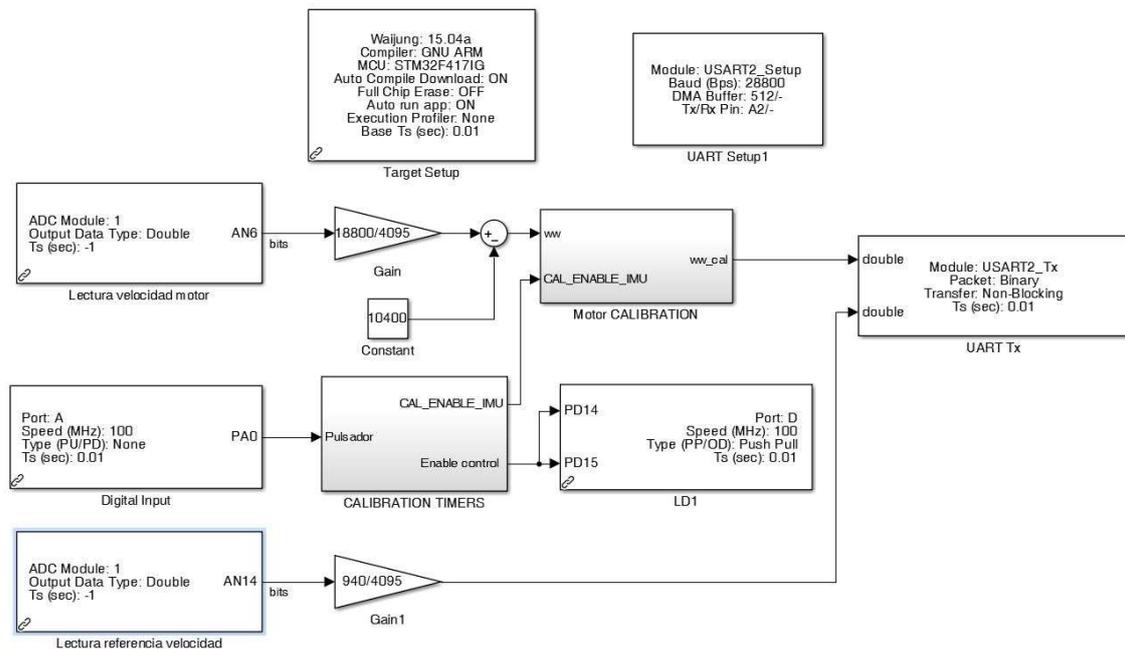


Figura 50: Diagrama de Simulink para la comparación referencia velocidad angular del disco con la velocidad real

Para comprobar que se han ajustado correctamente los parámetros, se realizan lecturas de las series temporales de la referencia de la velocidad y la velocidad real. Se utiliza para esto dos convertidores AD. De esta manera se prueban varias referencias de velocidad hasta aproximadamente 850 rpm, ya que en el control no serán necesarios giros más rápidos salvo en el salto.

8.3.3. Sensores (IMU)

Los sensores que se utilizan para medir la orientación y velocidad del péndulo se denominan IMU (Inertial Measurement Unit), que significa Unidad de Medida Inercial. Este aparato electrónico utiliza una combinación de acelerómetros y giróscopos para obtener esas

medidas. Para este sistema, se utilizan las medidas de aceleraciones en los ejes X e Y y el giróscopo en los ejes Z, como se explica en los apartados [3.2.1](#) y [3.2.2](#).

Los sensores se comunican con el microprocesador mediante el protocolo I2C para realizar las medidas. Las medidas de la IMU no son triviales, no es enchufar y medir. Antes hay que realizar una serie de configuraciones de todos los registros y transformaciones a las medidas para que el resultado final esté en las unidades adecuadas.

Lo primero que hay que hacer es asegurarse de que se tiene el ID correcto del sensor, por lo tanto se hace una prueba inicial muy sencilla en la cual se pide que envíe su nombre llamando al registro “Who Am I”. Esto es importante ya que al utilizar dos sensores, una de las patillas del MPU6050 tiene que estar conectada en un caso a Vcc y en el otro a tierra, y devuelven un ID diferente. Este ensayo se realiza en las dos IMU.

Para escribir o leer un registro en la IMU, hay que indicar en el bloque de conexión I2C la dirección de cada uno. El MPU6050 permite, conectando la patilla de AD0 a Vcc o a tierra, estar los dos conectados a la misma línea de I2C, la diferencia es que la dirección del que está conectado a tierra es 0x68, y el que está conectado a Vcc es 0x69. Además a esta dirección, hay que añadir en el bit 0 si se va a escribir o a leer.

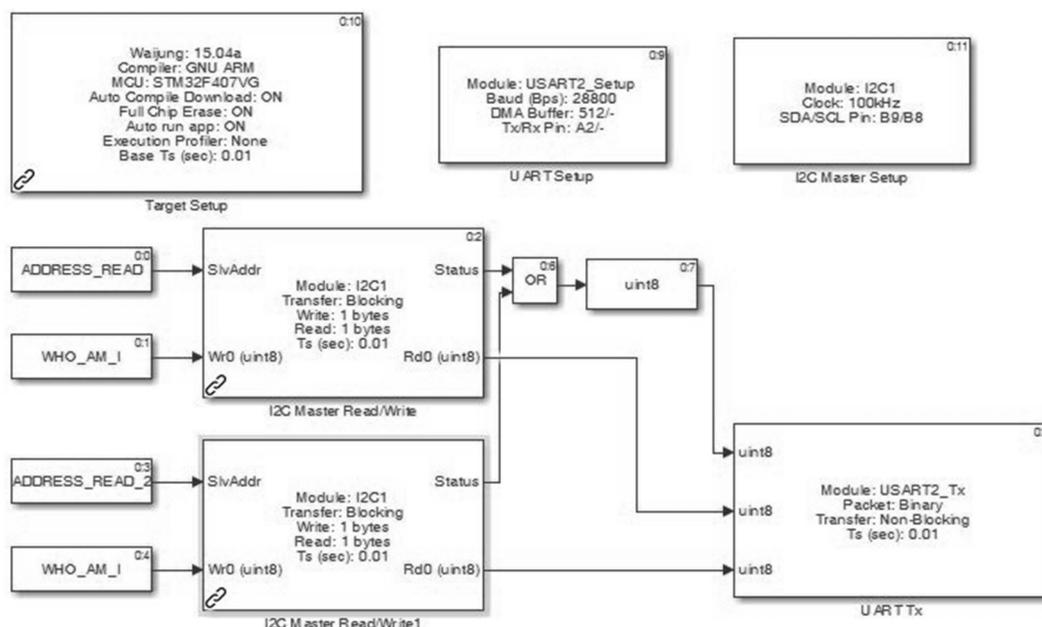


Figura 51: Prueba de IMU (Who Am I)

JULIO 2016

Una vez ya se ha comprobado que la comunicación funciona, se procede a comprobar que las lecturas del acelerómetro no son erróneas y no están desfasadas. Esto lo logramos dibujando una línea recta desde el eje de rotación del péndulo a su esquina opuesta, y situamos un transportador de ángulos tal que su centro de coordenadas también coincida con el eje de rotación del péndulo. Cargando el programa, se comprueba si el ángulo real coincide con el de las lecturas de la IMU.

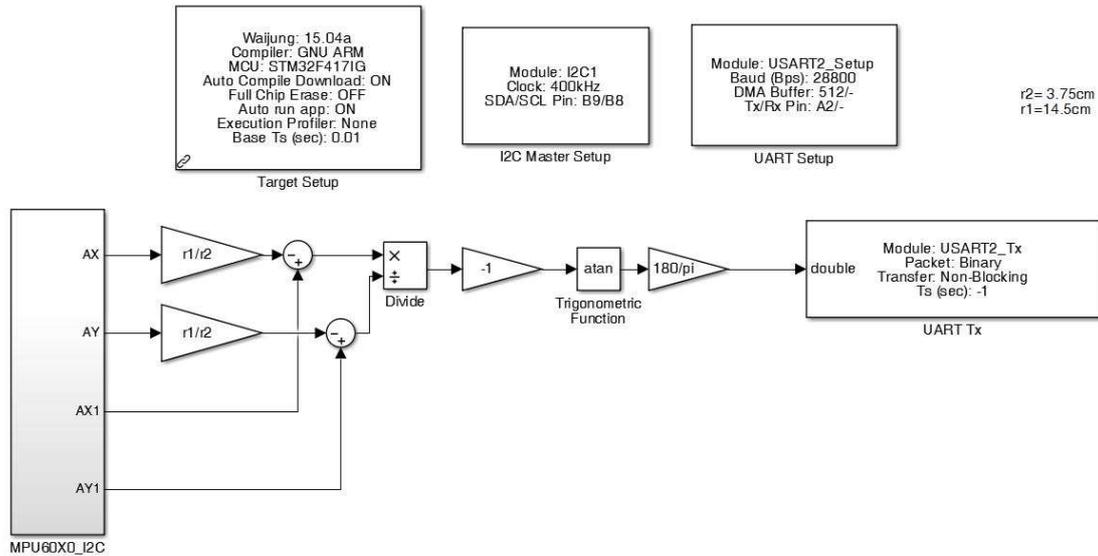


Figura 52: Prueba de IMU (lectura de ángulo)

Una vez ya se ha conseguido que las lecturas del ángulo sean correctas, se realizan los mismos pasos para comprobar que el giróscopo funciona correctamente.

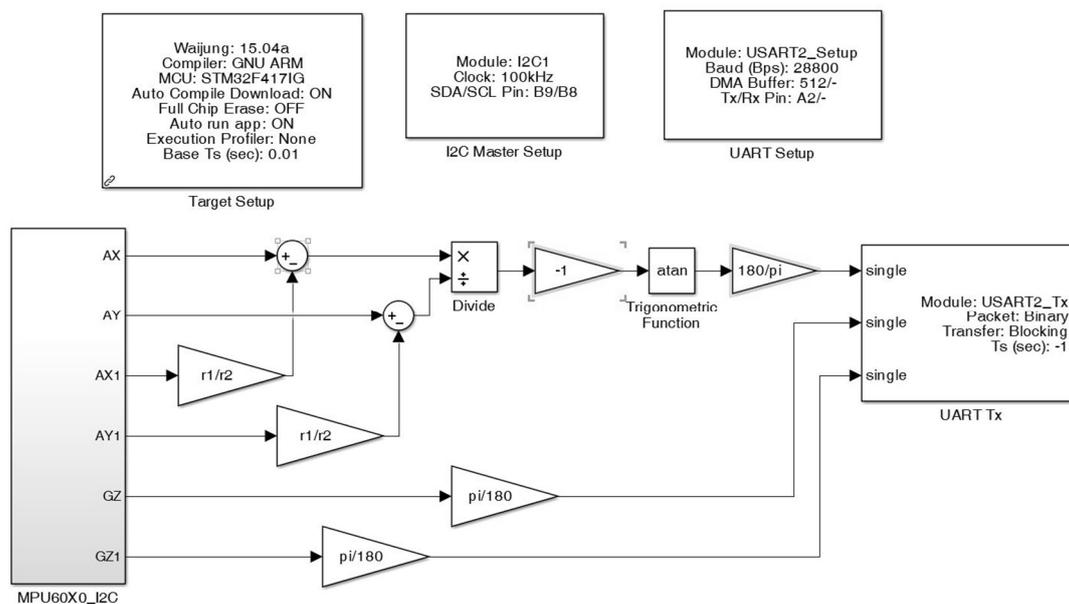


Figura 53: Prueba de IMU (lectura de ángulo y velocidad angular)

8.3.4. Máquina de estados

Es importante mencionar, que esta prueba no es para mejorar el control, eso se hace aparte, aunque más adelante, una vez esté el control optimizado, sí que se piensa usar esta máquina de estados para controlar todo el proceso. Este ensayo tiene dos utilidades, comprobar que la unión de los módulos y sus transiciones funcionan correctamente, y buscar la velocidad angular del disco óptima para realizar levantarse lo suficiente para que el control de balanceo pueda ejercer su función de manera óptima.

La puesta en conjunto del sistema real incluye partes de las pruebas anteriores combinadas con el control de balanceo y la máquina de estados, formando un sistema complejo de mostrar en un solo diagrama de Simulink. Para simplificar el uso del programa y conseguir que sea más comprensible, se divide en varios subsistemas:

- Máquina de estados: En la máquina de estados se programan las transiciones del sistema y se envían las señales que indican cambios en los actuadores. Para su visualización y programación, en un principio se planteó activar y desactivar diferentes subsistemas utilizando un bloque “enable”. El problema que este método conllevaba, era el efecto rebote del pulsador, hacía que se saltara estados o se activaran varios a la vez. La solución, utilizando “If” hacía que el código no estuviera optimizado, por lo tanto se hace uso del bloque “Stateflow”. Este bloque, es un bloque muy intuitivo para diseñar una máquina de estados utilizando formas (flechas y rectángulos).

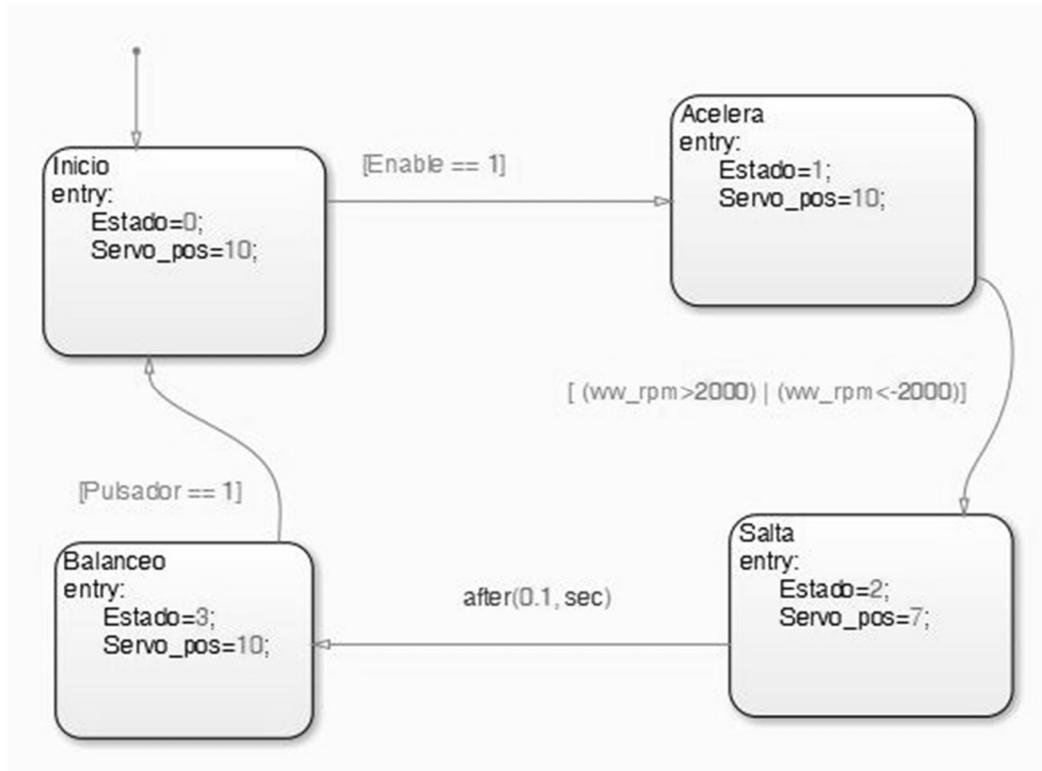


Figura 54: Implantación de máquina de estados

Las flechas indican las transiciones, y en los cuadrados los cambios de valores de algunas variables ya sea al entrar, salir o durante el estado. Las entradas y salidas indicadas en el apartado 6.3 no son las que se usan explícitamente en la máquina de estados de Simulink, para su implantación, era más fácil usarlo de la siguiente manera:

- Entradas: “Enable”, pulsador y la velocidad angular del volante de inercia, y el temporizador escribirlo como transición directamente (“after(0.1, sec)”).
 - Salidas: La posición del servo y el estado actual, y ya dependiendo del estado, el resto de subsistemas actuarán de una manera o de otra activando aquello que se denominaba salidas en el apartado 7.3.
- Lectura de entradas: El subsistema de lectura de entradas contiene todos los bloques o conjuntos de bloques utilizados en las pruebas para recoger las medidas de los diferentes sensores o pulsadores. Las salidas se conectan con los otros subsistemas mediante los bloques “goto” y “from”. Las entradas son el “enable”, la ω_w [rpm], el θ_b [rad] y la ω_b [rad/s]. Las últimas tres son las variables de estado del sistema, y la primera, el “enable”, es la señal de habilitación que permite que el sistema arranque una vez se hayan calibrado el motor y los giróscopos.

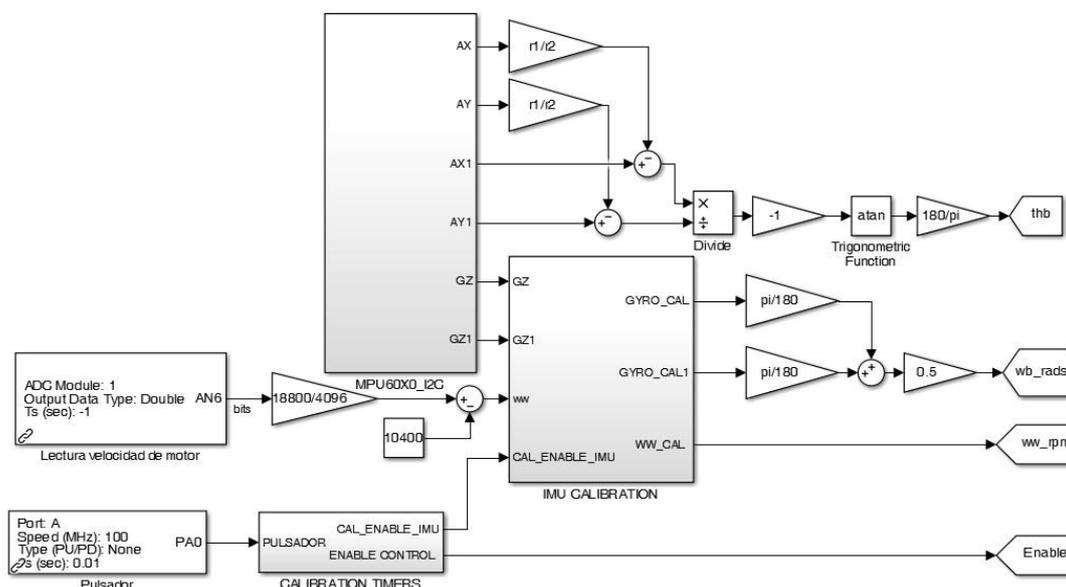


Figura 55: Implantación de la lectura de entradas

- Salidas: En subsistema de salidas, se controla la tensión de referencia que se envía al motor para controlar el mando (la intensidad del motor) y la posición del servo. Esto se consigue utilizando un “switch”. La variable de control del switch es el estado, cuando el estado es mayor que dos (solamente hay un estado mayor que dos), actúan las ganancias del control de balanceo, haciendo que el péndulo se equilibre sobre el ángulo de referencia. En el caso de que el estado sea diferente del de control (inicio, aceleración y salto), se crea una función llamada “Tension_motor_ref”, en la cual se especifica lo siguiente:
 - o Si el estado es 1 (aceleración), y el péndulo se encuentra en un ángulo negativo, acelera el volante al máximo en dirección de las agujas del reloj.
 - o Si el estado es 1 (aceleración), y el péndulo se encuentra en un ángulo positivo, acelera el volante al máximo en la otra dirección.
 - o Si el estado es diferente de 1 (inicio o salto), no dar corriente al motor.

JULIO 2016

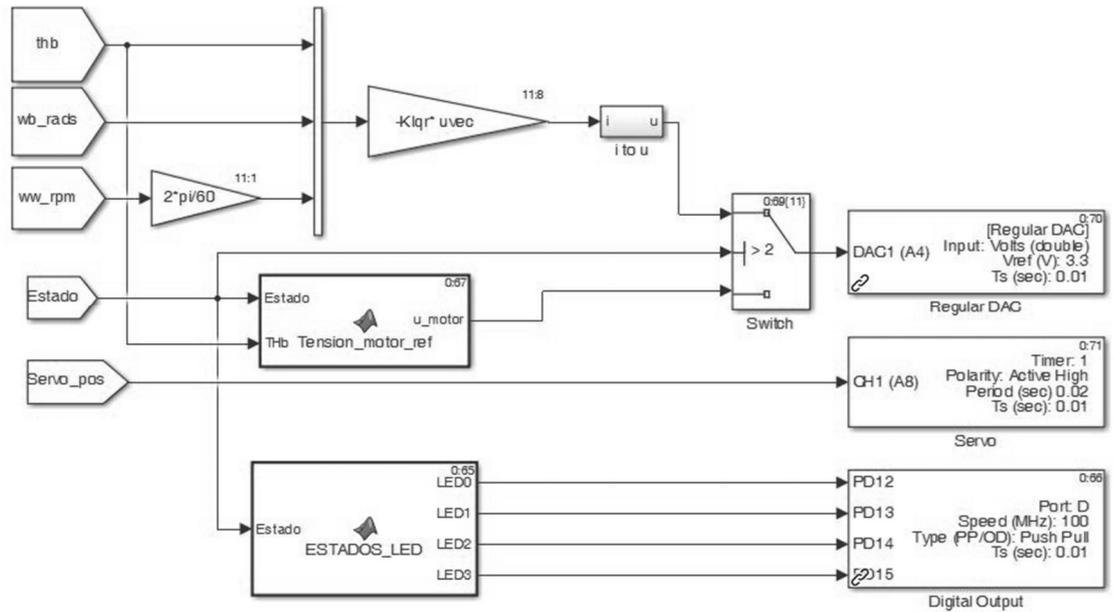


Figura 56: Implementacion del subsistema de salidas

El servomotor queda controlado solamente por la variable “servo_pos”, que como se menciona anteriormente se modifica dentro del subsistema “stateflow”.

Por último, se añaden cuatro salidas que son los LED, cada LED indica un estado. Al ser cuatro LEDs, se podría haber utilizado solamente dos e indicar el estado actual mediante código binario, pero queda más claro de esta forma.

Juntando los tres subsistemas y añadiendo, los bloques de configuración de la placa y de la comunicación I2C, queda el diagrama de Simulink mostrado en la figura 57.

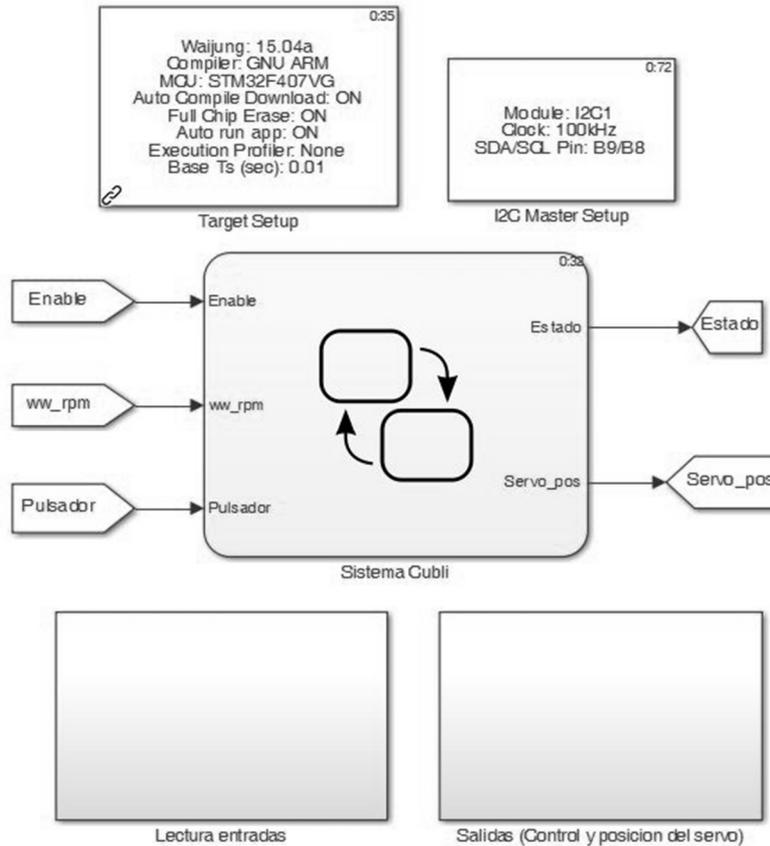


Figura 57: Implementación del sistema total

8.4. Prueba del control LQR

La prueba del control LQR consiste en la combinación algunas de las entradas y salidas de las variables de la máquina de estados. De entradas al control se seleccionan las variables de estado $(\theta_b \dot{\theta}_b \dot{\theta}_w)$, y de salida solamente se coge el mando, la intensidad del motor, formando el siguiente diagrama:

JULIO 2016

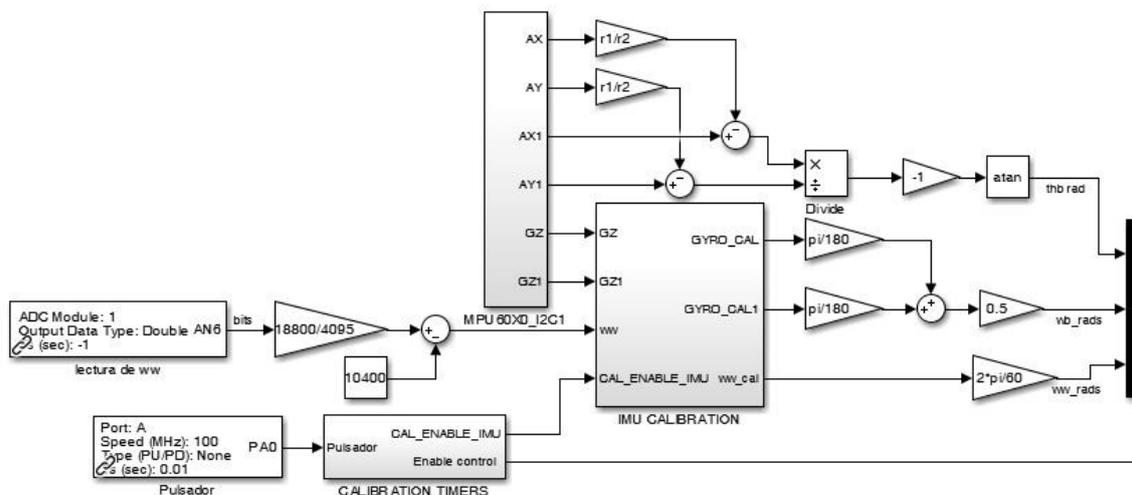


Figura 58: Lecturas del control LQR

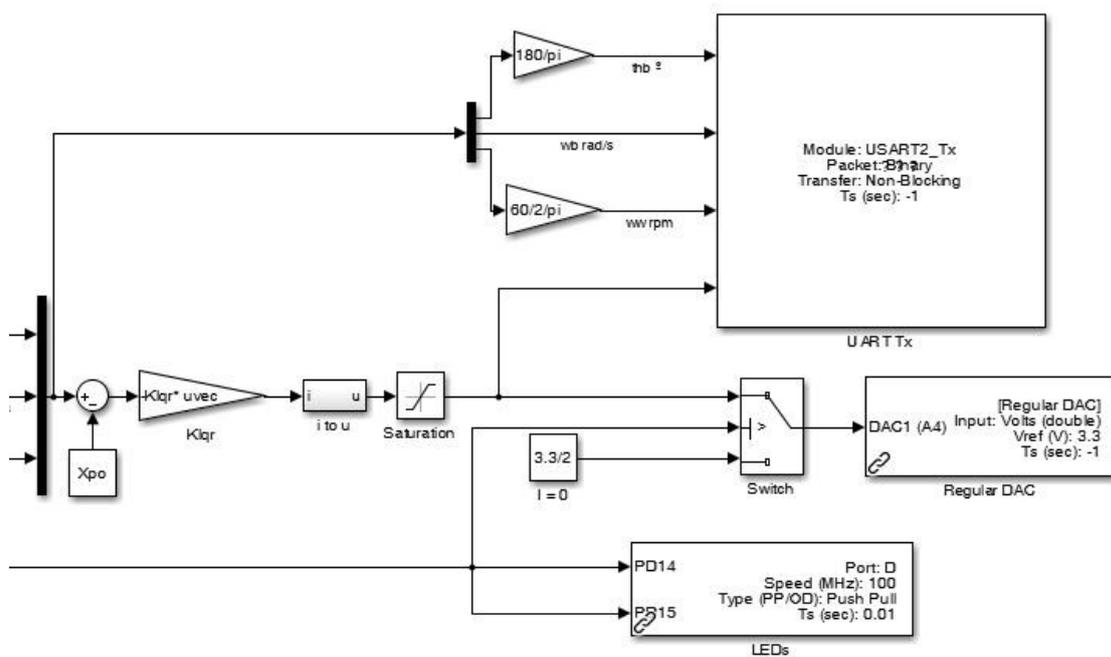


Figura 59: Salidas del control LQR

Como se puede comprobar, en las lecturas no cambia nada respecto a la máquina de estados, son las mismas ya que todas son necesarias, tres son las variables de estado, y la cuarta es el habilitador para el control, para que no se active ninguna salida hasta que se haya calibrado el sistema.

En cuanto a las salidas, en este caso es interesante enviar los valores de las variables de estado, y del mando del sistema para poder analizarlos y compararlos con los resultados de las

simulaciones. También se puede comprobar, que cuando el temporizador no ha terminado de contar para calibrar, el mando que se envía al motor es de 1.5 V, que significa intensidad igual a 0 A. También se incluyen dos LEDs que se encienden cuando el sistema está calibrado para indicar que el control ya está activado.

Puesto que el control está diseñado con un sistema no lineal para controlar la corriente del motor directamente, después de la ganancia del control, es necesario convertir la corriente en tensión para poder enviar el valor del mando como variable analógica. El pequeño subsistema utiliza la siguiente transformación:

$$V = \frac{V_{cc}}{2 * K_i} * I + K_i \quad (28)$$

Donde:

- $V_{cc} = 3.3 \text{ V}$
- $K_i = 1.36 \text{ A}$ (Máxima corriente capaz de aguantar el motor)

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 9: Resultados

9.1. Resultados de las simulaciones

Para comprender mejor el funcionamiento del sistema se han realizado varias simulaciones. En éstas se han tomado distintos parámetros para su mejor representación, y se mostrarán únicamente los parámetros más significativos.

9.1.1. Equilibrio desde ángulo inicial pequeño

La primera simulación es la estabilización de un ángulo en 0° , definiendo como condiciones iniciales un ángulo pequeño tal que $\theta_b = 2.5^\circ$. Para ello se ha obtenido un control que logre estabilizar el péndulo rápidamente, haciendo que el sobrepaso no sea mayor del 15%.

El primer control se realiza dando el mismo peso a todas las variables de estado y el mando. La matriz de ganancias y el resultado obtenido se muestran en la siguiente figura:

$$Klqr1 = (-396.41 \quad -65.39 \quad -0.69)$$

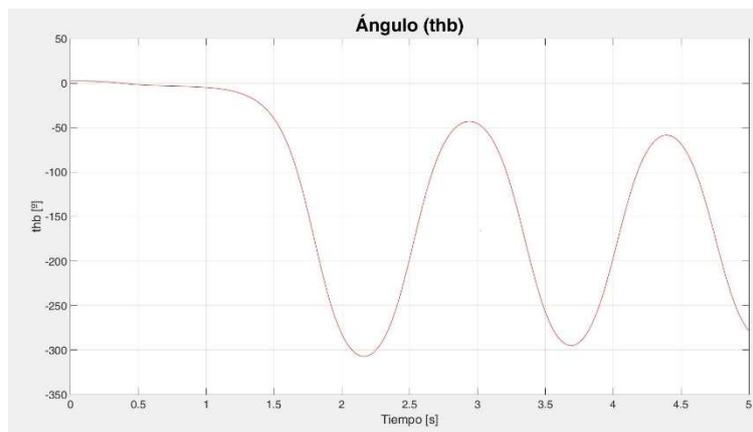


Figura 60: Resultado de la simulación utilizando mismo peso para las variables de estado y mando

El siguiente paso lógico es dar más peso a la variable que se pretende estabilizar, en este caso el ángulo, por lo que se procede a repetir el control dando esta vez un peso de 100 al ángulo y mantener en 1 el resto de variables.

La matriz de ganancias obtenida es igual que en el primer control, por lo que la respuesta será la misma. Se observa que al cambiar únicamente el peso de la variable θ_b , la matriz se

JULIO 2016

mantiene constante. Es remarcable el hecho de que no por darle mayor peso a la variable a estabilizar, el control será más eficaz.

El siguiente paso es aumentar también el peso a la velocidad angular del péndulo. Esta velocidad es importante, ya que si se controla mejor, es más fácil estabilizar el ángulo. Se da un peso de 1000 θ_b y a ω_b .

$$Klqr1 = (-390.28 \quad -67.48 \quad -0.64)$$

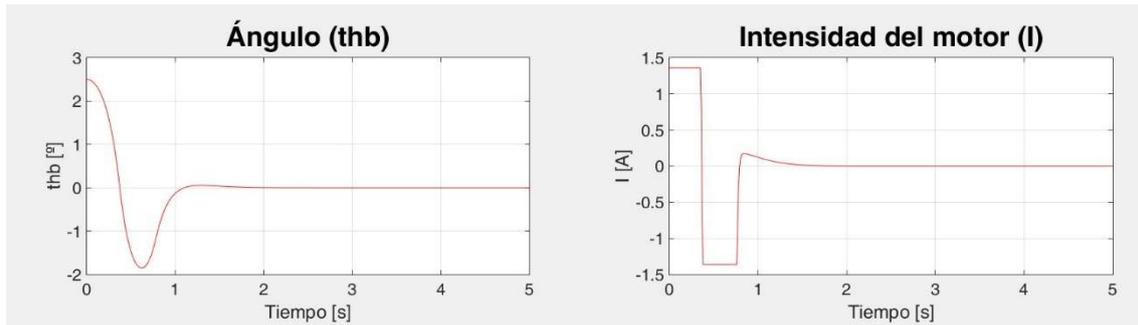


Figura 61: Resultado de la simulación dando peso 1000 a θ_b y a ω_b

Los valores de la matriz de ganancias no han obtenido una variación notable, pero se observa que aumentar el peso sobre la velocidad angular del disco mejora el control. A pesar de haber conseguido estabilizarse, el sobrepaso en el ángulo es muy elevado, casi del 67%, y el mando satura durante prácticamente todo el transitorio. El control todavía es mejorable.

La nueva modificación en el control será dar mayor peso al mando debido a la continua saturación del mismo en el caso anterior. Se mantienen los pesos sobre las variables de estado respecto de la última simulación y se aumenta el del mando a 50.

$$Klqr3 = (-112.79 \quad -18.95 \quad -0.12)$$

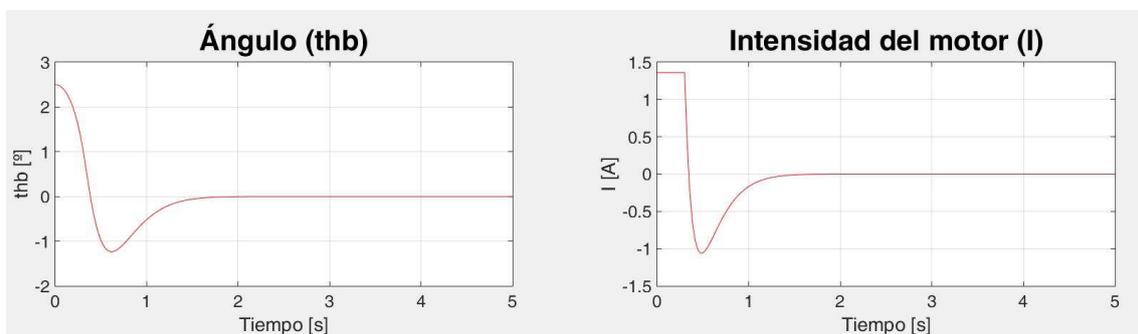


Figura 62: Resultado de la simulación dando peso 1000 a θ_b y a ω_b y peso 50 a la I

Se observa que al aumentar el peso sobre el mando, ha disminuido mucho los valores de la matriz de ganancias del control. Este resultado, por ahora, es el mejor hasta el momento, se ha

conseguido reducir el sobrepaso hasta un 50% y la intensidad satura durante la mitad de tiempo, (aunque) por otro lado la respuesta se ha vuelto un poco más lenta, pero no lo suficiente como para ser alarmante.

La velocidad del péndulo, teóricamente está controlada por las provocadas por el disco. En el siguiente ensayo se prueba a disminuir el peso de la ω_b 1 de nuevo y aumentar el de la ω_w a 1000. El resultado es:

$$Klqr4 = (-728.22 \quad -120.20 \quad -1.35)$$

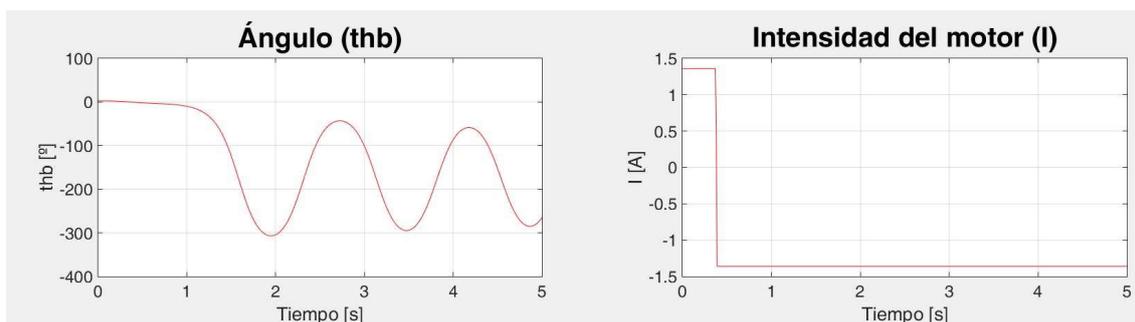


Figura 63: Resultado de la simulación con peso 1000 en θ_b y ω_w y peso 50 en la I

La matriz de ganancias ha aumentado al doble respecto a los primeros resultados, y la señal vuelve a ser inestable. SE observa una mejoría en el control cuando el peso de las variables θ_b , ω_b y el mando es mayor, y la matriz de ganancias tiene valores más pequeños.

En el próximo ensayo se aumenta a 1000 el valor del peso en el mando:

$$Klqr5 = (-62.48 \quad -10.26 \quad -0.03)$$

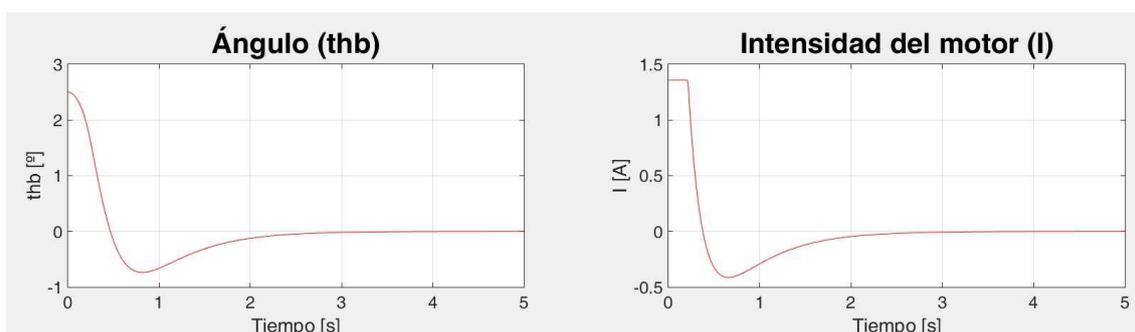


Figura 64: Resultado de la simulación con peso 1000 en θ_b , ω_w e I

Los valores obtenidos de la matriz de ganancias son los más pequeños hasta el momento; ocurre lo mismo con el sobrepaso (30%) y la duración de la saturación de la intensidad. Hasta el momento son los mejores resultados obtenidos. Por otro lado parece que la respuesta se vuelve

JULIO 2016

algo más lenta, pero aun así, el tiempo de establecimiento sigue por debajo de dos segundos, lo cual se sigue considerando óptimo.

Analizando los resultados anteriores, parece que aumentando todos los pesos excepto el de la ω_w el control mejora, por lo tanto en el siguiente ensayo se prueba eliminar completamente el peso sobre esta variable.

$$Klqr6 = (-48.11 \quad -7.90 \quad -0.0017)$$

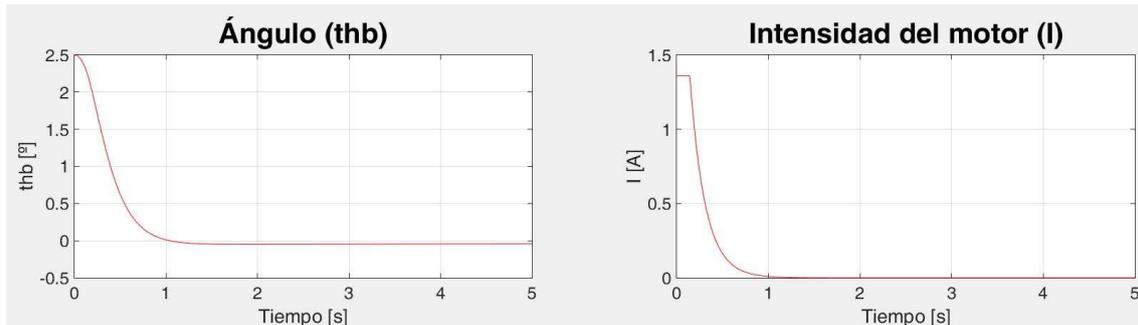


Figura 65: Resultado de simulación eliminando el peso de ω_w

Eliminando el peso sobre el valor de la velocidad angular del disco de inercia se elimina completamente el sobrepaso, (0%), y la duración de la saturación de la intensidad del motor es mínima. Realmente si se hiciera zoom sobre el gráfico, se podría comprobar que el ángulo no llega a estabilizarse nunca, pero es un valor tan pequeño para el ojo humano que se considera aceptable.

Después de haber realizado varios ensayos, parece que los mejores controles son aumentando el peso sobre todas las variables menos en la velocidad angular del volante de inercia. Las ganancias con las que se va a realizar estas pruebas son Klqr3, por ser la más rápida que se mantiene estable con un sobrepaso menor del 50%, Klqr6, por ser la que menor sobrepaso tiene, y Klqr5, por ser un valor intermedio entre las dos.

Aumentando el ángulo primero de 0.5 en 0.5 grados, después de 0.1 en 0.1, y por último de 0.01 en 0.01, se llega al resultado de que el máximo ángulo que aguanta el control más robusto es 3.14°. El control que consigue esto es el de Klqr6, el de menor sobrepaso. La ganancia Klqr5 es capaz de estabilizar desde 3.13° y la ganancia Klqr3 desde 2.9°.

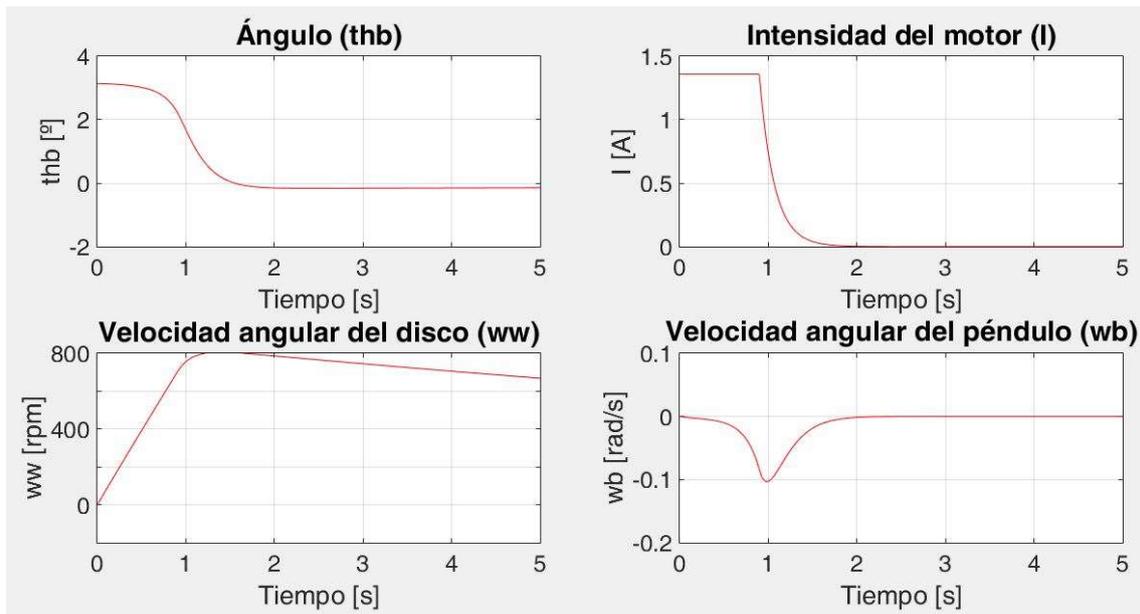


Figura 66: Resultado de simulación para máximo ángulo con la matriz $Klqr6$

Este control es el más robusto, el ángulo límite es 3.14° . Sin embargo, en el sistema real, es posible que no sea capaz de mantener su equilibrio en una esquina debido al ruido en las lecturas de las variables. Las diferencias entre los tres controles probados son mínimas, por lo que se tendrán otros factores en cuenta para las próximas simulaciones. Es interesante valorar la respuesta del control ante condiciones iniciales que simulen la recepción del salto, para decidir qué control es más apropiado.

Analizando las velocidades angulares se observa el comportamiento de las variables ω_b y ω_w . La velocidad angular del péndulo sigue una tendencia lógica, primero se acelera para intentar recuperar el ángulo y volver a 0° , y después decelera para equilibrarse con suavidad. El comportamiento de la velocidad angular del volante de inercia es peculiar, se acelera durante el período de equilibrio y se decelera lentamente debido a que la corriente es nula al haber alcanzado el punto de estabilidad. Es posible que sea por la fuerza de rozamiento del eje del motor, por la que el péndulo no llega exactamente a 0° , si no que se queda unas décimas fuera.

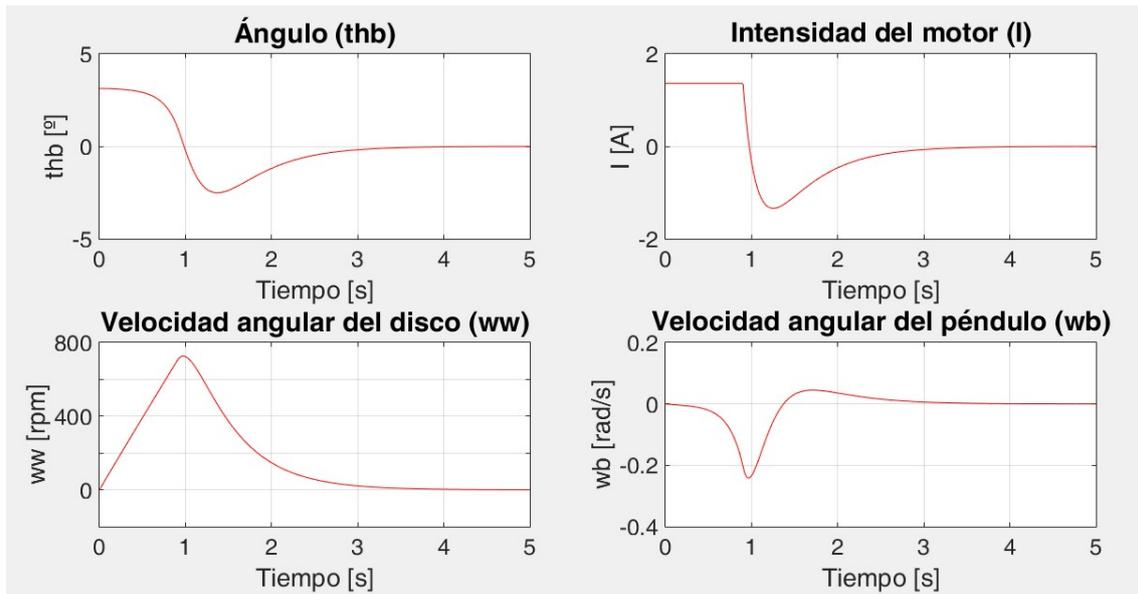


Figura 67: Resultado de la simulación para el ángulo máximo del control 5

La hipótesis planteada acerca del comportamiento de la velocidad angular del volante de inercia se ve reflejada en la figura 67. El sobrepaso del ángulo es debido al intento de devolver la velocidad angular del volante de inercia a su estado de reposo. Si se deja libre, una vez se alcanza un ángulo muy cercano a 0°, el péndulo no sufre inestabilidades y se equilibra mejor.

Para el objetivo del proyecto, el control más efectivo de los simulados ha sido el 6, aun no estabilizando en 0° exactos. Si el objetivo fuese una máxima precisión, la matriz de ganancias más apropiada sería la 5.

9.1.2. Equilibrio en recepción de salto

Para imitar las condiciones de la recepción de un salto, como condiciones iniciales se utiliza un ángulo de 10°, y lo que se buscará es qué rango de ω_b es capaz de aguantar cada control.

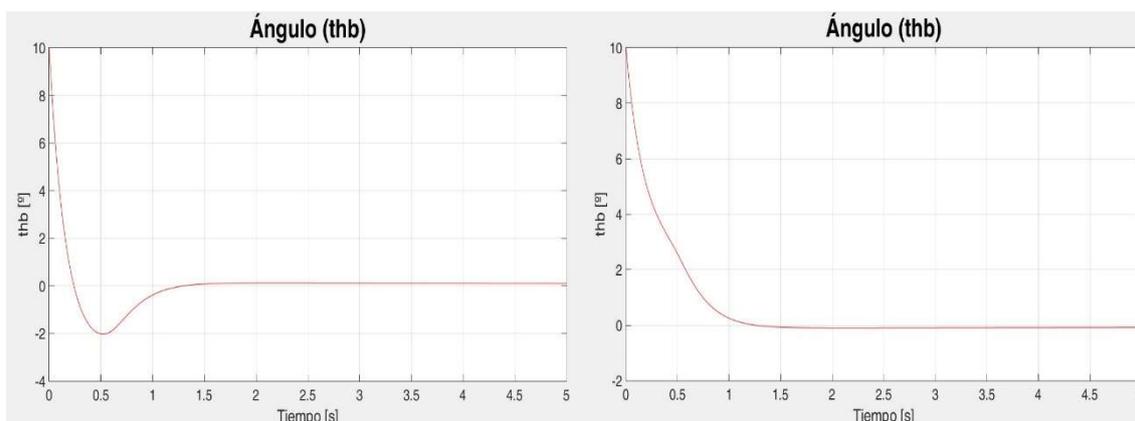


Figura 68: Resultado de simulación de salto para K1qr6 (izquierda $\omega_b = -1.4$ rad/s, derecha $\omega_b = -0.74$)

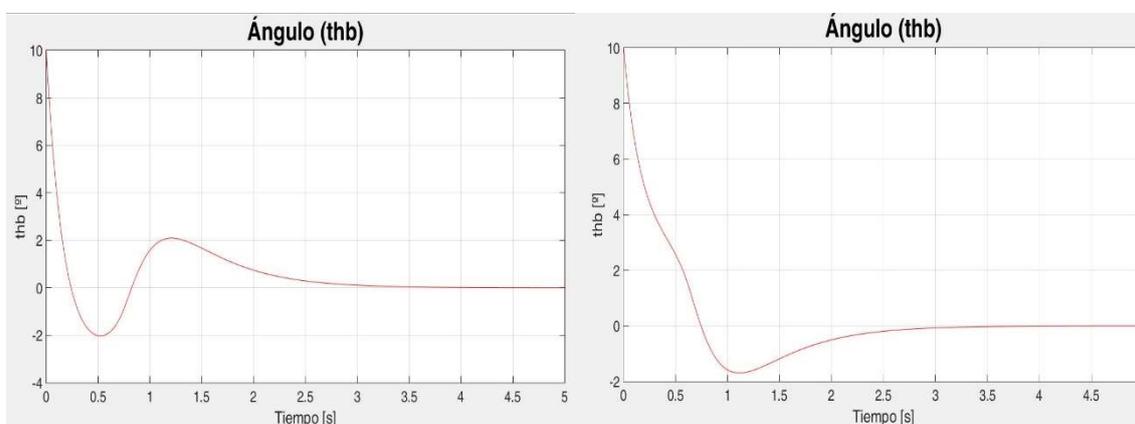


Figura 69: Resultado de simulación de salto para K1qr5 (izquierda $\omega_b = -1.4$ rad/s, derecha $\omega_b = -0.74$)

En las dos figuras anteriores, se comparan como responden al salto las matrices de ganancias 5 y 6, la 3 no está incluida ya que el rango de velocidades angulares provenientes del salto que consigue equilibrar es menor.

Realizando una comparativa de los dos mejores controles, parece claramente que la respuesta del control 6, aquel que no tiene ningún peso sobre la ω_w es mucho más estable. El control número 5, para ser capaz de equilibrar el péndulo da muchos más tumbos antes de estabilizarse.

En conclusión, modificando los pesos de las matrices provenientes de la ecuación de Riccati para hallar la matriz de ganancias óptimas afecta de la siguiente manera:

- Θ_b (Q11): Modificando el peso sobre esta variable, no parece afectar mucho al resultado del control. Para notar algún efecto significativo debe ser al menos tres órdenes de magnitud diferente del resto. Al aumentar este peso se disminuye el sobrepaso y el tiempo de establecimiento.

JULIO 2016

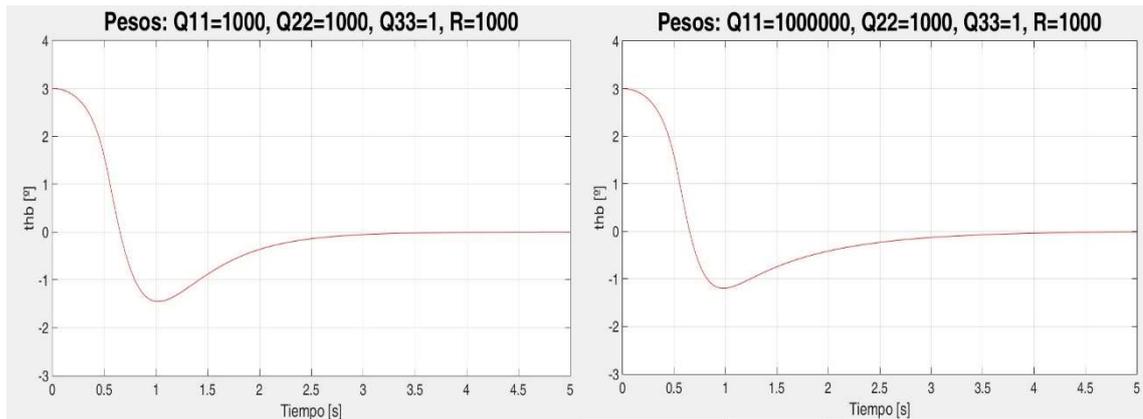


Figura 70: Comparación del efecto de la modificación del peso del control sobre el ángulo

- ω_b (Q22): Modificando el peso sobre la velocidad angular del péndulo los efectos son más notables que en el caso del ángulo. Al aumentarlo, el sobrepaso disminuye y los tiempos aumentan ligeramente.

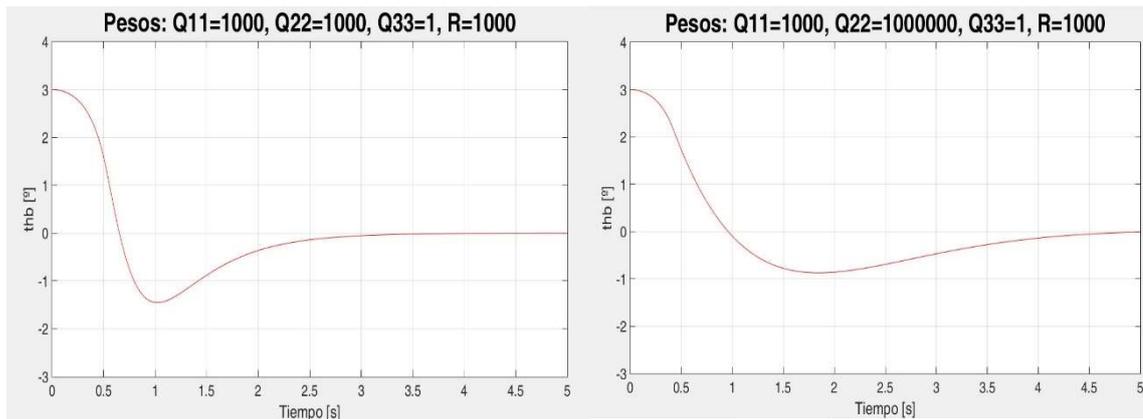


Figura 71: Comparación del efecto de la modificación del peso del control a la velocidad angular del péndulo

- ω_w (Q33): El peso de la velocidad angular del disco sí que es muy significativo. Si se aumenta, o se mantiene al nivel del resto de variables, la respuesta se convierte en inestable. Conviene configurarlo al menos dos órdenes de magnitud por debajo del resto. En el caso de que se elimine por completo, el sobrepaso se minimiza el máximo posible.

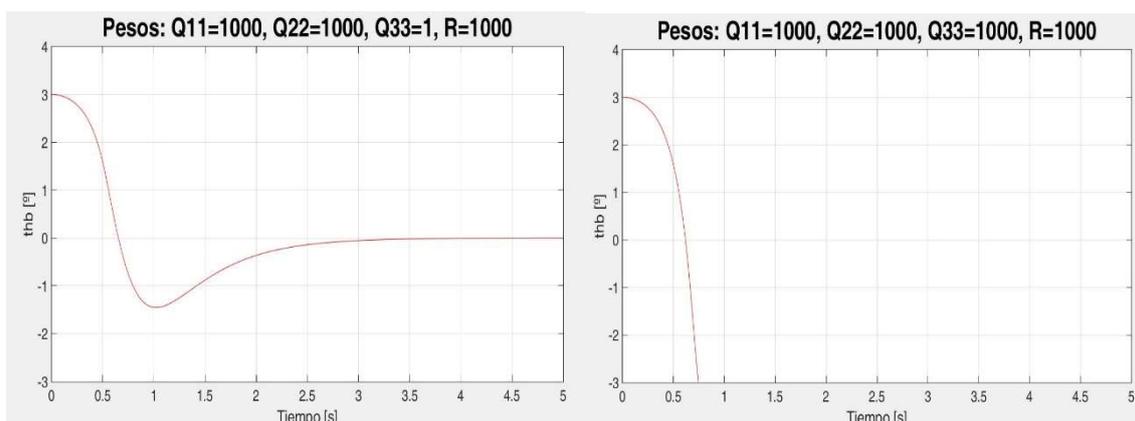


Figura 72: Comparación del efecto de la modificación del peso del control a la velocidad angular del disco

- I (R): El peso sobre el mando parece ser también muy importante. Si el control no se centra en intentar no saturar la corriente del motor, va dando tumbos de un límite a otro, provocando que el control no sea capaz de equilibrar el péndulo. Conviene situar este peso en el mismo orden de magnitud que el peso del ángulo.

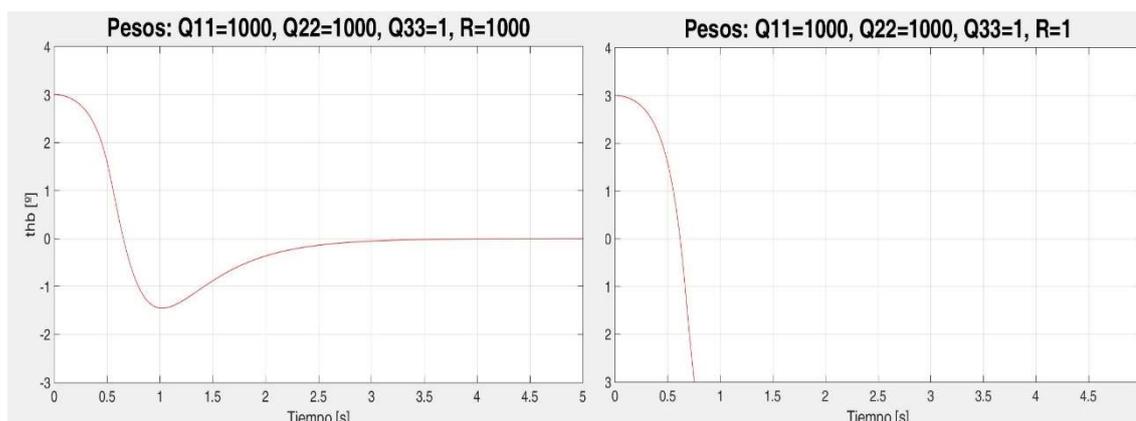


Figura 73: Comparación del efecto de la modificación del peso del control a la corriente del motor

9.1.3. Aumento del par motor

Haciendo un análisis de los resultados anteriores, no parece que el control, incluso siendo óptimo, sea especialmente bueno. Analizando los parámetros de la planta del sistema, manteniendo todo igual, tiene que haber algo relacionado con el mando o con los actuadores que pueda mejorar el control sin tener que modificar el mucho el diseño.

El parámetro que parece más propenso a mejorar el control es la constante del par motor (K_m). Esta constante es proporcional a la corriente del motor. Un mayor par, provocaría un

JULIO 2016

aumento en las fuerzas que van contra la gravedad para vencer la energía potencial. Esta idea surge de que en ambos proyectos ya existentes sobre el cubli, utilizan motores más potentes que los utilizados en este proyecto.

El ensayo que se realiza a continuación es, suponiendo el resto de variables constantes, se dobla la constante del par motor.

$$Klqr_aumento_Km = (-24.07 \quad -4.03 \quad -0.0009)$$

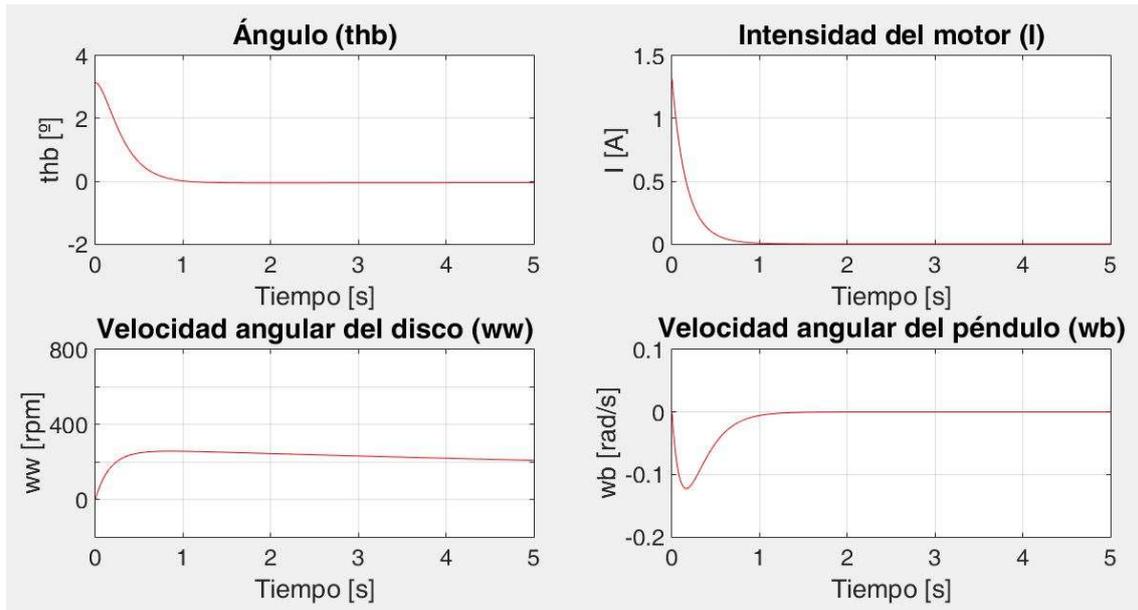


Figura 74: Resultados de simulación aumentando la constante del par motor respecto a la del control número

6

Lo primero que llama la atención es que la matriz de ganancias se ha disminuido en aproximadamente la mitad. Este valor de $Klqr$, con los parámetros anteriores habría sido inestable. Observando la gráfica, se puede comprobar que la respuesta es mucho más suave, y la corriente no satura. El siguiente paso es probar desde qué ángulo se puede equilibrar con estos nuevos parámetros.

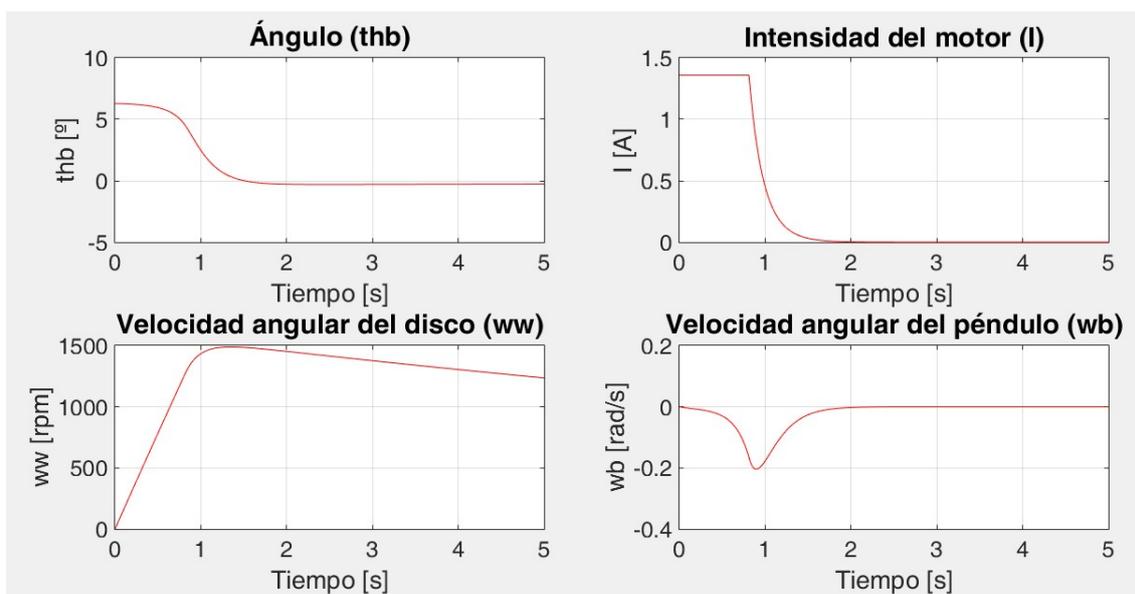


Figura 75: Resultados de la simulación con el par aumentado maximizando el ángulo hasta 6.28°

Utilizando los nuevos parámetros, consigue balancear un ángulo de 6.28° . Si se analizan estos resultados en profundidad, se comprueba que al haber doblado el par del motor, manteniendo constante el resto de parámetros, se consigue equilibrar un ángulo el doble de grande, la velocidad angular del disco aumenta hasta el doble, y ocurre lo mismo con la velocidad angular del péndulo. Lo que no cambian son los tiempos.

Este resultado es muy importante, ya que es clave para mejorar el control utilizar uno más potente. Es cierto que utilizando uno más grande, no significa que el control mejore el doble, ya que también significa que otros parámetros como el peso, rozamientos, o distancias a centro de gravedad cambiarían, pero es muy probable que mejoraría.

9.2. Resultados de las pruebas

Los resultados de las pruebas sirven para identificar errores y corregirlos en cada módulo por separado antes de juntarlos. Para ser capaces de analizar estos resultados, se hace uso de comunicación serie a través de UART con el ordenador, que transmite las lecturas de las diferentes variables y las guarda en el “workspace”.

JULIO 2016

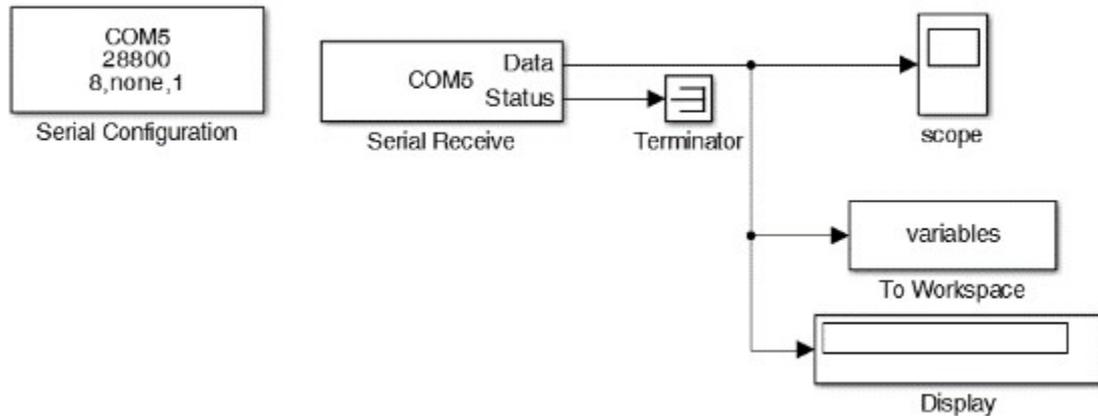


Figura 76: Lectura de variables de interés mediante la comunicación a través de UART

Los bloques que se utilizan para estas lecturas son:

- Serial configuration: En este bloque se configura el formato de los datos y la velocidad de transmisión (“baudrate”). Para que la comunicación sea exitosa los parámetros tienen que coincidir con aquellos de envío, y por lo tanto se envían paquetes de 8 bits sin paridad y un bit de parada.
- Serial receive: Este bloque es el que recibe la información. Es necesario indicar el tamaño del vector de datos que recibe además del formato de los datos (int, double...) y el tiempo de muestreo. En general se han utilizado “double” en casi todos los casos, y el tiempo de muestreo es siempre 0.01 segundos.
- Scope: Para las variables que se representan en función del tiempo. Con este bloque se ve de forma gráfica como evolucionan estas variables.
- To workspace: En algunos casos, para representar de una manera más visual, o eliminar el ruido y ser capaz de analizar mejor los datos realizando un tratamiento especial, es necesario guardar las series temporales en el “workspace” de Matlab.
- Display: Muestra el valor actual de la variable. Es más útil para variables cuyo valor no varía mucho o nada con el tiempo.

El mayor problema de la lectura de datos con la comunicación serie, es que estos llegan a su destino con bastante ruido. Por esta razón es necesario para ser capaces de analizar bien los datos obtener mejores conclusiones, realizar un procesamiento de datos específico para eliminar valores anormales y tratar de disminuir al máximo el ruido. El tratamiento consta de los siguientes pasos:

1. Separar del formato “timeseries” los datos y el tiempo en diferentes variables. Para esto, se utilizan dos comandos:

- a. Variable_timeseries.Data
- b. Variable_timeseries.Time

El tiempo se guarda, de esta manera, en un vector columna donde cada fila es una unidad de tiempo. Los datos, se guardan en una matriz tridimensional en la cual la primera dimensión es el número de variables, la segunda dimensión siempre tiene solo un valor, y la tercera es el número de valores que contiene esa variable.

- 2. Realizar un bucle en la matriz de datos para separar las variables en un vector columna cada una. Aprovechando el bucle, se eliminan valores anormales.

La manera de eliminar estos valores es si el valor absoluto de la diferencia entre el valor i y el valor $i-1$ es mayor que un rango determinado (depende de qué variable se esté mirando en ese momento), se sustituye el valor i por el valor $i-1$. Este es el primer filtro que sirve para eliminar ruido.

- 3. Creación de un filtro de medias móviles [27]. El filtro de medias móviles es un filtro común en el procesamiento de señales digitales. Es un filtro óptimo para la eliminación del ruido aunque manteniendo una respuesta a un escalón suficientemente nítida.

Su funcionamiento es muy sencillo, cambia el valor de la lectura de una señal por la media de los últimos n valores, siendo n el número de valores que uno quiera.

El filtro de medias móviles se aplica a la lectura de las señales del ángulo medido por los acelerómetros, la velocidad angular de los giróscopos y la velocidad angular de los motores medida por los sensores hall.

- 4. Elaborar gráficos con los datos para analizar correctamente los resultados. Algunos gráficos serán solamente de las señales por separado y otros de comparación de señales según lo que se quiera analizar en cada caso.

9.2.1. Servomotor

Probando diferentes posiciones del servo se llega a la conclusión de que las más óptimas son las que se muestran en la figura 77.

En la imagen de la izquierda se ve la posición de reposo. Se utiliza esta posición en vez de 90° para que en el momento en el que se habilite la señal que manda al servomotor a frenar el disco, tenga una menor distancia que recorrer pero permitiendo el paso del tornillo cuando no está accionada.

JULIO 2016

En la imagen de la derecha se muestra la posición de frenado. En el instante en el que el brazo del servo entra en el camino de los tornillos fijados al volante de inercia recibe un golpe en la dirección de los ejes de los engranajes. Este golpe frena en seco el disco transmitiendo su energía para saltar.

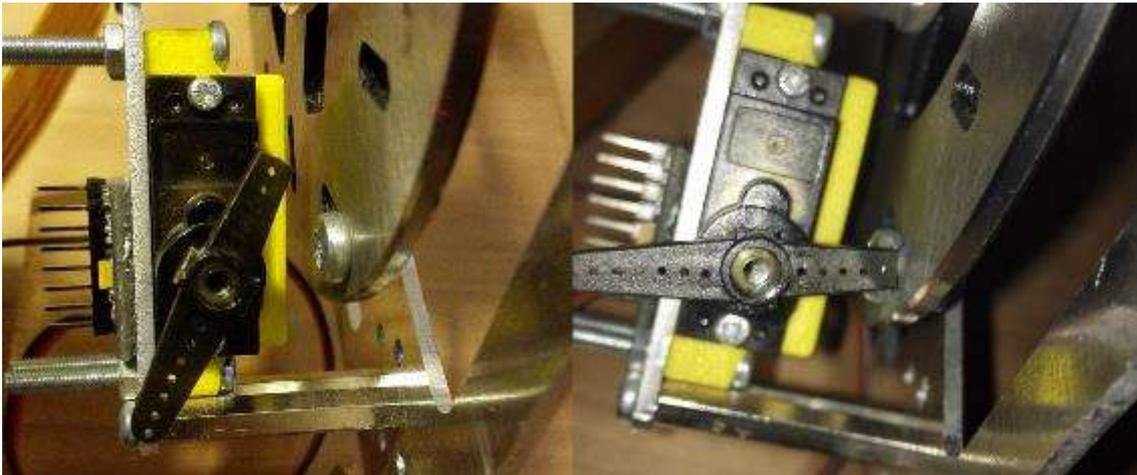


Figura 77: Posición de reposo (izquierda) y de frenado (derecha) del servomotor

9.2.2. *Motor DC*

Para el motor DC, en el procesamiento se denominan como valores anormales aquellos en los que la diferencia entre dos valores seguidos es mayor de 30 rpm, y se utiliza en filtro de medias móviles con 100 valores, equivalente realizar la media con los valores del último segundo.

En la figura 78 se muestra el funcionamiento de la calibración. En el segundo 5 se activa el comienzo del temporizador de calibración, y su duración es de 20 segundos.

Se aprecia claramente que al principio existe un offset de aproximadamente 30 rpm en el valor medio de la señal, del segundo 10 al 15 se intentó ajustar demasiado, ya que justo al comienzo de la calibración el valor de la velocidad estaba en su punto mayor, y la media por lo tanto era grande.

A partir del segundo 15 ya se podría tomar como calibrado, pero por seguridad se añaden otros 10 segundos de calibración para asegurarse de que se ha realizado correctamente.

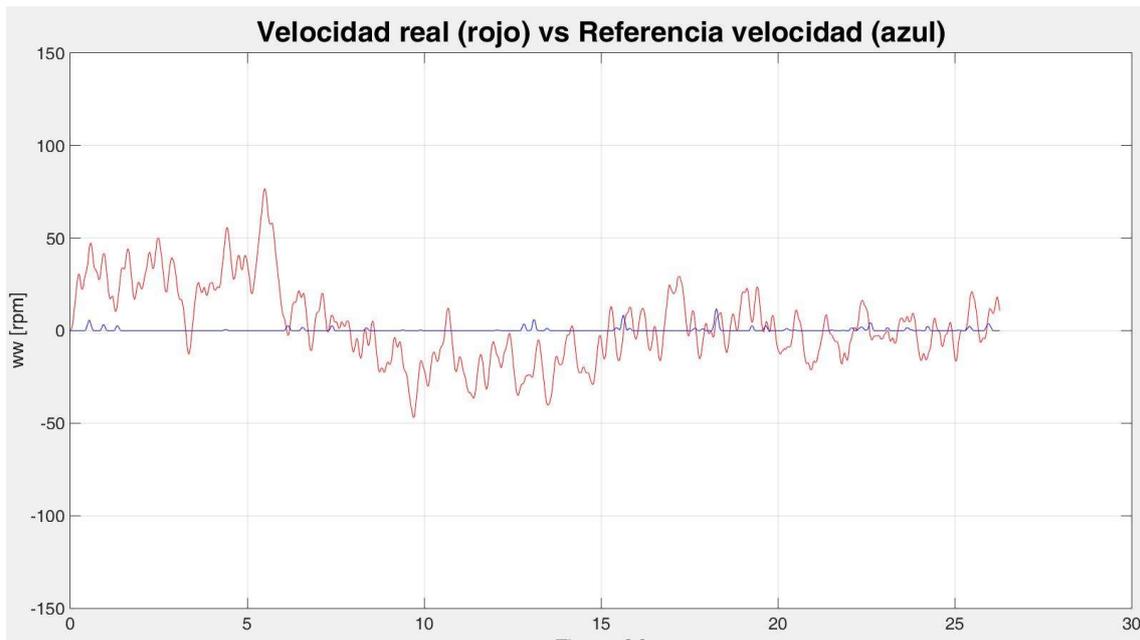


Figura 78: Calibración del offset del motor

Una vez se ha calibrado, se procede a dar aumentar la referencia de la velocidad para comparar si las medidas son acordes con la velocidad solicitada. En la figura 79 se muestran los resultados.

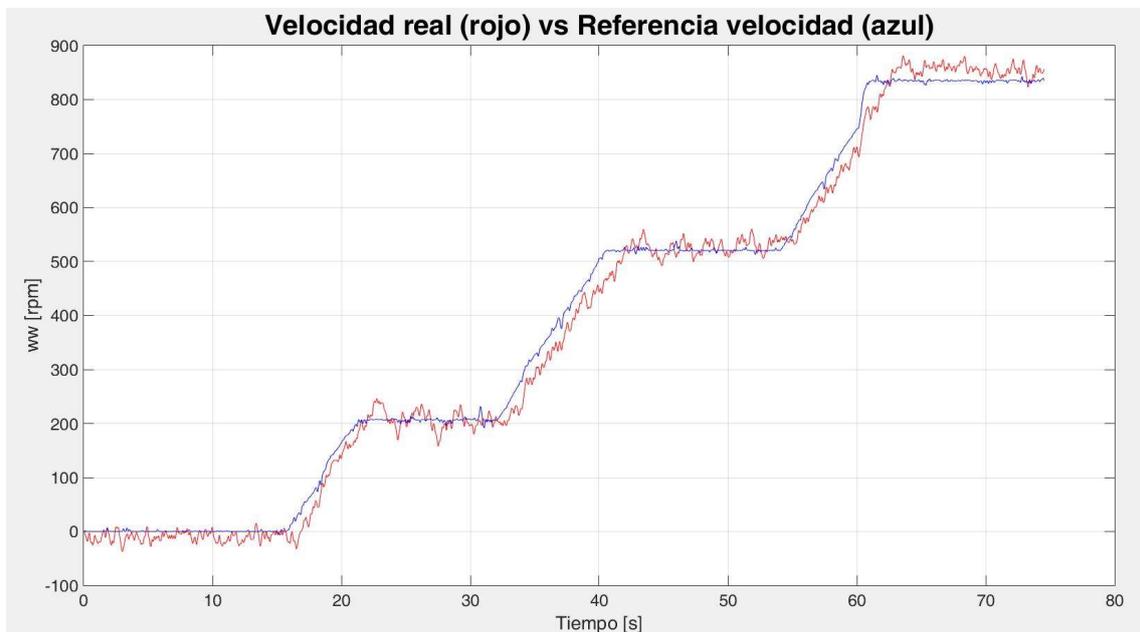


Figura 79: Comparación de la referencia de velocidad angular y la velocidad angular real del motor DC (ω_w)

JULIO 2016

Lo que más llama la atención a primera vista es que la medida de la velocidad contiene bastante ruido incluso después de haberla procesado con los filtros mencionados anteriormente. Realmente no es ruido debido a la transmisión de los datos por la comunicación serie. Cuando se referencia la velocidad directamente, sin ser el mando la corriente, el ESCON Studio utiliza un regulador en bucle cerrado. Este regulador funciona intentando ajustar la velocidad del motor a la demandada enviando impulsos de corriente. El problema surge de que los impulsos de corriente que envía son relativamente grandes incluso configurando para que dé el mínimo posible (0.173 A). Esto para velocidades pequeñas, en las cuales los momentos son también pequeños, consigue que al intentar ajustar la velocidad, el par motor sea demasiado grande y acabe dando tumbos por encima y por debajo de la referencia como se puede apreciar en la figura.

Olvidándose del ruido, lo siguiente que se aprecia es que la señal de la velocidad real está desfasada entre uno o dos segundos respecto a la de referencia. Esto es normal debido a que es lógico que primero tenga que aumentar la referencia para que el motor la siga.

Aparte de estos dos detalles, se podría mencionar que aproximadamente en el segundo 65, parece que el valor real sobrepasa a la referencia. Esto podría ser a que en el segundo 60 se aumenta la referencia con un mayor gradiente y por lo tanto el motor recibe una mayor aceleración. Esto implica que se pasa de la referencia y luego, debido a los filtros aplicados, y el control de velocidad lento del ESCON Studio, tarde más en recuperar esa velocidad. Los resultados de la prueba del motor son satisfactorios, la calibración actúa correctamente eliminando el offset de medida y el motor se ajusta a la referencia recibida con suficiente precisión.

9.2.3. *Sensores (IMU)*

9.2.3.1. *Who Am I*

La primera prueba que se realiza con los sensores era de comprobar que la comunicación era correcta, y para esto se preguntaba cuál era su dirección.

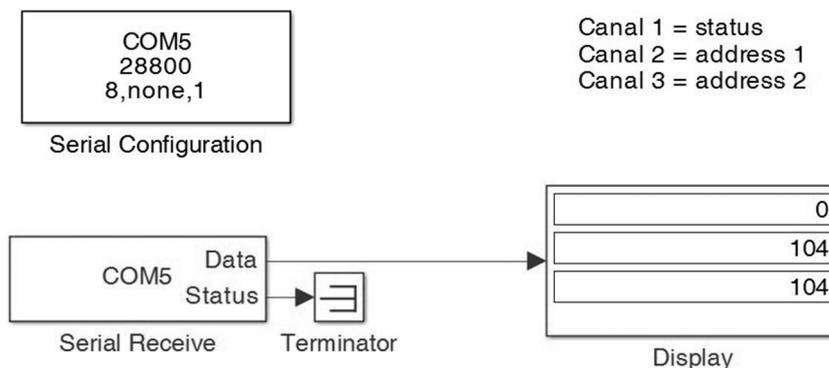


Figura 80: Lectura de la prueba de "Who Am I" de las IMU

En los resultados de la prueba se comprueba que la comunicación es correcta, status = 0. Además la dirección de los sensores también es la que corresponde, $104_{dec} = 0x68_{bin}$.

En el apartado [8.3.3](#), se menciona que uno de los dos sensores debería tener un ID 0x68 y otro 0x69. Esto solo sirve para indicar la diferenciarse entre sí cuando uno tiene conectado el pin AD0 a tierra y el otro a Vcc y se comunican mediante el protocolo I2C. Su dirección real para diferenciarse con otro tipo de sensores es 0x68, como se muestra en la figura.

9.2.3.2. Lectura de los acelerómetros

Lo siguiente a analizar son los resultados de la estimación del ángulo utilizando los acelerómetros de la IMU. Observando las figuras de la 81 a la 83, se puede ver que el ángulo es negativo cuando el péndulo está tumbado hacia la derecha y positivo cuando está hacia la izquierda. Realmente es al revés, pero el único lugar donde se podía enganchar el transportador de ángulos era en la parte posterior, por lo tanto cuando se hable de ángulo negativo se dirá hacia la izquierda y positivo hacia la derecha.

En las figuras a continuación se realiza la comparación de la lectura real del ángulo del péndulo (línea amarilla) en comparación con las lecturas de la comunicación serie. Hay que tener en cuenta que estas lecturas tienen un poco de ruido y simplemente se han cogido en un momento puntual, por lo tanto hay que darle un margen al ángulo real de $\pm 0.5^\circ$.

JULIO 2016



Figura 81: Lectura de ángulo 1 (θ_b)

En la figura 81 el ángulo de la diagonal del péndulo que pasa por el centro es de 49° . Solamente se desvía de la lectura por 0.54° . A pesar de que se le da un rango de error a la medida de 0.5 grados, que sobrepase este por 0.04 no es muy significativo así que se da por correcta.



Figura 82: Lectura de ángulo 2 (θ_b)

En la figura 82, se busca medir el ángulo 0. Es sobre este punto sobre el que va a tener que equilibrarse el sistema así que es importante que esta medida no de mucho error. Como se puede comprobar, -0.3° está dentro del rango permitido por lo que se da por correcta también.

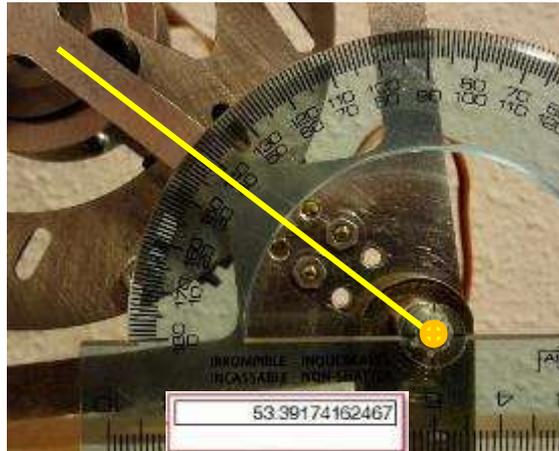


Figura 83: Lectura de ángulo 3 (θ_b)

En la figura 83, el transportador de ángulos marca un ángulo de 51° , mientras que la lectura de los acelerómetros indica 53° . En este caso ya sí que se puede denominar como fuera de rango.

Analizando las medidas de ángulos grandes, en las cuales las dos se han desviado respecto del valor real del ángulo, podría decirse que hay que mejorar la lectura. Sin embargo, la lectura del ángulo 0 sí que es correcta, está dentro del rango. Además, en términos relativos, el mayor error que se obtiene, es el de la lectura número 3 (figura 64) es solamente del 4,5%. Este porcentaje de error es muy pequeño, y podría ser debido al ruido que contiene la señal. Para comprobar esto se recurre al análisis del gráfico de la serie temporal que se muestra en la figura a continuación.

JULIO 2016

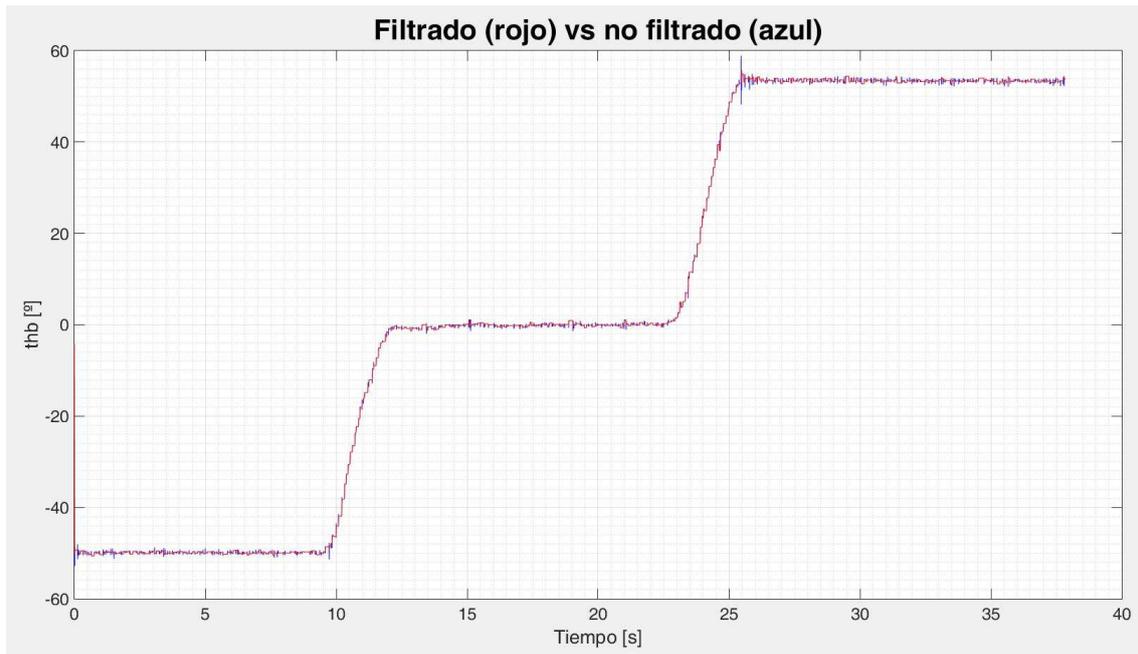


Figura 84: Gráfico de las lecturas de los ángulos filtrando el ruido de la UART

Para las lecturas del ángulo, también hay que realizar un procesado previo para eliminar valores anómalos. En este caso, cualquier valor cuyo valor absoluto sea 10 unidades mayor o menor que el valor anterior, se elimina, y se utiliza un filtro de medias móviles con los 12 valores previos.

En la figura 84 se muestran la curva sin filtrar, pero con los valores anómalos ya eliminados, en color azul, y en rojo la curva filtrada. Con la curva ya filtrada. Observando a partir del segundo 25, donde debería indicar 51°, sí que parece estabilizarse en 53°. Por lo tanto parece que sí que existe un pequeño error de medida, del 3.7%. Este valor es muy pequeño, pero a valores grandes de ángulo sí que provoca un cambio significativo en el valor absoluto del ángulo. Sin embargo, en el rango de balanceo y equilibrio del péndulo, $\pm 5^\circ$, no es lo suficientemente relevante, y por lo tanto se dan las medidas por correctas y se confirma el funcionamiento de los acelerómetros.

9.2.3.3. Lectura de los giróscopos

Para procesar los datos del giróscopo se sigue el mismo método que para el motor y los acelerómetros. Para eliminar los valores anómalos, en este caso si la diferencia entre dos valores contiguos es mayor que 5, se elimina, y el filtro de medias móviles igual que en los acelerómetros, se calcula el valor medio con 12 datos.

Una vez se ha realizado el procesamiento, se entra en el análisis. Lo primero que hay que comprobar es que los dos giróscopos midan aproximadamente lo mismo. Para ello se superponen las series temporales de cada uno y se analiza la gráfica. Esto se muestra en la figura 85. Se puede comprobar que sí que miden lo mismo ya que debería haber una lectura en rojo y la otra en azul, y lo que se ve es una gráfica de color morado, indicando que ambos giróscopos están funcionando correctamente y leyendo prácticamente el mismo valor.

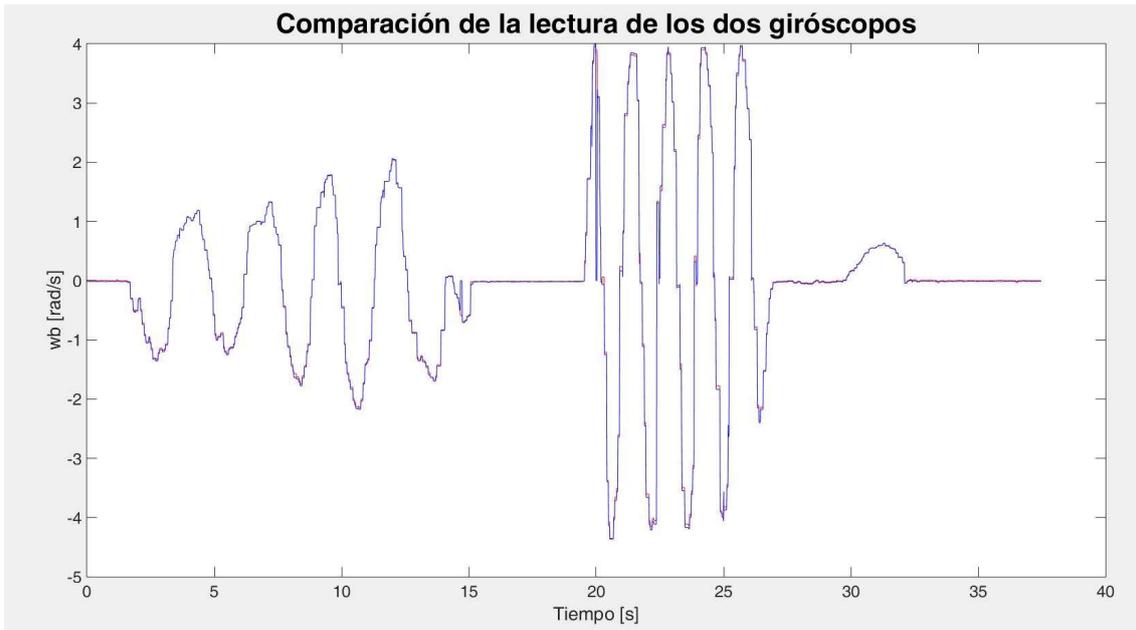


Figura 85: Superposición de las lecturas de los dos giróscopos

Después de comprobar que ambos giróscopos miden prácticamente lo mismo, para reducir el error al máximo, la medida de la velocidad angular del péndulo ω_b se calcula haciendo la media de las lecturas los dos.

JULIO 2016

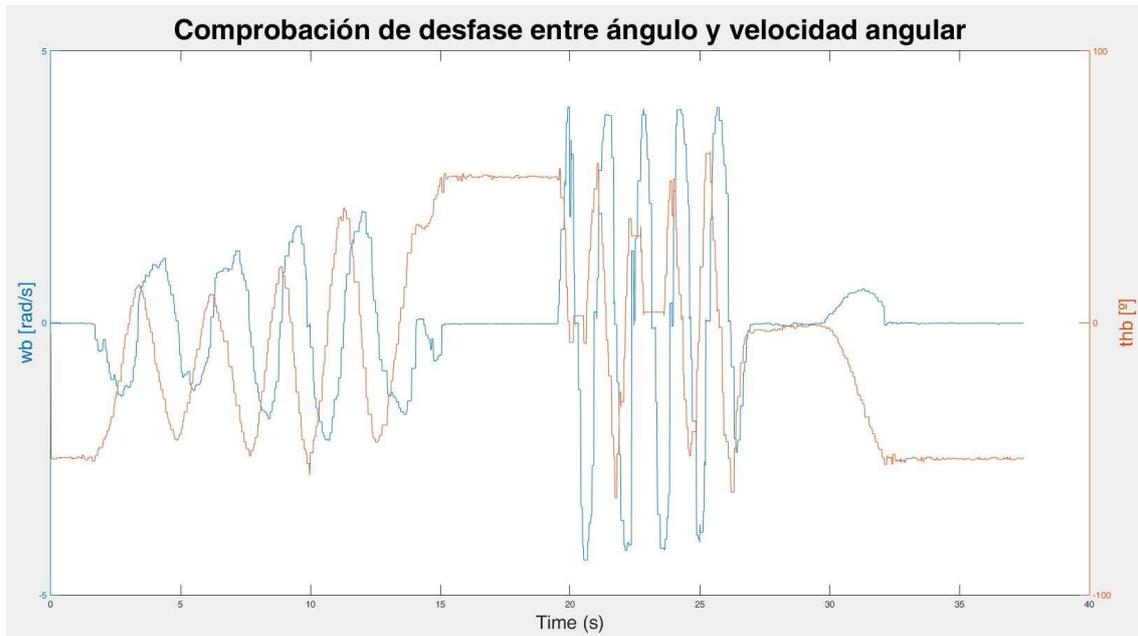


Figura 86: Comprobación del desfase entre las lecturas del ángulo y la velocidad angular

Por último, hay que asegurarse que la combinación de las medidas de los acelerómetros y los giróscopos son coherentes, deberían estar desfasadas entre sí. Cuando la $\omega_b = 0$ rad/s, el ángulo debería o ser constante o estar cambiando de dirección. En la figura 86 se puede comprobar que esto es así, cuando hay variaciones en la velocidad angular también las hay en el ángulo, y cuando el disco no se mueve el ángulo se mantiene constante.

9.2.4. Máquina de estados

Los resultados de la prueba de la máquina de estados son muy importantes, ya que si no se programa correctamente, el sistema no se coordina bien y no se consiguen los objetivos.

Lo primero es probar que las transiciones entre estados fluyen correctamente, para ello se quita el brazo del servo, la transición del estado de aceleración al de salto se reducen las revoluciones a 200 rpm y el tiempo de transición del estado dos (salto) al tres (balanceo) se aumenta a 4 segundos. De esta manera hay tiempo suficiente para comprobar que el LED de cada estado se enciende en el momento adecuado.

Como era de esperar, al cargar el programa y encender la placa, se enciende el LED del estado 0. Al pulsar el pulsador, después de los 20 segundos de calibración, se enciende el LED del estado 1 y comienza a acelerar el disco. Una vez este alcanza la velocidad configurada, se enciende el LED del estado 2, y al cabo de unos 4 segundos, la del estado 3.

Lo próximo es poner el brazo del servo en su sitio y comenzar a aumentar la ω_w para que realice un salto lo suficientemente grande. A partir de 1500 rpm, cuando comienza a saltar lo suficiente, el brazo del servo se dobla y se rompe. Al cambiar el brazo por otro ocurre lo mismo. No sirven los brazos del servomotor de plástico puesto que se doblan.

Para evitar la rotura de los brazos, se refuerzan con acero rodeando el brazo de plástico. Parece que sí que aguanta las 1500 rpm, pero a partir de 1800 rpm, lo que se rompe es el tornillo que sujeta el brazo en su sitio.



Figura 87: Brazos de servomotor inservibles

En conclusión, la máquina de estados ha funcionado correctamente, pero el sistema de frenado utilizado no es capaz de resistir el golpe, por lo tanto no se puede poner en conjunto el sistema y hay que centrarse en mejorar el control por separado.

9.3. Resultados del control sobre el sistema real

El control con el que se obtuvieron los mejores resultados en la simulación fue el control número 6, con matriz de ganancias:

$$Klqr6 = (-48.11 \quad -7.90 \quad -0.0017)$$

La prueba sobre el sistema real se realiza con este control. La hipótesis planteada es que al equilibrar un ángulo tan pequeño, el control no sería efectivo. El resultado al cargar el control en el sistema real es el siguiente:

JULIO 2016

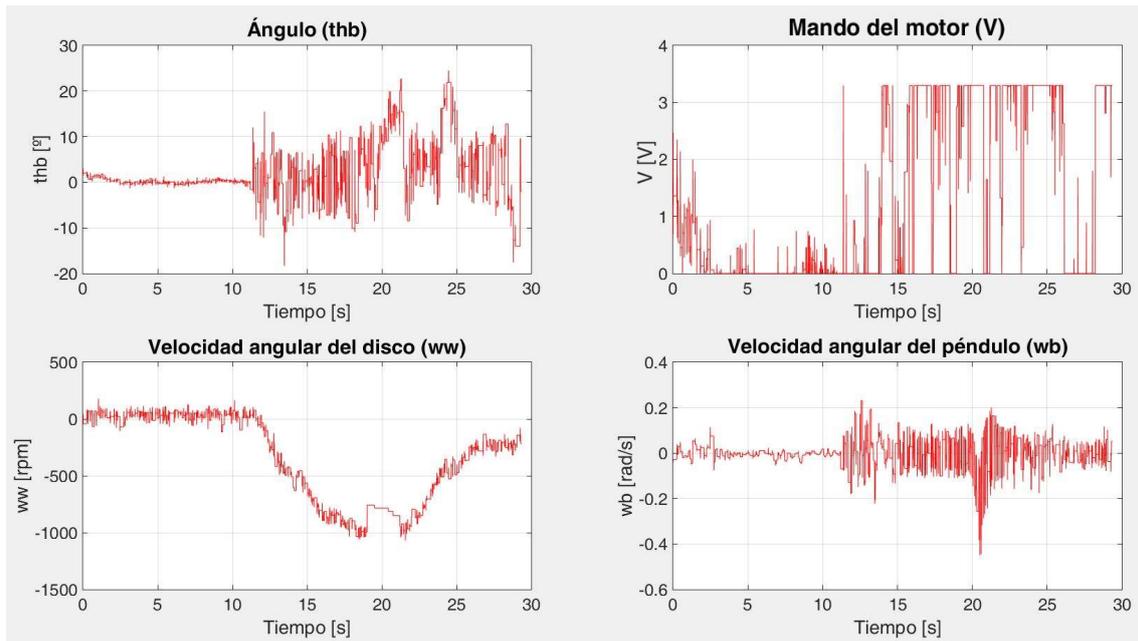


Figura 88: Resultados del control en el sistema real

A partir de $t=10$ s el control comienza a actuar. Se observa que la cantidad de ruido medido en la señal del ángulo, llevando al mando a saturación. El ángulo fijado para la comprobación de las medidas se mantiene entre -20° y 20° y se consigue forzándolo manualmente para que el péndulo no caiga.

El ruido se debe a la vibración del volante de inercia cuando gira, que afecta a la IMU que estima el ángulo del péndulo a partir de los acelerómetros. La mejor solución a tomar sería poder eliminar esas vibraciones, pero se opta por la implantación de un limitador de gradiente a la medida actuando de filtro. De esta manera se reduce el ruido, obteniendo los siguientes resultados:

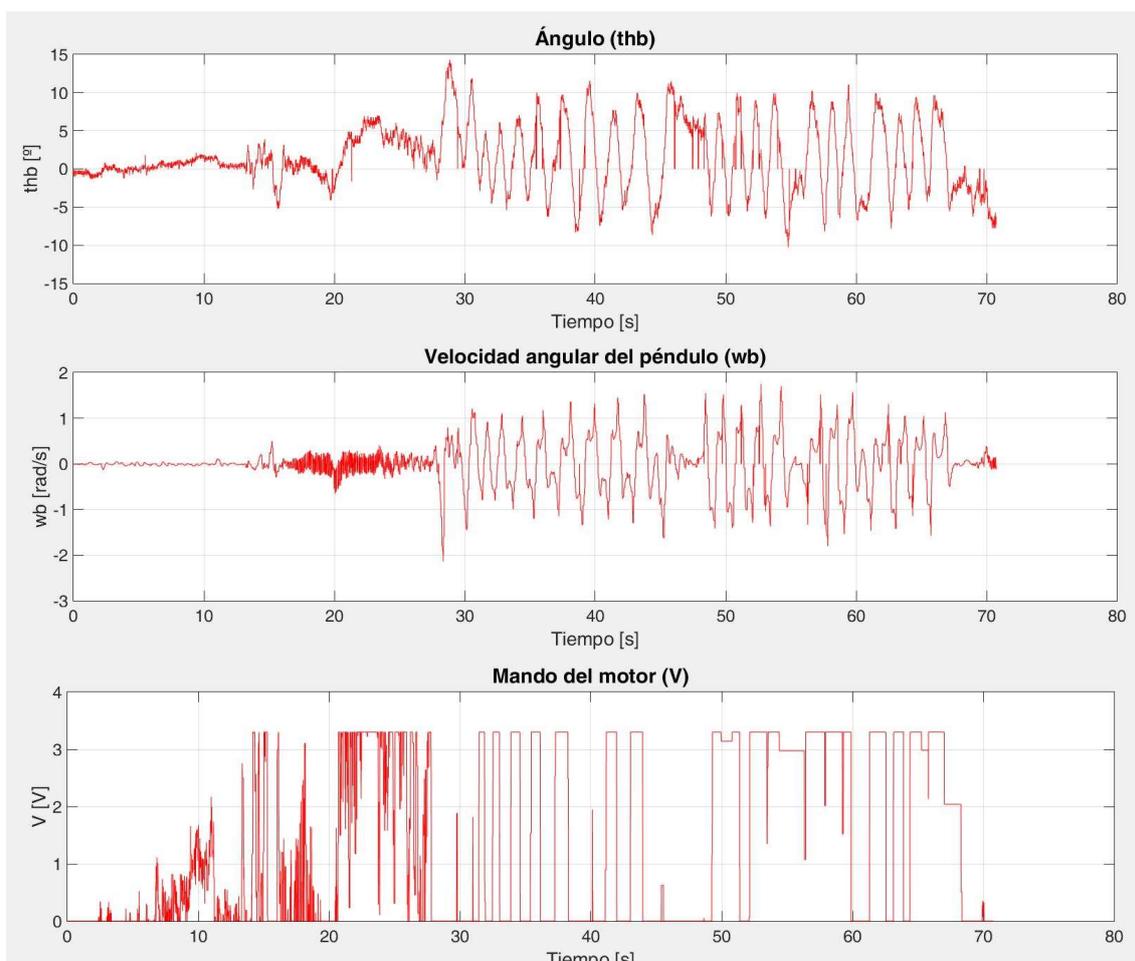


Figura 89: Resultados del control en el sistema real filtrando la lectura del ángulo antes del control

De nuevo se activa el control aproximadamente a partir de $t=10$ s. Hasta $t=25$ s se sujeta y se mueve poco a poco para estudiar la reacción del control. Al estar siendo sujeto suavemente, se aprecia que la velocidad angular del péndulo sufre muchas vibraciones, y no es capaz estabilizarse.

A partir del $t=25$ s, se fijan dos puntos de apoyo permitiendo un máximo ángulo de 10° . Al fijar los límites, cuando llega a ese punto, el control es ayudado y rebota hacia el otro lado, como se puede comprobar durante el resto del ensayo. Un factor a mejorar sería la fuerza de reacción para poder controlar de manera más efectiva el equilibrio del péndulo. Con esta mejora el control sí sería válido.

Al no tener suficiente fuerza, como se había previsto, el control trata de estabilizar el sistema enviando la máxima corriente posible saturando el motor, y aun así no es capaz equilibrarse sobre ángulo 0.

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 10: Conclusiones

Los objetivos del proyecto son diseñar y construir un péndulo invertido, controlado por un volante de inercia, que sea capaz de estabilizarlo sobre su esquina, y hacerlo saltar desde una posición de reposo hasta alcanzar la misma. Además, construir un modelo digital donde se pudieran realizar diferentes ensayos y pruebas para ser capaz de mejorar el modelo.

Se ha diseñado un prototipo de péndulo invertido controlado por volante de inercia partiendo de cero. En cuando al diseño del péndulo ha sido correcto. Una posible mejora del sistema es el anclaje al eje, minimizando las vibraciones que afectan al control sobre el sistema real para que el control LQR sea capaz de actuar de una manera más adecuada.

El diseño del sistema de frenado no ha sido el óptimo, los brazos de plástico no soportan la fuerza del impacto necesario para el frenado del volante de inercia y no son capaces de transmitir la energía de frenado necesaria para el salto. Se ha estudiado la implementación de un brazo con funda metálica pero en este caso, el fallo lo presentaba el sistema de tornillos, no siendo capaz de soportar la fuerza y desmontando el tornillo. Una alternativa interesante al anclaje de este tipo de brazo es la instalación de paredes, ancladas a la pieza que mantiene el servo en su posición, que soporten al brazo del servo, redistribuyendo la energía del golpe en el conjunto de frenado, no reduciendo el esfuerzo al atornillado.

La simulación sobre el sistema real ha tenido una serie de complicaciones. El control obtenido mediante la simulación equilibra el péndulo para un rango de ángulo muy pequeño. Sin embargo, al implantarlo en el sistema real, hay dos factores a tener en cuenta que empeoran a la efectividad del control, el ruido de medida y la potencia del motor.

El ruido está producido por el volante de inercia y las vibraciones que producen las altas velocidades de giro, que afectan a la lectura de las variables de estado, en especial al ángulo. El ángulo es la variable de estado más importante, y se pretende estabilizar a partir del control del resto de variables. Esta medida no es real debido a la sensibilidad de los acelerómetros ante la vibración, implementando un sistema de amortiguación estas lecturas esto se corregiría.

El par de fuerzas producidas por el motor proporcionan la inercia y la velocidad al volante para el equilibrio del sistema. Se ha comprobado en las simulaciones que el motor escogido no es lo suficientemente potente para proporcionar el par necesario, significando esto que el sistema no acelera lo suficientemente rápido para contrarrestar la energía potencial y estabilizarse. Utilizando un motor más potente, sería capaz de conseguir la aceleración necesaria y estabilizarse.

JULIO 2016

Para ser capaz de controlar todo el sistema en su conjunto y recibir unas lecturas de la manera más eficiente posible y conteniendo el mínimo error posible, se ha diseñado toda la arquitectura de control, llegando a tener que inicializar registros de la IMU uno a uno. Además se han implementado sistemas de calibración para que los diferentes módulos del sistema no contengan offsets en la medida. Todo esto se ha implementado en una máquina de estados, que consigue que todas las fases de los controles de salto y balanceo actúen correctamente de no ser por los pequeños problemas mecánicos encontrados.

Todas las mejoras mencionadas, consiguen que el sistema sea más efectivo. No se han llevado a cabo debido al alcance del proyecto y a los recursos disponibles. Sin embargo, el simulador diseñado es de gran precisión y muestra que el objetivo del proyecto es alcanzable modificando los parámetros del sistema comentados.

Bibliografía

- [1] Castro Salguero R., *Sistemas Mecánicos Subactuados*, Universidad Nacional de Ingeniería, Facultad de Ingeniería Eléctrica y Electrónica, Marzo 2001.
- [2] Hernández-Guzmán V. M., Silva-Ortigoza R. y Carrillo-Serrano R. V., *Control Automático: Teoría de Diseño, Construcción de Prototipos, Modelado, Identificación y Pruebas Experimentales*, Colección, CIDETEC–IPN, México, DF, México, 2013.
- [3] Gajamohan M., Merz M., Thommen I., and D’Andrea R., *The Cubli: A Cube that can Jump Up and Balance*, in Proc. IEEE/RSJ International Conference of Intelligent Robots and Systems (Algarve, Portugal), pp. 3722-3727, October 2012.
- [4] Muehlebach M., Mohanarajah G., and D’Andrea R., *Nonlinear Analysis and Control of a Reaction Wheel-based 3D Inverted Pendulum*, in Proc. Conference on Decision and Control, CDC 2013 (Florence, Italy)
- [5] Gajamohan M., Muehlebach M., Widmer T., and D’Andrea R., *The Cubli: A Reaction Wheel-based 3D Inverted Pendulum*, in Proc. European Control Conference (Zurich, Switzerland), pp. 268-274, July 2013.
- [6] Acosta J. Á., *Furuta’s Pendulum: A Conservative Nonlinear Model for Theory validation and Practise*, Math. Probl. Eng., vol. 2010, pp. 1-29, 2010.
- [7] Hellerstein J. L., Diao Y., Parekh S., Tibury D. M., *Feedback Control of Computing Systems*, John Wiley & Sons, Chichester (2004)
- [8] Block D. J., Åström K. J., and Spong M. W., *The Reaction Wheel Pendulum*, 1st ed., vol. 1, M. W. Spong, Ed. Morgan & Claypool, 2007, p. 112.
- [9] Loría A., Nijmeijer H., *Passivity-based control*, in Encyclopaedia of Life Support Systems (EOLSS), Vol. Perspectives and Overview of Life Support Systems and Sustainable Development (Prof. Unbehauen, ed.), EOLSS Publishers Ltd., 2002.
- [10] Trimpe S. and D’Andrea R., *Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom*, in IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 2630–2636.
- [11] Alonso D., Paolini E., and Moiola J., *Controlling an inverted pendulum with bounded controls*, in Dynamics, Bifurcations, and Control, ser. Lecture Notes in Control and Information Sciences. Springer Berlin / Heidelberg, 2002, vol. 273, pp. 3–16.
- [12] Stimac A. K., *Standup and Stabilization of the Inverted Pendulum*, Department of Mechanical Engineering, Massachusetts Institute of Technology, June, 1999.

JULIO 2016

- [13] Hardesty L., *Surprisingly simple scheme for self-assembling robots*, October, 4, 2013, <http://news.mit.edu/2013/simple-scheme-for-self-assembling-robots-1004>
- [14] *Segway*, <http://www.segway.es/mirada.asp>
- [15] Bazylev D.N.; et al. (2015), *Approaches for stabilizing of biped robots in a standing position on movable support*, Scientific and Technical Journal of Information Technologies, Mechanics and Optics 15 (3): 418
- [16] *ASIMO*, <http://asimo.honda.com/default.aspx>
- [17] *Mecánica lagrangiana*, Wikipedia, La enciclopedia libre, https://en.wikipedia.org/w/index.php?title=Lagrangian_mechanics&oldid=723209385
- [18] Weisstein, Eric W., *Euler-Lagrange Differential Equation*, MathWorld, <http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html>
- [19] Bjerke, E., Pehrsson, B., *Development of a Nonlinear Mechatronic Cube*, Department of Signal and Systems, Division of Automatic control, Automation and Mechatronics, Chalmers University of Technology, 2016
- [20] Colwell C.H., *Center of mass*, PhysicsLab, http://dev.physicslab.org/DocumentPrint.aspx?doctype=3&filename=RotaryMotion_CenterMass.xml
- [21] Murray R M, *Lecture 2 – LQR Control*, California Institute of Technology, January, 11, 2006
- [22] Muñoz Frías J.D., *Máquinas de estado finitos*, en *Introducción a los sistemas digitales* (pp. 149-175). January 2012
- [23] *Waijung blockset*, <http://waijung.aimagin.com/>
- [24] Sánchez Miralles A., *Módulo PWM*, en *Libro SED* (pp. 314-322).
- [25] *I2C*, <http://i2c.info/>
- [26] Zamora Macho J. L., *Modelado en Espacio de estado*.
Smith W. S., *Moving Average Filters*, en *California Technical Publishing, The Scientist and Engineer's Guide to Digital Signal Processing* (pp 277-284).

Anexo A: Código fuente

Código del diseño del control LQR

```

clc
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FICHERO MAESTRO CONTROL CUBLI %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOMENCLATURA
% Subíndice b = cuerpo cubli (incluido motor, sin contar volante)
% Subíndice w = volante + rotor
% Subíndice m = motor
% Subíndice d = discretizada
% O = punto de anclaje
% O' = eje motor
% G = centro de gravedad

%% PARAMETROS
% Tiempo de muestreo
ts=0.01;

% Gravedad [m/s^2]
g = 9.81;
% Distancia punto anclaje a eje motor (OO') [m]
l = 0.091925;
% Distancia punto anclaje a centro de gravedad cuerpo cubli (OG) [m]
lb = 0.085;

% Masa cuerpo del cubli [kg]
mb = 0.2073;
% Momento inercia cuerpo cubli [kg*m^2]
% Ib = 3.34*10^-3; % Cubli de ETH
Ib = 8.0*10^-3;
% Coeficiente dinámico fricción cuerpo cubli
% Cb = 1.02*10^-3; % Cubli de ETH
Cb = 1*10^-3;

% Masa volante de inercia + rotor motor [kg]
mw = 0.20775;
% Momento inercia volante + rotor [kg*m^2]
% Iw = 0.57*10^-3; % Cubli de ETH
Iw = 0.24*10^-3;
% Coeficiente dinámico fricción volante + rotor
% Cw = 0.05*10^-3; % Cubli de ETH
Cw = 0.013*10^-3;

% Constante de par de motor DC [N*m/A]
% Km = 33.5*10^-3; % Cubli de ETH
% Km = 2*14.67*10^-3; % Doble de par
Km = 14.67*10^-3;

```

JULIO 2016

```

% Vector de parametros
Parametros = [g l lb mb Ib Cb mw Iw Cw Km];

% Stall Current (A)
% Ki=23.3; % Cubli de ETH
% Ki=5.24; % Esto es la que aguanta solo un par de segundos
Ki=1.36;

%% Condiciones iniciales
thb0 = 2*3.14; %[°]

% x=[thb wb ww]
x0 = [thb0*pi/180 0 0];
u0 = 0;

%% Punto de operacion
Xpo = [0 0 0];

Upo = 0; % como pongo que varíe

%% Modelo lineal
[matA, matB, matC, matD] = linmod('Planta_Cubli_i',Xpo, Upo);

% Ponemos la planta en diferentes formatos
Pss = ss(matA,matB,matC,matD);
Ptf = tf(Pss);
Pzpk = minreal(zpk(Pss));

% Controlabilidad
co=ctrb(Pss);
controllability = rank(co);

if (controllability ~= rank(matA))
    fprintf('ERROR!! La matriz no es controlable');
end

% Discretización de la planta
Pssd = c2d(Pss, ts, 'zoh');

matAd = Pssd.a;
matBd = Pssd.b;
matCd = Pssd.c;
matDd = Pssd.d;

%% LQR control
Q=[1000 0 0; 0 1000 0 ; 0 0 1];
R=1000;

[klqr,s,e]=dlqr(matAd,matBd,Q,R);
klqr
    
```


JULIO 2016

```

subplot(1,2,1)
plot(tiempo_pendolo,grados_pendolo,'b')
hold on
plot(tiempo_pendolo,grados_filter,'r')
T = title('Real(azul) vs Filtrado(rojo)');
Y = ylabel('thb [°]');
X = xlabel('Tiempo [s]');
set(T,'FontSize',22)
set(Y,'FontSize',20)
set(X,'FontSize',20)
set(gca,'FontSize',18);

figure(3)
plot(tiempo_pendolo, rads_filter,'g')
hold on
plot(tiempo_pendolo, grados_filter*pi/180,'r')
T = title('Grados (red) vs Radianes (green)');
Y = ylabel('thb');
X = xlabel('Tiempo [s]');
set(T,'FontSize',22)
set(Y,'FontSize',20)
set(X,'FontSize',20)
set(gca,'FontSize',18);
% the reading in degrees and then transformed to radians is better (has
% less noise)

% we manually pick the sections where the pendulo is left free (7areas)
figure(4)
plot(tiempo_pendolo,grados_filter*pi/180,'r')
hold on
plot(tiempo_pendolo,omega_filter,'g')
T = title('thb vs wb (Filtro medias móviles)');
X = xlabel('Tiempo [s]');
set(T,'FontSize',22)
set(X,'FontSize',20)
set(gca,'FontSize',18);
% we manually pick the sections where the pendulo is left free (7areas)
%those areas are:
wb_thb_1(:,1)=(0:0.01:33.64);
wb_thb_1(:,2)=wb_1(:,2);
wb_thb_1(:,3)=thb_1(:,2);
wb_thb{1}=wb_thb_1;

wb_thb_2(:,1)=(0:0.01:41.87);
wb_thb_2(:,2)=wb_2(:,2);
wb_thb_2(:,3)=thb_2(:,2);
wb_thb{2}=wb_thb_2;

wb_thb_3(:,1)=(0:0.01:49.09);
wb_thb_3(:,2)=wb_3(:,2);
wb_thb_3(:,3)=thb_3(:,2);
wb_thb{3}=wb_thb_3;

wb_thb_4(:,1)=(0:0.01:49.08);
wb_thb_4(:,2)=wb_4(:,2);
wb_thb_4(:,3)=thb_4(:,2);
wb_thb{4}=wb_thb_4;

wb_thb_5(:,1)=(0:0.01:54.58);
wb_thb_5(:,2)=wb_5(:,2);
wb_thb_5(:,3)=thb_5(:,2);
wb_thb{5}=wb_thb_5;

wb_thb_6(:,1)=(0:0.01:46.68);
wb_thb_6(:,2)=wb_6(:,2);
wb_thb_6(:,3)=thb_6(:,2);
wb_thb{6}=wb_thb_6;

wb_thb_7(:,1)=(0:0.01:89.07);
wb_thb_7(:,2)=wb_7(:,2);
wb_thb_7(:,3)=thb_7(:,2);
wb_thb{7}=wb_thb_7;

```

```

%% Second order filter to get acceleration
% Creation of second order filter to get acceleration from angular velocity
s=tf('s');
wn=0.6283; % extracted from plots T=10->wn=2*pi/T
% time in tenths of second (x axis)
seta=0.7;

% filter initialization
% filter_2nd_der=s^2/(s^2/wn^2+2*seta*s/wn+1);
% filter_2nd=s/(s^2/wn^2+2*seta*s/wn+1);
filter_2nd_der=s^2/(s^2/wn^2+2*seta*s/wn+1)/(1+s/wn);
filter_2nd=s/(s^2/wn^2+2*seta*s/wn+1)/(1+s/wn);

% Acceleration (time,wb,thb,a_filter,wb_filter,thb_filter)
for ind=1:length(wb_thb)

wb_thb{ind}(:,4)=lsim(filter_2nd_der,wb_thb{ind}(:,2),wb_thb{ind}(:,1));
wb_thb{ind}(:,5)=lsim(filter_2nd,wb_thb{ind}(:,2),wb_thb{ind}(:,1));
wb_thb{ind}(:,6)=lsim(filter_2nd,wb_thb{ind}(:,3),wb_thb{ind}(:,1));
end

% a=filter(b,a,wb_thb{7}(:,4));
% omega=lsim(filter_2nd,wb,tiempo_pendulo);
% angulo=lsim(filter_2nd,thb,tiempo_pendulo);
figure(5)
plot(wb_thb{7}(:,1),wb_thb{7}(:,6),'g') % thb_filter
hold on
plot(wb_thb{7}(:,1),wb_thb{7}(:,5),'r') % wb_filter
hold on
plot(wb_thb{7}(:,1),wb_thb{7}(:,4),'b') % a_filter
T = title('thb (verde) vs wb (rojo) vs alfab (azul) filtro tercer orden');
Y = ylabel('[rad] vs [rad/s] vs [rad/s^2]');
X = xlabel('Tiempo [s]');
set(T,'FontSize',22)
set(Y,'FontSize',20)
set(X,'FontSize',20)
set(gca,'FontSize',18);

figure(6)
plot(wb_thb{7}(:,1),wb_thb{7}(:,6),'g') % thb_filter
hold on
plot(wb_thb{7}(:,1),wb_thb{7}(:,3),'r') % thb
T = title('Filtrado tercer orden (verde) vs Filtrado medias móviles (rojo)');
Y = ylabel('thb [rad]');
X = xlabel('Tiempo [s]');
set(T,'FontSize',22)
set(Y,'FontSize',20)
set(X,'FontSize',20)
set(gca,'FontSize',18);

figure(7)
plot(wb_thb{7}(:,1),wb_thb{7}(:,5),'g') % wb_filter
hold on
plot(wb_thb{7}(:,1),wb_thb{7}(:,2),'r') % wb
T = title('Filtrado tercer orden (verde) vs Filtrado medias móviles (rojo)');
Y = ylabel('wb [rad/s]');
X = xlabel('Tiempo [s]');
set(T,'FontSize',22)
set(Y,'FontSize',20)
set(X,'FontSize',20)
set(gca,'FontSize',18);

%% Extraction of constants of equation
% parameters
mb=0.2073;
lb=0.085;
mw=0.20775;
l=0.091925;
Iw=0.24*10^-3;

```

JULIO 2016

```

% calculo de constantes
% Y=(mb*lb+mw*l)*9.81*sin(wb_thb{7}(:,6));
% A=[wb_thb{7}(:,4) wb_thb{7}(:,5)];
% theta=A\Y;
%
% Cb=-theta(2);
% Ib=-theta(1)-Iw-mw*l^2;

theta={};
for ind=1:length(wb_thb)
    Y=(mb*lb+mw*l)*9.81*sin(wb_thb{ind}(:,6));
    A=[wb_thb{ind}(:,4) wb_thb{ind}(:,5)];
    theta{ind}=A\Y;
%     theta{ind}(:, :)
end

theta1(1,1:2)=theta{1};
theta1(2,1:2)=theta{2};
theta1(3,1:2)=theta{3};
theta1(4,1:2)=theta{4};
theta1(5,1:2)=theta{5};
theta1(6,1:2)=theta{6};
theta1(7,1:2)=theta{7};

Cb=mean(theta1(:,2))
Ib=(-mean(theta1(:,1))-Iw-mw*l^2)
    
```

Ejemplo de tratamiento de datos

```

%% Tratamiento resultados pruebas
%% Carga de datos
load('angle_estimation_2.mat')

%% angle estimation data
% eliminar valores anómalos
for (i=1:angle_estimation.Length)
    angle(i,1)=angle_estimation.Data(i);
    if i>1 && abs(angle(i,1)-angle(i-1,1))>20
        angle(i,1)=angle(i-1,1);
    end
end

% Time
t_angle = angle_estimation.Time/10;

%% Creation of moving average filter
% Initialization of filter parameters (average with last 12 values)
a=1;
b=ones(1,12)/12;

% Filter data
% angle_filter = filter(b1,a,angle);
angle_filter = filter(b,a,angle);

%% Plot timeseries data
figure(1)
plot(t_angle,angle,'b')
hold on
plot(t_angle,angle_filter,'r')
grid
grid minor
Title = title('Filtrado (rojo) vs no filtrado (azul)');
Y = ylabel('thb [°]');
X = xlabel('Tiempo [s]');
set(Title,'FontSize',30)
set(Y,'FontSize',20)
set(X,'FontSize',20)
set(gca,'FontSize',18);

```

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Anexo B: Datasheets

En esta sección se adjuntan los enlaces a las hojas de datos de los componentes electrónicos utilizados para la realización de este proyecto:

1. STM32F407
<http://udel.edu/~furkan/cpeg423/13summer/stm32f4/DM00037051.pdf>
2. Motor EC 45 flat 12 W
http://www.maxonmotor.es/medias/sys_master/root/8821067612190/16-298-EN.pdf
3. ESCON Module 24/2
http://www.maxonmotor.es/medias/sys_master/root/8818449842206/466023-ESCON-Module-24-2-Referencia-del-Dispositivo-Es.pdf
4. Servomotor MG90S
http://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf
5. IMU MPU6050 Datasheet
<https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
6. IMU MPU6050 Register Map
https://www.olimex.com/Products/Modules/Sensors/MOD-MPU6050/resources/RM-MPU-60xxA_rev_4.pdf

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Parte 2: Planos

UNIVERSIDAD PONTIFICIA COMILLAS

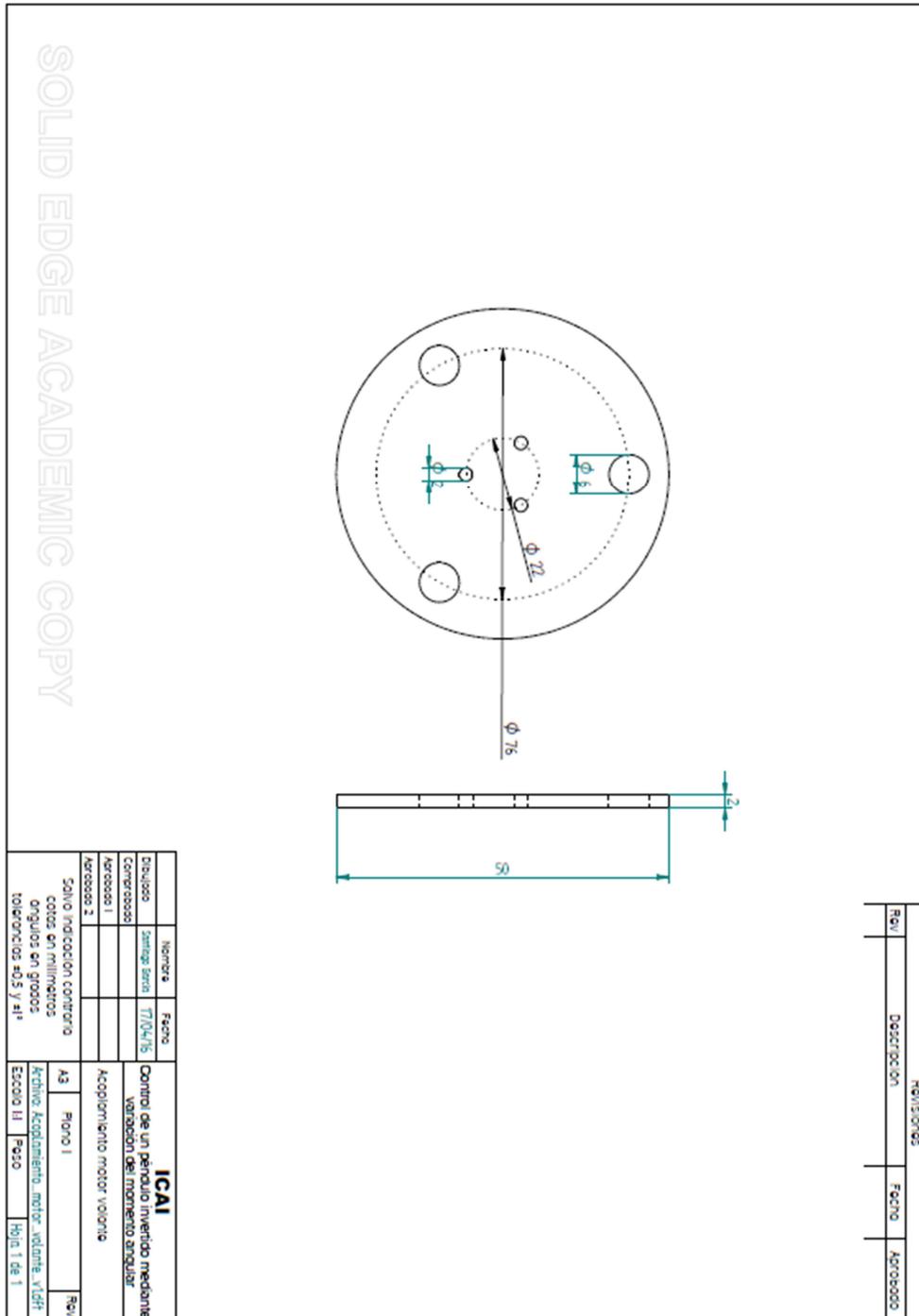
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

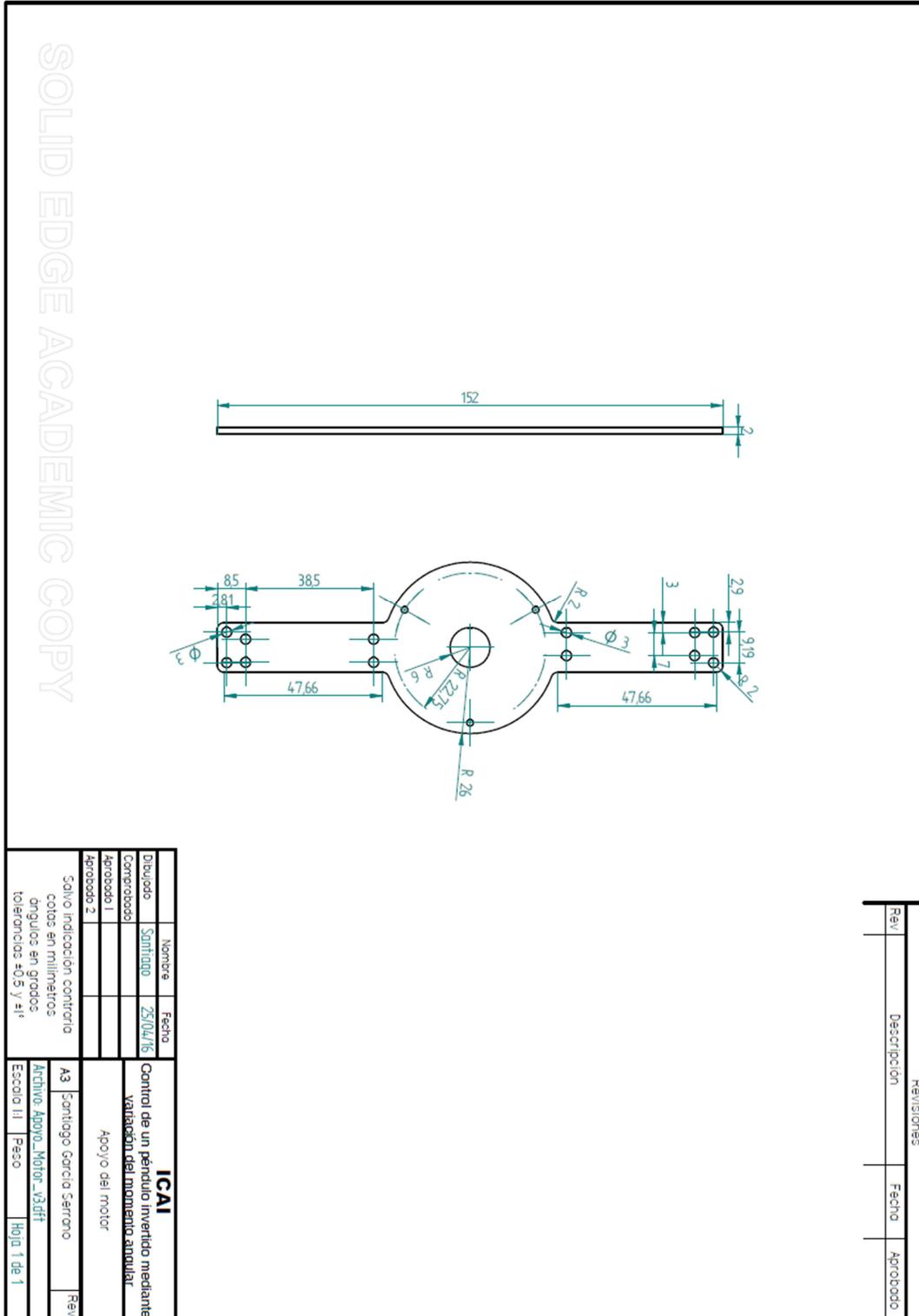
Capítulo 1: Planos de piezas

1.1. Acoplamiento del motor al volante de inercia

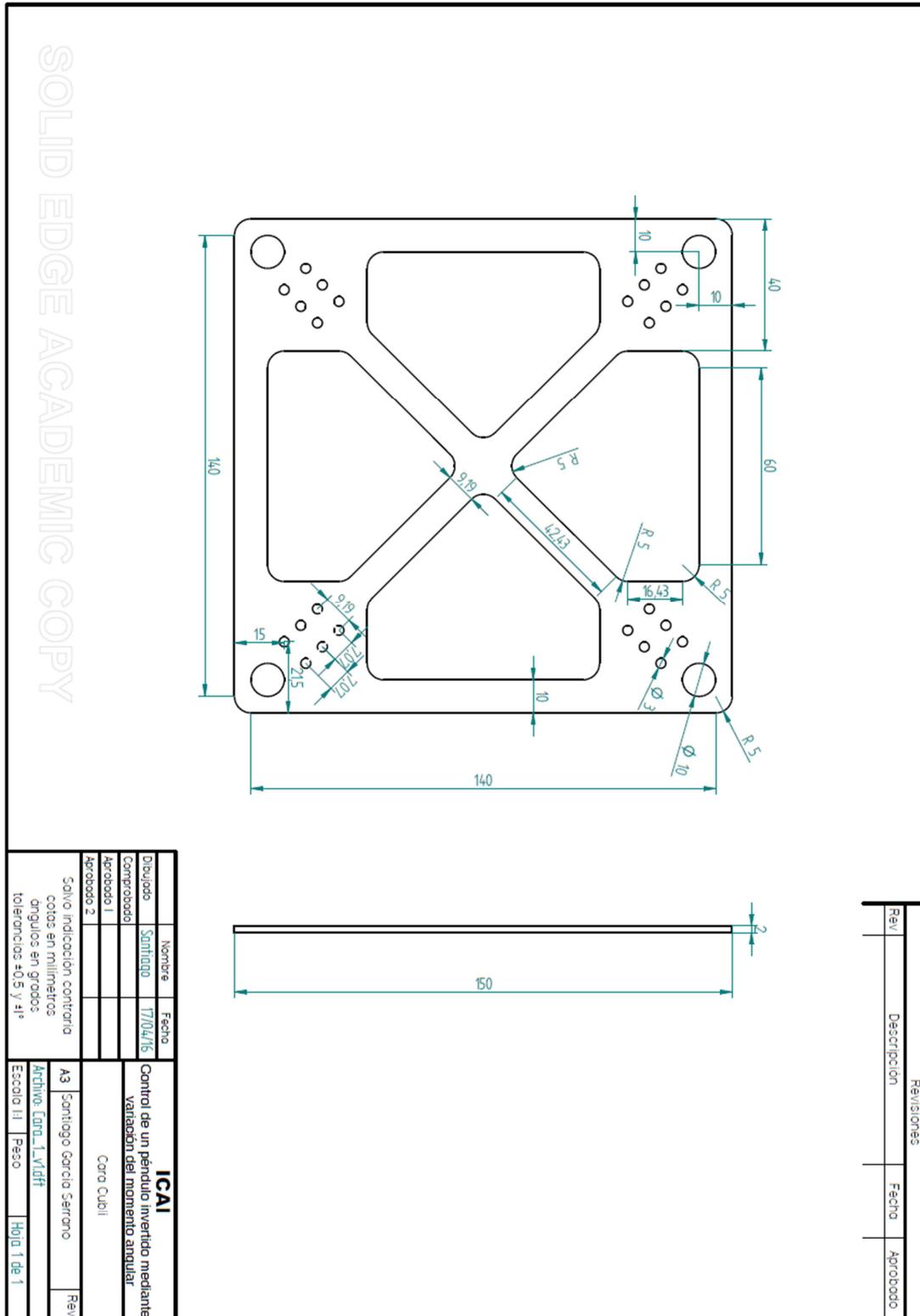


JULIO 2016

1.2. Apoyo del motor

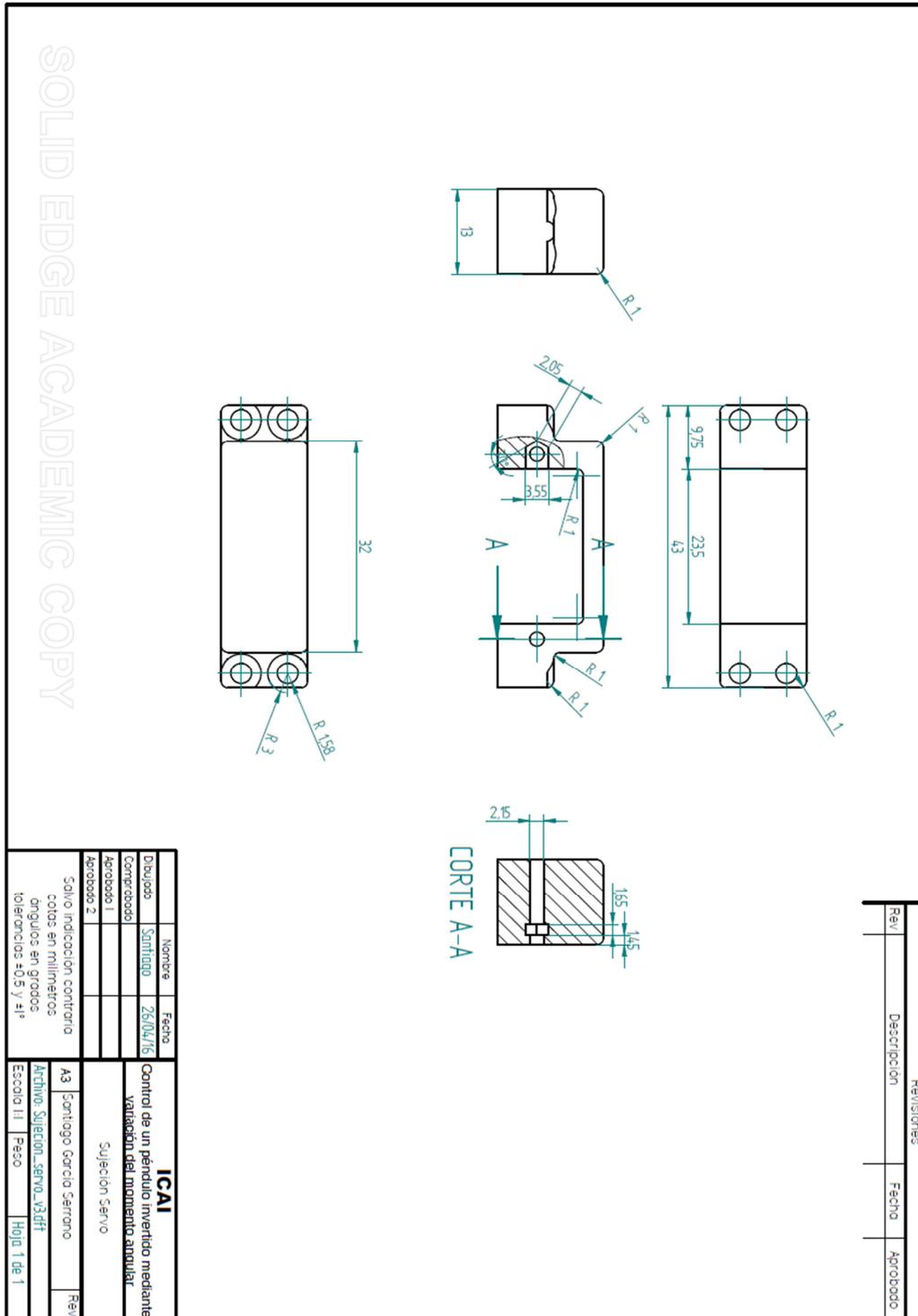


1.3. Cara del cubo

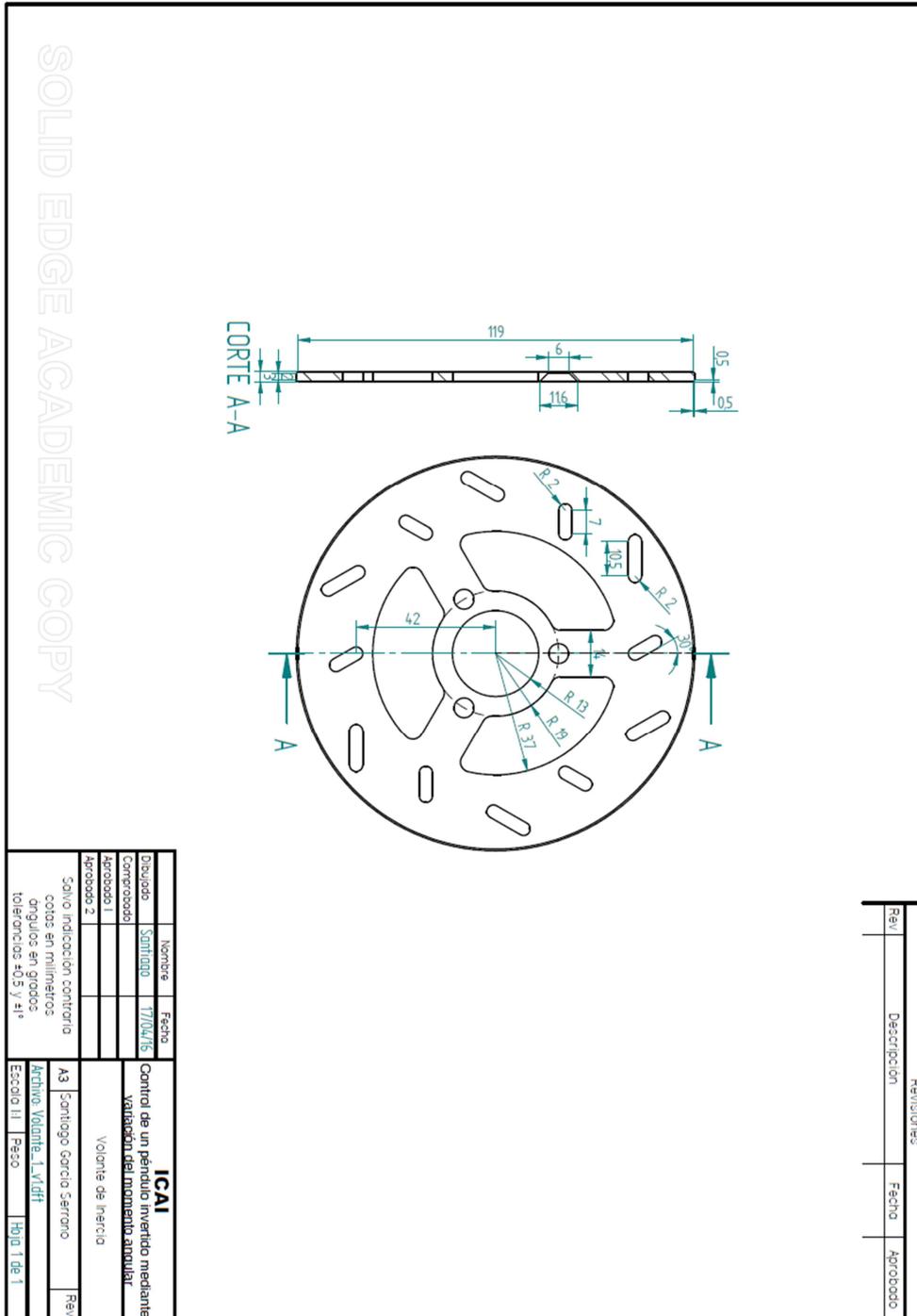


JULIO 2016

1.4. Sujeción del servomotor



1.5. Volante de inercia



Revisiones		
Rev	Descripción	Aprobado

Nombre	Fecha	ICAI
Santiago	17/04/16	
Control de un péndulo invertido mediante variación del momento angular		Volante de inercia
Aprobado I		
Aprobado 2		Rev
Salvo indicación contraria cotos en milímetros ángulos en grados tolerancias +0.5 y ±1°		
Archivo	Escuela II	Hoja 1 de 1
A3	Santiago García Serrano	

UNIVERSIDAD PONTIFICIA COMILLAS

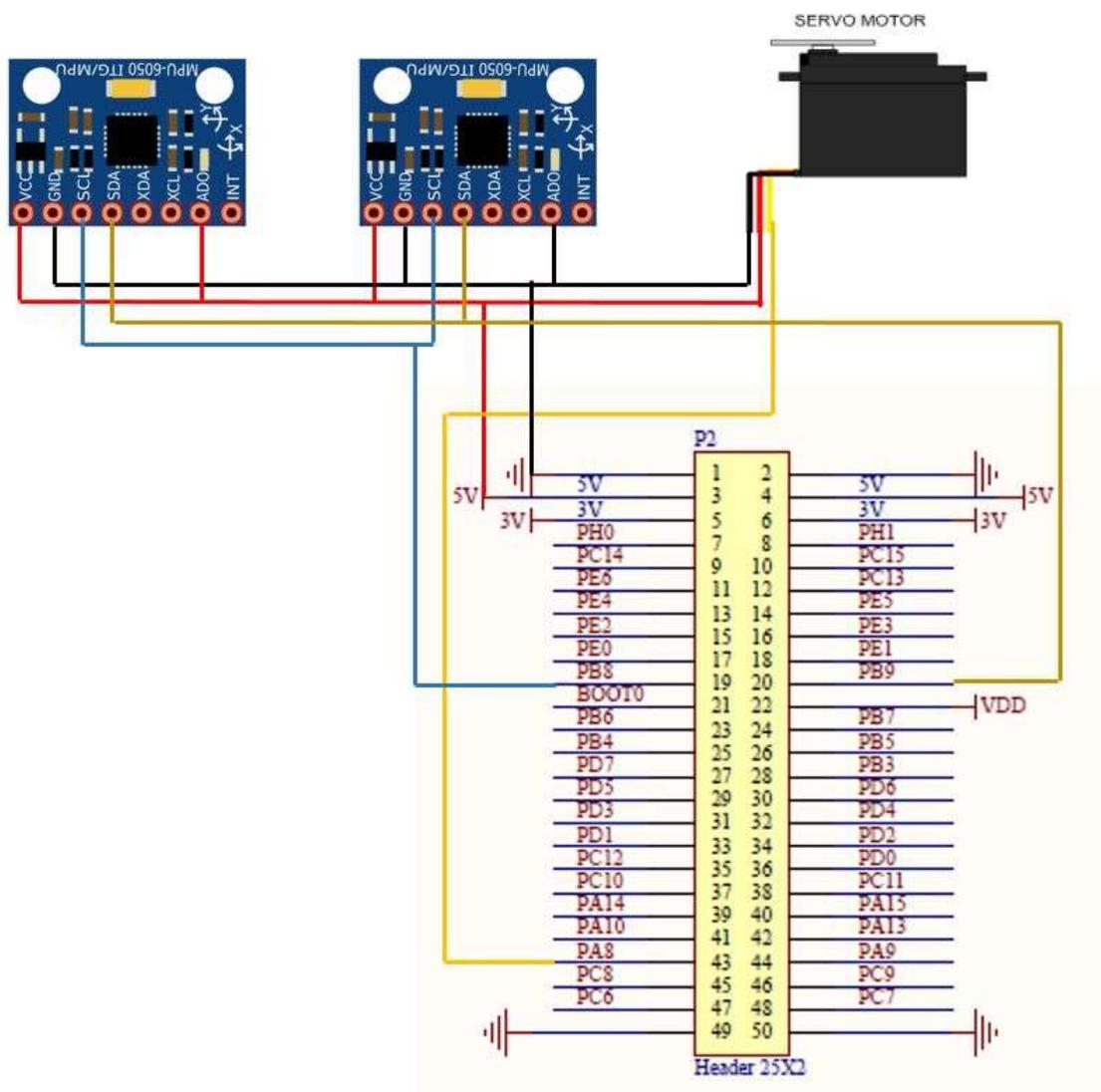
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

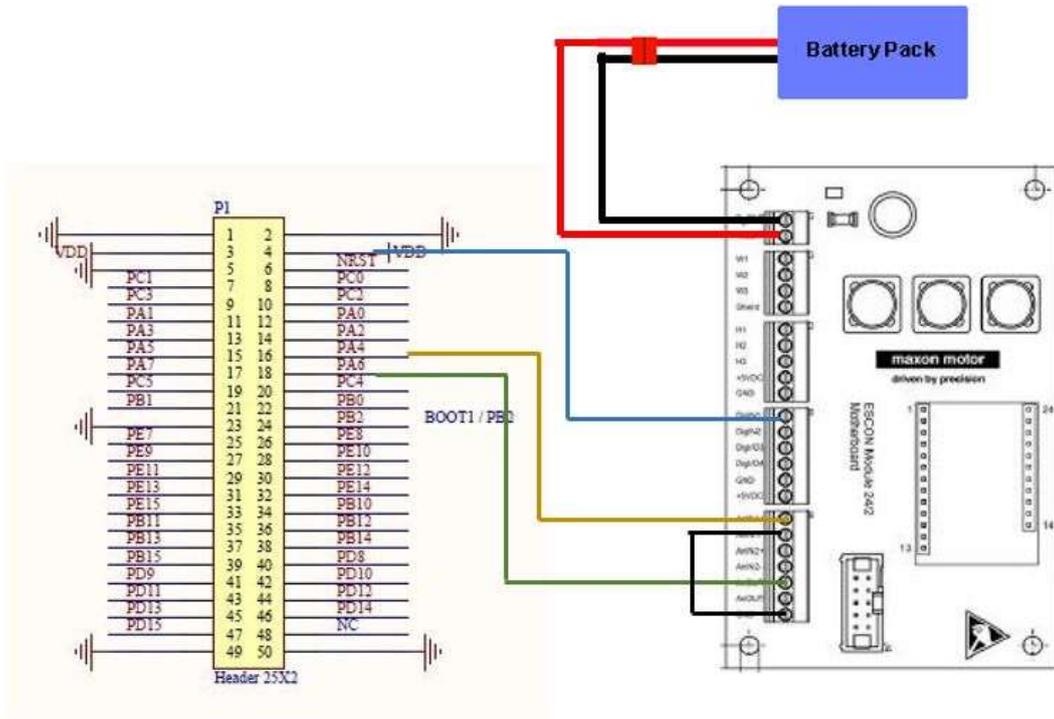
Capítulo 2: Esquemas electrónicos

2.1. Conexión de IMU y servomotor

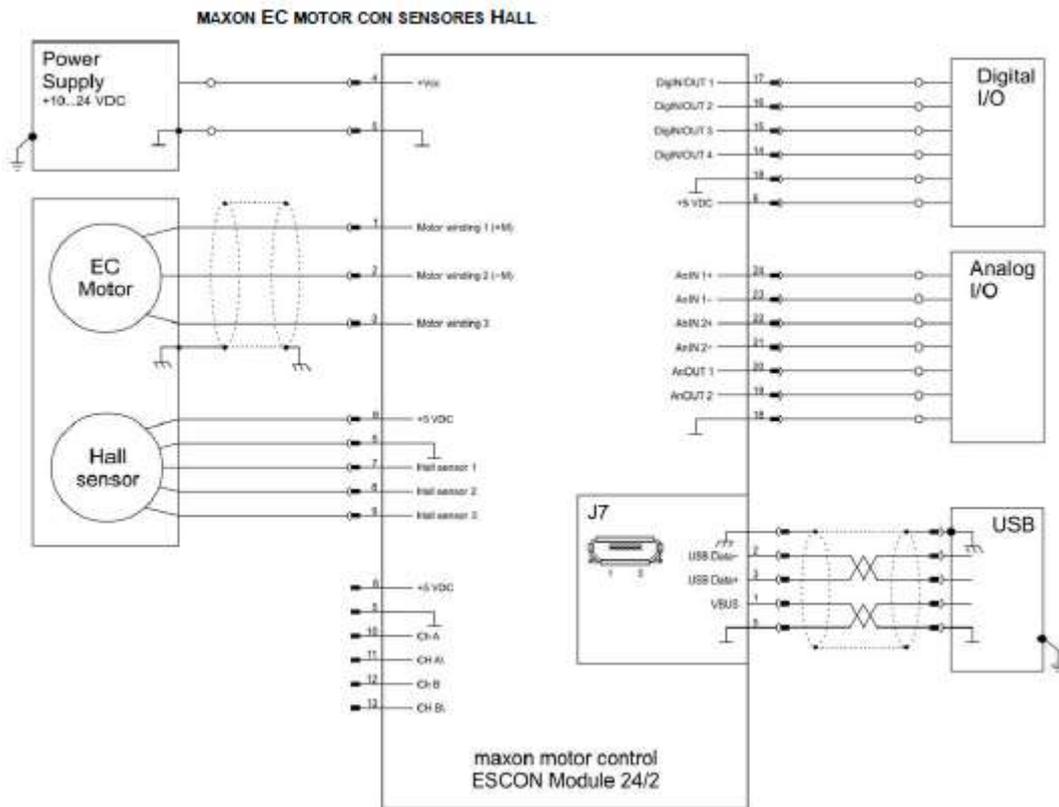


JULIO 2016

2.2. Conexionado ESCON Module 24/2



2.3. Conexión Motor EC



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Parte 3: Presupuesto

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 1: Mediciones

1.1. Componentes principales

CÓDIGO	COMPONENTES	CANTIDAD
0101	Lámina de aluminio 300x300x2 mm	1
0102	Servo MG90s	1
0103	Motor Maxon EC 45 flat 12W, con sensores hall	1
0104	STM32F4DISCOVERY	1
0105	ESCON Module 24/2	1
0106	ESCON Placa madre Module 24/2	1
0107	Batería Li-Po 1000mAh	1
0108	Disco freno minimoto 120 mm	1
0109	Giróscopo/Acelerómetro MPU6050	2

1.2. Componentes del montaje

CÓDIGO	COMPONENTES	CANTIDAD
0201	Juego 10 cables macho-hembra 200 mm	2
0202	Tornillos M3	10
0203	Tuercas M3	4
0204	Tornillos M2	2
0205	Tuercas M2	2
0206	Tornillos M6	3
0207	Tuercas M6	3
0208	Separador 25mm H/M M3	4

JULIO 2016

1.3. Equipo y herramientas

CÓDIGO	ELEMENTO	CANTIDAD
0301	Ordenador	1
0302	Osciloscopio 2 canales	1
0303	Polímetro	1
0304	Máquina de mecanizado automática	1
0305	Torno / Fresadora profesional	1
0306	Soldador	1
0307	Otras herramientas	1

1.4. Software

CÓDIGO	PROGRAMA	CANTIDAD
0401	Matlab / Simulink	1
0402	Solid Edge ST7	1
0403	Escon Studio	1
0404	VCarve Pro 6.0	1

1.5. Mano de obra

CÓDIGO	PROGRAMA	HORAS
0501	Diseño hardware	100
0502	Montaje hardware	5
0503	Diseño del control	150
0504	Pruebas y solución de problemas	300
0505	Documentación del proyecto	100

Capítulo 2: Precios unitarios

2.1. Componentes principales

CÓDIGO	COMPONENTES	€/UNIDAD
0101	Lámina de aluminio 300x300x2 mm	14.66
0102	Servo MG90s	4.90
0103	Motor Maxon EC 45 flat 12W, con sensores hall	68.45
0104	STM32F4DISCOVERY	20.78
0105	ESCON Module 24/2	159.84
0106	ESCON Placa madre Module 24/2	83.81
0107	Batería Li-Po 1000mAh	4.81
0108	Disco freno minimoto 120 mm	7.56
0109	Giróscopo/Acelerómetro MPU6050	10.54

2.2. Componentes del montaje

CÓDIGO	COMPONENTES	€/UNIDAD
0201	Juego 10 cables macho-hembra 200 mm	2.90
0202	Tornillos M3	0.0291
0203	Tuercas M3	0.0256
0204	Tornillos M2	0.0291
0205	Tuercas M2	0.0256
0206	Tornillos M6	0.0291
0207	Tuercas M6	0.0256
0208	Separador 25mm H/M M3	0.1324

JULIO 2016

2.3. Equipo y herramientas

CÓDIGO	ELEMENTO	€/UNIDAD
0301	Ordenador	499.00
0302	Osciloscopio 2 canales	456.00
0303	Polímetro	19.70
0304	Máquina de mecanizado automática	400.00
0305	Torno / Fresadora profesional	259.00
0306	Soldador	7.48
0307	Otras herramientas	97.00

2.4. Software

CÓDIGO	PROGRAMA	€/UNIDAD
0401	Matlab / Simulink	134.00
0402	Solid Edge ST7	130.00
0403	Escon Studio	0.00
0404	VCarve Pro 6.0	660.00

2.5. Mano de obra

CÓDIGO	PROGRAMA	€/HORA
0501	Diseño hardware	50.00
0502	Montaje hardware	20.00
0503	Diseño del control	40.00
0504	Pruebas y solución de problemas	60.00
0505	Documentación del proyecto	40.00

Capítulo 3: Sumas parciales

3.1. Componentes principales

CÓDIGO	COMPONENTES	CANTIDAD	€/UNIDAD	COSTE TOTAL (€)
0101	Lámina de aluminio 300x300x2 mm	1	14.66	14.66
0102	Servo MG90s	1	4.90	4.90
0103	Motor Maxon EC 45 flat 12W, con sensores hall	1	68.45	68.45
0104	STM32F4DISCOVERY	1	20.78	20.78
0105	ESCON Module 24/2	1	159.84	159.84
0106	ESCON Placa madre Module 24/2	1	83.81	83.81
0107	Batería Li-Po 1000mAh	1	4.81	4.81
0108	Disco freno minimoto 120 mm	1	7.56	7.56
0109	Giróscopo/Acelerómetro MPU6050	2	10.54	21.08
			Total	364.81

3.2. Componentes del montaje

CÓDIGO	COMPONENTES	CANTIDAD	€/UNIDAD	COSTE TOTAL (€)
0201	Juego 10 cables macho-hembra 200 mm	2	2.90	4.80
0202	Tornillos M3	10	0.0291	0.291
0203	Tuercas M3	4	0.0256	0.124
0204	Tornillos M2	2	0.0291	0.0582
0205	Tuercas M2	2	0.0256	0.0512
0206	Tornillos M6	3	0.0291	0.0873
0207	Tuercas M6	3	0.0256	0.0768
0208	Separador 25mm H/M M3	4	0.1324	0.5296

JULIO 2016

	Total	6.02
--	--------------	-------------

3.3. Equipo y herramientas

CÓDIGO	ELEMENTO	CANTIDAD	€/UNIDAD	COSTE TOTAL (€)
0301	Ordenador	1	499.00	499.00
0302	Osciloscopio 2 canales	1	456.00	456.00
0303	Polímetro	1	19.70	19.70
0304	Máquina de mecanizado automática	1	400.00	400.00
0305	Torno / Fresadora profesional	1	259.00	259.00
0306	Soldador	1	7.48	7.45
0307	Otras herramientas	1	97.00	97.00
			Total	1,738.15

3.4. Software

CÓDIGO	PROGRAMA	CANTIDAD	€/UNIDAD	COSTE TOTAL (€)
0401	Matlab / Simulink	1	134.00	130.00
0402	Solid Edge ST7	1	130.00	130.00
0403	Escon Studio	1	0.00	0.00
0404	VCarve Pro 6.0	1	660.00	660.00
			Total	924.00

3.5. Mano de obra

	PROGRAMA	HORAS	€/UNIDAD	COSTE TOTAL (€)
0501	Diseño hardware	100	50.00	5,000
0502	Montaje hardware	5	20.00	100
0503	Diseño del control	150	40.00	6,000
0504	Pruebas y solución de problemas	300	60.00	18,000
0505	Documentación del proyecto	100	40.00	4,000
			Total	33,100.00

UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI) – INGENIERO INDUSTRIAL

JULIO 2016

(Esta página se ha dejado en blanco a propósito)

Capítulo 4: Presupuesto general

CONCEPTO	COSTE (€)
COMPONENTES PRINCIPALES	364.81
COMPONENTES DEL MONTAJE	6.02
EQUIPO Y HERRAMIENTAS	1,738.15
SOFTWARE	924.00
MANO DE OBRA	33,100.00
TOTAL	36,132.98