

Universidad Nacional de Ingeniería

Facultad de Ciencias



Técnicas físicas para la industria

“Control de un péndulo invertido mediante una rueda de reacción”

Elaborado por:
Jesus Antonio Tapia Gallardo.

Profesor:
Abel Gutarra Espinoza.

Ciclo 2022-2

Resumen

En el presente informe realizaremos el análisis físico del sistema rueda-marco de un péndulo invertido y veremos las condiciones de equilibrio para la elección del controlador, luego simularemos el sistema en Matlab para obtener el mejor performance considerando el tiempo de asentamiento. Después, diseñaremos la rueda de reacción y el marco, para finalmente implementar el sistema total y realizar las pruebas de control.

Keywords: Control, rueda de reacción, péndulo.

Abstract

In this report we will perform the physical analysis of the wheel-frame system of an inverted pendulum and we will see the equilibrium conditions for the choice of the controller, then we will simulate the system in Matlab to obtain the best performance considering the settling time. Then, we will design the reaction wheel and the frame, to finally implement the total system and perform the control tests.

Keywords: Control, reaction wheel, pendulum.

Índice

Resumen	1
Índice	2
Índice de figuras	3
1. Introducción	4
1.1. Antecedentes	4
1.2. Objetivo General	4
1.3. Objetivos Específicos	5
2. Fundamento Teórico	5
2.1. Modelamiento del sistema	5
2.2. Diseño de control	6
2.2.1. Control PD	6
2.2.2. Control PID	7
2.3. Motor Brushless	7
2.4. Acelerómetro-Giroscopio	8
2.5. Protocolo I2C	8
3. Diseño del sistema	9
3.1. Componentes de Hardware	9
3.2. Software	10
3.3. Simulación del sistema marco-rueda	10
3.4. Diseño de la rueda	11
3.5. Diseño del marco	12
3.6. Implementación	13
4. Resultados	13
5. Discusión de resultados	14
6. Conclusiones	15

Índice de figuras

1.	Cubli haciendo equilibrio en una esquina(Muehlebach et al., 2013).	4
2.	Vista lateral del sistema implementado en (Belascuen y Aguilar, 2018).	4
3.	Diagrama esquemático de un péndulo invertido 1D basado en una rueda de reacción(Muehlebach et al., 2013).	5
4.	Sistema de control en lazo cerrado (Sandeep, 2021).	6
5.	Motor DC Brushless (“What are Brushless DC Motors”, s.f.).	7
6.	Los cambios en la aceleración producen cambios en la capacitancia (C1 y C2)(“Aplicar la fusión de sensores a acelerómetros y giroscopios”, s.f.).	8
7.	Microestructura general de un giroscopio(Dejan, 2015).	8
8.	Conexión I2C (“I2C - Puerto, Introducción, trama y protocolo”, 2017).	9
9.	Envío de datos mediante el protocolo I2C (“Diferencias de los protocolos de comunicación UART vs I2C vs SPI”, s.f.).	9
10.	Respuesta en el tiempo al controlador u .	11
11.	Respuesta en el tiempo al controlador no lineal.	11
12.	Detalle de la rueda de referenciaRuss et al., 2019 mostrando el momento de inercia bajo (izquierda) y el momento de inercia alto (derecha).	12
13.	Rueda diseñada.	12
14.	Marco diseñado de .	12
15.	Diagrama electrónico.	13
16.	Prototipo final.	13
17.	Ángulo (azul) y señal de control(rojo).	14
18.	Ángulo (azul) y señal de control(rojo) para el caso real.	14

1. Introducción

1.1. Antecedentes

Existen múltiples trabajos en donde se usan ruedas de reacción para equilibrar un sistema, por ejemplo, Cubli es un sistema de 3 ruedas de reacción que permiten que un cubo sea capaz de balancearse en uno de sus vértices (ver figura 1) en Muehlebach et al., 2013 se hace el modelado matemático del sistema así como el diseño del controlador para el caso bidimensional (péndulo invertido) y 3D.

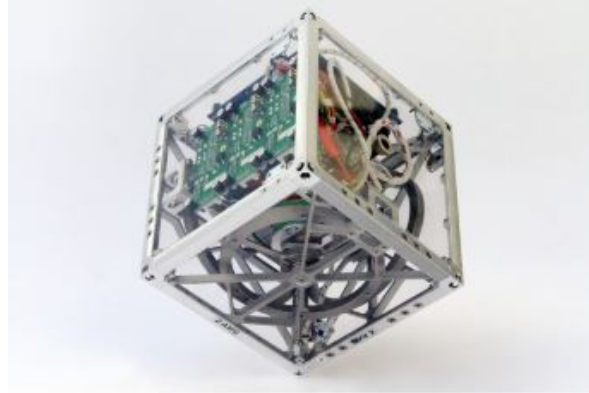


Figura 1: Cubli haciendo equilibrio en una esquina(Muehlebach et al., 2013).

En el trabajo de Belascuen et al. Belascuen y Aguilar, 2018 realizaron la selección de parámetros de una rueda de reacción para el óptimo control de un péndulo invertido(ver figura 2), además hicieron el desarrollo matemático para comparar el performance de distintos motores.



Figura 2: Vista lateral del sistema implementado en (Belascuen y Aguilar, 2018).

Un trabajo que se enfoca en el control es el de Mark W. Spong et al. en donde usan distintos algoritmos de control no lineal para que el péndulo se encuentre en posición vertical Spong et al., 2001.

1.2. Objetivo General

- Realizar la implementación de un péndulo invertido con el fin de conseguir que el sistema se encuentre en posición vertical mediante el control de la aceleración de la rueda de reacción.

1.3. Objetivos Específicos

- Modelar el sistema físico para realizar el análisis matemático de un péndulo invertido.
- Simular el sistema considerando controladores PID y los propuestos en la literatura.
- Diseñar y realizar la impresión 3D de la rueda de reacción y del marco.
- Implementar el sistema total y realizar las pruebas usando control PID.

2. Fundamento Teórico

2.1. Modelamiento del sistema

Para el análisis del péndulo invertido se ha tomado como referencia el trabajo de Michael Muehlebach et al Muehlebach et al., [2013](#).

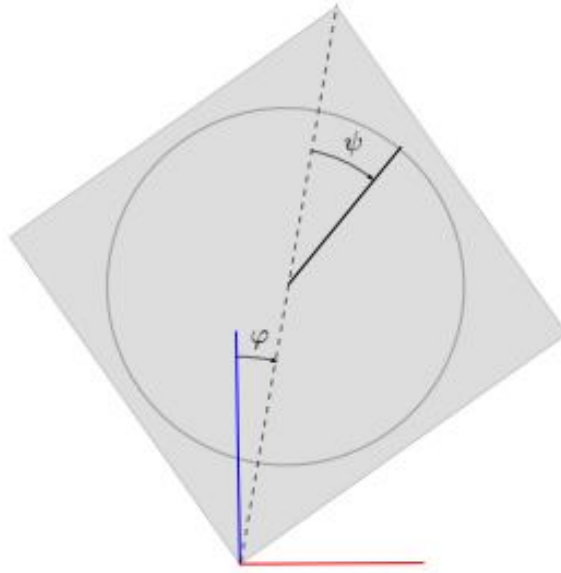


Figura 3: Diagrama esquemático de un péndulo invertido 1D basado en una rueda de reacción(Muehlebach et al., [2013](#)).

Sea φ y ψ que describen la posición del péndulo tal y como se muestra en la figura [3](#). Denotaremos como θ_w el momento de inercia de la rueda de reacción y θ_0 será el momento del sistema alrededor del punto pivote en el marco de coordenadas fijas del cuerpo, m_{tot} y l representan la masa total masa y la distancia entre el punto de pivote y el centro de gravedad de todo el sistema.

El lagrangiano del sistema viene dado por:

$$\mathcal{L} = \frac{1}{2}\hat{\theta}_0\dot{\varphi} + \frac{1}{2}\theta_w(\dot{\varphi} + \dot{\psi})^2 - mg\cos(\varphi) \quad (1)$$

siendo $\hat{\theta}_0 = \theta_0 - \theta_w$, $m = m_{total} \cdot l$ y g la constante de la gravedad, los momentos generalizados serán:

$$p_\varphi = \frac{\partial \mathcal{L}}{\partial \dot{\varphi}} = \theta_0\dot{\varphi} + \theta_w\dot{\psi} \quad (2)$$

$$p_\psi = \frac{\partial \mathcal{L}}{\partial \dot{\psi}} = \theta_w(\dot{\varphi} + \dot{\psi}) \quad (3)$$

Denotaremos T como el torque aplicado a la rueda de reacción por el motor. Entonces, las ecuaciones del movimiento usando la mecánica lagrangiana serán:

$$\dot{p}_\varphi = \frac{\partial \mathcal{L}}{\partial \varphi} = mgsin(\varphi) \quad (4)$$

$$\dot{p}_\psi = \frac{\partial \mathcal{L}}{\partial \psi} + T = T \quad (5)$$

Debido a que la posición de la rueda no es relevante para el análisis introducimos en su lugar a $x = (\varphi, p_\varphi, p_\psi)$ para representar la dinámica del sistema.

$$\dot{x} = \begin{pmatrix} \dot{\varphi} \\ \dot{p}_\varphi \\ \dot{p}_\psi \end{pmatrix} = f(x, T) = \begin{pmatrix} \hat{\theta}_0^{-1}(p_\varphi - p_\psi) \\ mgsin(\varphi) \\ T \end{pmatrix} \quad (6)$$

Para que el sistema se encuentre en equilibrio se debe cumplir $\varphi = 0$ y $p_\psi = cte$. Mecánicamente esto significa que el péndulo se encuentra en posición vertical y que la rueda gira con una velocidad angular constante.

2.2. Diseño de control

Los algoritmos de control más usados son los que combina los tres modos básicos de control, es decir, los modos proporcional (P), derivativo (D) e integral (I). En la figura 4 se muestra el sistema a usar.

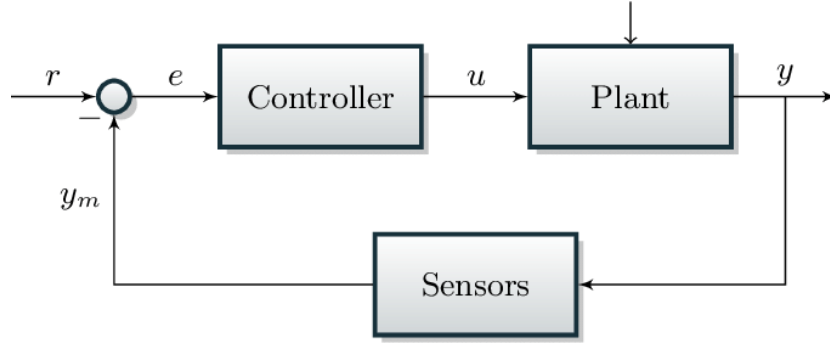


Figura 4: Sistema de control en lazo cerrado (Sandeep, 2021).

2.2.1. Control PD

Un controlador PD reduce los transitorios como el tiempo de subida, el sobreimpulso y las oscilaciones en la salida. Útil para cambiar la magnitud y desea agregar un adelanto de fase a la salida (Sandeep, 2021). El controlador PD en el dominio del tiempo se describe mediante la relación:

$$u(t) = k_p \cdot e + k_d \cdot \frac{de}{dt}$$

siendo e la diferencia entre el valor deseado y el obtenido y u la ley de control.

2.2.2. Control PID

En este caso se agrega el componente integral al controlador anterior, esto produce que el error en estado estable tienda a cero. El controlador se describe como:

$$u(t) = k_p \cdot e + k_d \cdot \frac{de}{dt} + k_i \cdot \int e dt$$

La elección de las constantes k_p, k_i y k_d se puede hacer mediante el análisis de simulación (tantear y comprobar) o usando el método de Ziegler Nichols.

El método Ziegler-Nichols es otro método popular para sintonizar un controlador PID. Es muy similar al método de prueba y error en el que k_i y k_d se establecen en cero y k_p se aumenta hasta que el ciclo comienza a oscilar. Una vez que comienza la oscilación, se anotan la ganancia crítica k_c y el período de oscilaciones P_c . Las constantes luego se ajustan según la tabla 1 (“Teoria PID explicada”, 2006).

Control	k_p	k_i	k_d
P	$0.5K_c$	-	-
PI	$0.45K_c$	$0.54K_c/P_c$	-
PD	$0.8K_c$	-	$0.1K_cP_c$
PID	$0.6K_c$	$1.2K_c/P_c$	$0.075K_cP_c$

Cuadro 1: Tuneo Ziegler-Nichols, utilizando el método de oscilación.

2.3. Motor Brushless

Es un motor síncrono que utiliza un controlador electrónico para conmutar las corrientes de CC a los devanados del motor produciendo campos magnéticos que giran efectivamente en el espacio y a los que sigue el rotor.

El rotor es un imán permanente; las bobinas no giran, sino que están fijas en el estator. Como las bobinas no se mueven, no se necesitan escobillas ni conmutador (ver figura 5). La rotación se consigue cambiando la dirección de los campos magnéticos generados por las bobinas estacionarias. Para controlar la rotación, se ajusta la magnitud y la dirección de la corriente en estas bobinas (“What are Brushless DC Motors”, s.f.).

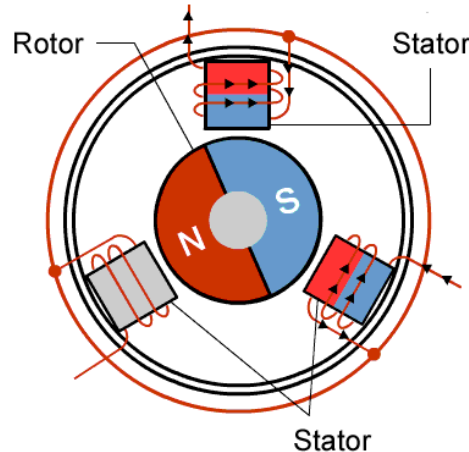


Figura 5: Motor DC Brushless (“What are Brushless DC Motors”, s.f.).

2.4. Acelerómetro-Giroscopio

Como el parámetro que controlaremos es el ángulo que forma la horizontal con el frame(φ en la figura 3) lo obtendremos de forma indirecta mediante el acelerómetro y giroscopio.

Los acelerómetros MEMS (sistemas microelectromecánicos) utilizan placas de silicio estacionarias y resortes mecánicos que responden a fuerzas externas (ver figura 6), de tal forma que a partir del cambio en la capacitancia se puede hallar la aceleración (“Aplicar la fusión de sensores a acelerómetros y giroscopios”, [s.f.](#)).

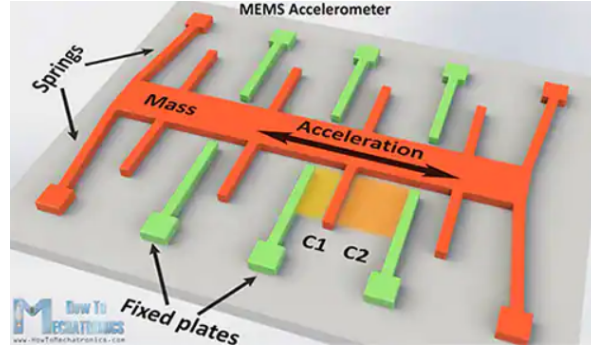


Figura 6: Los cambios en la aceleración producen cambios en la capacitancia (C1 y C2)(“Aplicar la fusión de sensores a acelerómetros y giroscopios”, [s.f.](#)).

El giroscopio mide la velocidad angular utilizando el efecto Coriolis. De manera similar al acelerómetro se producirá una fuerza (flecha roja -azul de la figura 7) que causará el desplazamiento perpendicular de la masa, este desplazamiento causará un cambio en la capacitancia que será medida y corresponderá a una velocidad angular particular (Dejan, [2015](#)).

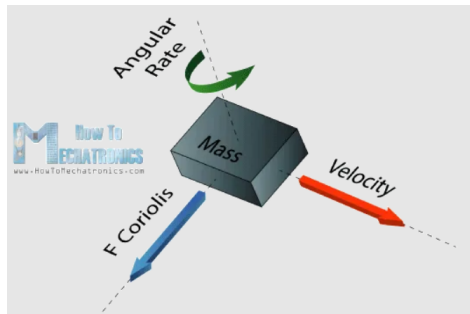


Figura 7: Microestructura general de un giroscopio(Dejan, [2015](#)).

2.5. Protocolo I2C

El acelerómetro elegido fue el MPU6050 el cual se comunica con el microcontrolador mediante el protocolo Inter-Integrated Circuit(I2C). Este protocolo usa dos hilos SCL (señal de reloj) y SDA (señal de datos), tiene la presencia de un Maestro (se encarga de iniciar y parar la comunicación) y de uno o más Esclavos (suelen ser sensores). En la figura 8 se muestra las conexiones de un maestro con tres esclavos.

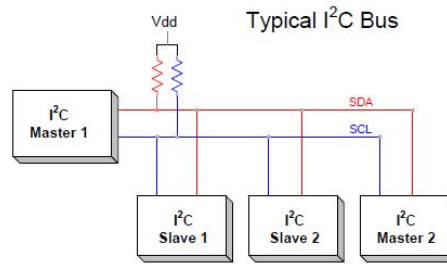


Figura 8: Conexión I2C (“I2C - Puerto, Introducción, trama y protocolo”, 2017).

El maestro envía a cada esclavo la dirección de 7 o 10 bits del esclavo y un bit de lectura/escritura al esclavo con el que desea comunicarse. El esclavo comparará la dirección con la suya. Si la dirección coincide, el esclavo devolverá un bit ACK que cambia la línea SDA a baja por un bit, en caso contrario pasa a alto. Después, el maestro enviará o recibirá la trama de datos. Al acabar esta acción, la transferencia de datos, el dispositivo receptor devuelve otro bit ACK al remitente para reconocer la transmisión exitosa. Por otro lado, para detener la transmisión de datos, el maestro envía una señal de parada al esclavo cambiando SCL alto antes de cambiar SDA alto (“Diferencias de los protocolos de comunicación UART vs I2C vs SPI”, s.f.). Todo lo mencionado se puede entender mejor usando la figura 9, en donde se envía la dirección de 7 bits junto con el bit de escritura o lectura (R/W) y por último el envío de los datos y la señal de parada (stop).

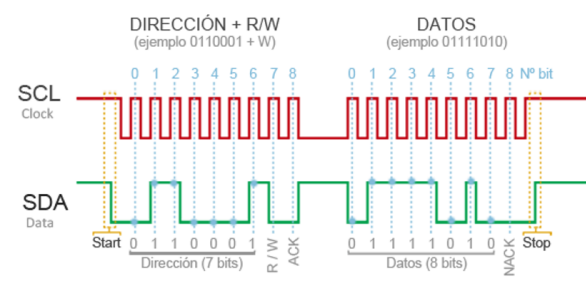


Figura 9: Envío de datos mediante el protocolo I2C (“Diferencias de los protocolos de comunicación UART vs I2C vs SPI”, s.f.).

3. Diseño del sistema

En esta sección desarrollaremos lo correspondiente a los componentes de software y hardware, las simulaciones, los diseños en Inventor para la impresión 3D y por último la implementación.

3.1. Componentes de Hardware

- **Motor NTM Prop Drive 28-36 1000KV.** Es un motor brushless de 1000KV(rpm/V) que puede trabajar hasta con 14V y soporta hasta 30A (“NTM Prop Drive 28-36 1000KV / 400W”, s.f.).
- **Controlador de velocidad ESC.** Nos permite el control electrónico de velocidad de motor brushless mediante señales PWM. Trabaja con un voltaje de entrada 5 V-12 V y puede soportar hasta 30A de intensidad de corriente eléctrica (“ESC Controlador de Velocidad para Motor Brushless”, s.f.).

- **MPU6050.** Este módulo permite medir el movimiento en 6 grados de libertad, combinando un giroscopio de 3 ejes y un acelerómetro de 3 ejes en un mismo chip. Integra un DMP (procesador digital de movimiento) capaz de realizar complejos algoritmos de captura de movimiento de 9 ejes. Entre las especificaciones técnicas más relevantes tenemos que el voltaje de operación es entre 3.3-5V, el rango del acelerómetro es 2g/4g/8g/16g (g es la aceleración de la gravedad) y en el caso del giroscopio posee un rango programable de 250/500/1000/2000°/s (“Módulo MPU6050: Acelerómetro, Giroscopio I2C”, [s.f.](#)).
- **Arduino UNO R3.** Es una placa basada en un microcontrolador ATMEL. Será usada para la lectura de datos del MPU6050 y para enviar la señal de control al ESC.
- **Marco y Rueda.** La aceleración del giro de la rueda de reacción provocará que el marco gire en dirección opuesta por conservación del momento angular.
- **Fuente DC.** Es necesario el uso de una fuente de 12V capaz de soportar picos de corriente de hasta 10A.

3.2. Software

- **MATLAB.** Es una plataforma de programación y cálculo numérico, la usaremos para la simulación de la respuesta de control considerando un controlador no lineal.
- **Arduino IDE.** Es un entorno de desarrollo que permite escribir, depurar, editar y grabar nuestro programa en la placa Arduino. El lenguaje de programación es similar a ANSI C.
- **Autodesk Inventor.** Es un software de modelado paramétrico de sólidos en 3D. Será usado para el diseño de la rueda y el marco.

3.3. Simulación del sistema marco-rueda

En esta sección analizaremos el comportamiento usando la ecuación 6:

$$\dot{x} = g(x) + h(x) \cdot u, \quad y = \varphi$$

Siendo y la salida deseada y $u = T$ el controlador usado. Realizando la segunda derivada de y para que aparezca la señal del controlador obtenemos:

$$\ddot{y} = \hat{\theta}_0^{-1} m g \sin(\varphi) - \hat{\theta}_0^{-1} u$$

Considerando un control de linealización de retroalimentación:

$$u = \hat{\theta}_0(\hat{\theta}_0^{-1} m g \sin(\varphi) + \nu)$$

$$\ddot{y} = \nu = k_p y + k_d \dot{y}$$

Siendo ν el controlador auxiliar escogido convenientemente, resultando finalmente:

$$u = m g \sin(\varphi) + K_p y + K_d \dot{y} + 0,2 p_\psi \quad (7)$$

Con estas consideraciones de la dinámica del sistema, podemos simular el comportamiento usando la función `ode45` (“Resolver ecuaciones diferenciales no rígidas; método de orden intermedio - MATLAB

ode45 - MathWorks América Latina”, [s.f.](#)) . En la figura 10 se muestra la respuesta al controlador u durante 10 segundos.

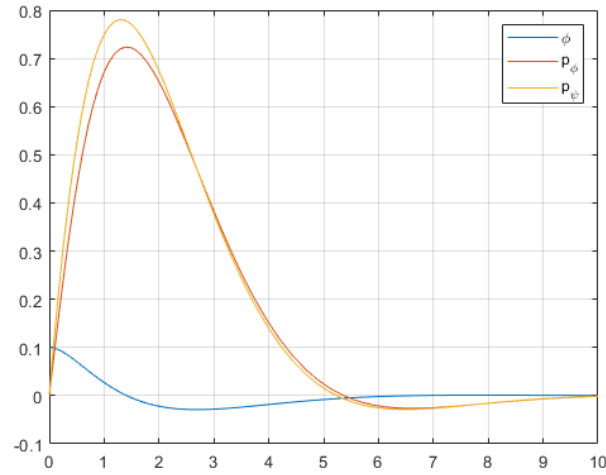


Figura 10: Respuesta en el tiempo al controlador u .

Usando el controlador no lineal planteado en Muehlebach et al., 2013, se obtiene lo mostrado en la figura 11. Notamos que en este caso el tiempo en el cual se alcanza el valor de equilibrio es mayor pero la amplitud durante los primeros segundos es menor, para la elección correcta es necesario realizar pruebas ya que la elección del motor (torque y potencia) definirá cual se comporta mejor.

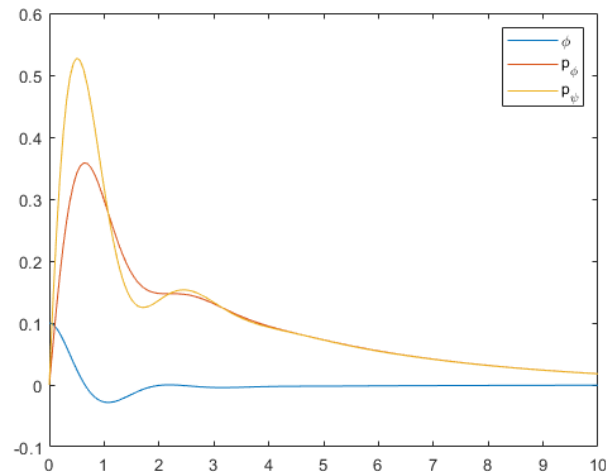


Figura 11: Respuesta en el tiempo al controlador no lineal.

3.4. Diseño de la rueda

Para el diseño de la rueda se ha considerado el modelo planteado por Andrew Russ et al. Russ et al., 2019 el cual se muestra en la figura 12.

Nuestra rueda tiene como diametro exterior de 13.5cm y uno interior de 10.4cm, en la figura 13 se observa el modelo STL usado para realizar la impresión 3D.

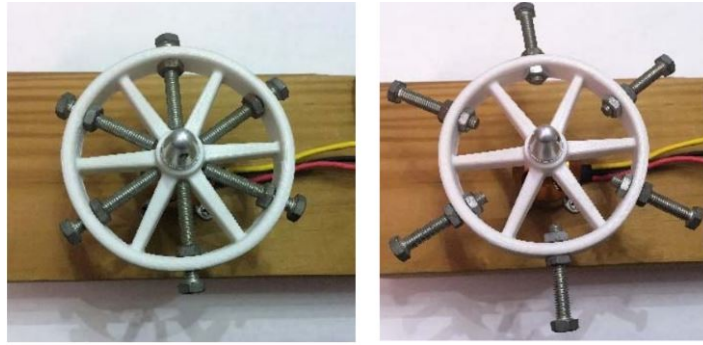


Figura 12: Detalle de la rueda de referencia Russ et al., 2019 mostrando el momento de inercia bajo (izquierda) y el momento de inercia alto (derecha).

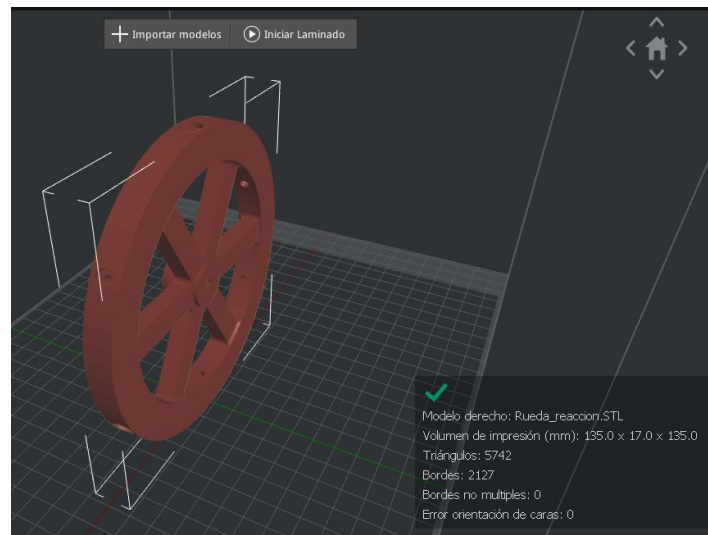


Figura 13: Rueda diseñada.

3.5. Diseño del marco

Para el diseño de marco se usó como referencia el trabajo de Cochrane, 2021, en la figura 14 se muestra el renderizado 3D de la pieza.

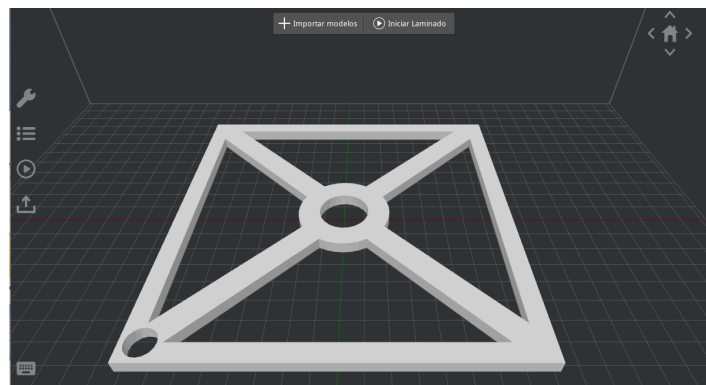


Figura 14: Marco diseñado de .

3.6. Implementación

En la figura 15 se muestran las conexiones electrónicas. El prototipo final se puede apreciar en la figura 16.

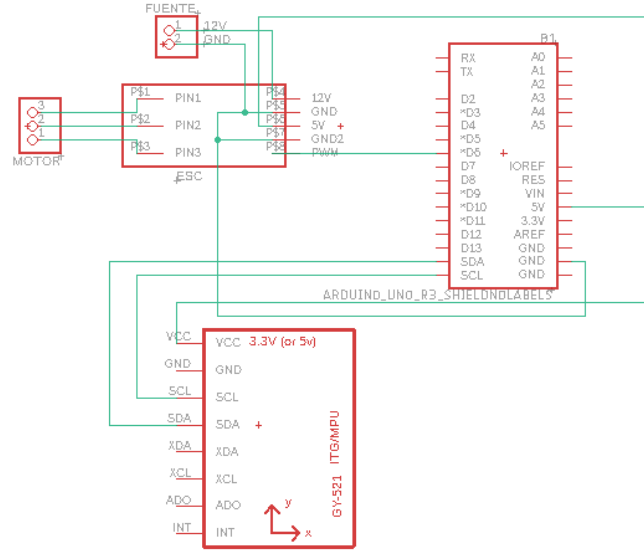
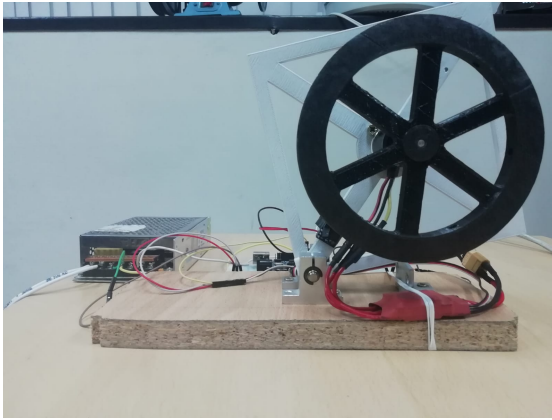
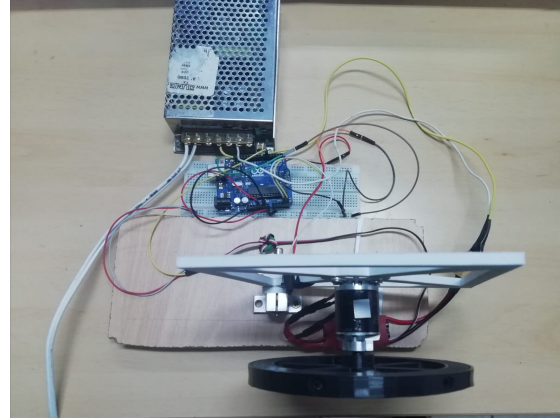


Figura 15: Diagrama electrónico.



(a) Vista de frente



(b) Vista superior.

Figura 16: Prototipo final.

4. Resultados

Para las pruebas se ha usado control PID, consideranco como setpoint un ángulo de 21 grados. La señal de control será la siguiente:

$$u = K_P \cdot error + K_I \cdot (error)dt + K_D \cdot (error - lastError)/dt$$

siendo $error = \psi - 21$. El tiempo transcurrido se halla mediante la función `millis()` de Arduino. Para controlar el motor se usa la librería `Servo` y la función `Write`, esta función tiene como entrada un valor entre 0 y 180, sin embargo a partir de las pruebas los valores deben estar entre 0 y 80 ya que en caso de

tomar un valor mayor la velocidad del motor es tan alta que la vibración impide al MPU leer el ángulo correctamente. En la figura 18 se muestra el ángulo y la señal (u) cuando se movió manualmente el marco para comprobar el correcto funcionamiento del programa.

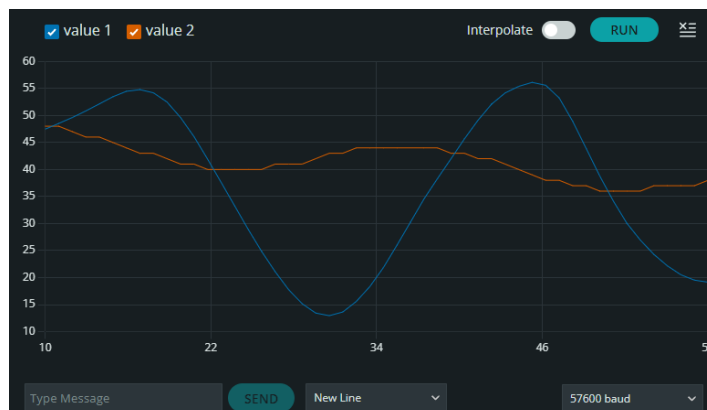


Figura 17: Ángulo (azul) y señal de control(rojo).

En la figura se observa las mismas señales pero sin intervención externa, después de aproximadamente 10s las vibraciones causan que el MPU se 'reinicie' y empieza a arrojar el valor cero grados a pesar de girar un ángulo, esto produce que la señal vaya a su máximo valor y que por lo tanto el motor gire a su máxima velocidad produciendo una vibración aún más grande.

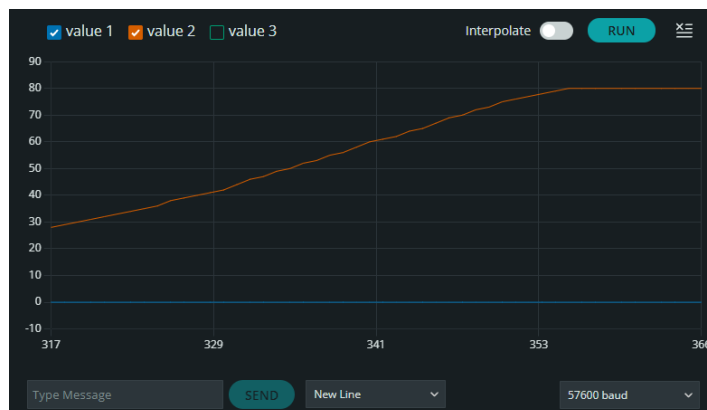


Figura 18: Ángulo (azul) y señal de control(rojo) para el caso real.

Los códigos para la calibración, el cálculo del ángulo de giro y el control PID han sido subidos al siguiente GitHub: <https://github.com/JesusTapia-dev/PenduloInvertido-UNI>.

5. Discusión de resultados

El sistema no pudo llegar a estar en posición vertical(oscila entre ambos extremos) a pesar de la variación de parámetros del control PID y del cambio de controlador a uno no lineal($u = mgsin(\varphi) + K_p \cdot \varphi + K_d \cdot error$), las gráficas que nos proporciona Arduino(Serial plotter) y las simulaciones realizadas en MATLAB nos indican que el problema no se encuentra en la programación así que pasaremos a analizar la parte mecánica.

Uno de los problemas observados respecto al motor es su incapacidad de hacer un aumento progresivo de la velocidad lo que conlleva que solo dará una respuesta adecuada cuando el marco se

encuentre en uno de los extremos, esto a su vez produce que los movimientos sean bruscos y que el MPU lea valores inadecuados o incluso incoherentes (ha llegado a leer ángulos mayores a 90 grados a pesar de que por la posición del sensor el ángulo máximo de giro posible es 79 grados), para solucionar este problema se plantea el uso de otro motor brushless con encoder como lo puede ser iPower Motor GM4108H-120T.

6. Conclusiones

- Se obtuvieron las ecuaciones mecánicas del sistema rueda-marco así como la función de transferencia del motor para su posterior análisis.
- Se realizó la simulación del sistema usando distintos tipos de controladores para observar el comportamiento de las variables controladas.
- Se diseñó e imprimió en 3D la rueda de reacción y el marco.
- Se implementó el sistema y realizaron las pruebas.
- El sistema no fue capaz de mantenerse en posición vertical debido a la vibración y al motor.

Referencias

- Aplicar la fusión de sensores a acelerómetros y giroscopios. (s.f.). <https://www.digikey.com/es/articles/apply-sensor-fusion-to-accelerometers-and-gyroscopes>
- Belascuen, G., & Aguilar, N. (2018). Design, Modeling and Control of a Reaction Wheel Balanced Inverted Pendulum. *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, 1-9. <https://doi.org/10.1109/ARGENCON.2018.8646093>
- Cochrane, J. (2021). DIY—1D-Wannabe-Cubli-Reaction-Wheel. <https://github.com/squirtifier/DIY---1D-Wannabe-Cubli-Reaction-Wheel>
- Dejan. (2015). What is MEMS? Accelerometer, Gyroscope & Magnetometer with Arduino. <https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>
- Diferencias de los protocolos de comunicación UART vs I2C vs SPI. (s.f.). <https://solectroshop.com/es/blog/diferencias-de-los-protocolos-de-comunicacion-uart-vs-i2c-vs-spi-n107>
- ESC Controlador de Velocidad para Motor Brushless [Recuperado: 4-1-2023]. (s.f.). https://electropro.pe/index.php?route=product/product&product_id=539
- I2C - Puerto, Introducción, trama y protocolo. (2017). <https://hetpro-store.com/TUTORIALES/i2c/>
- Módulo MPU6050: Acelerómetro, Giroscopio I2C. (s.f.). <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/33-modulo-mpu6050-acelerometro-giroscopio-i2c.html>
- Muehlebach, M., Mohanarajah, G., & D'Andrea, R. (2013). Nonlinear analysis and control of a reaction wheel-based 3D inverted pendulum. *52nd IEEE Conference on Decision and Control*.
- NTM Prop Drive 28-36 1000KV / 400W [Recuperado: 8-1-2023]. (s.f.). https://hobbyking.com/es_es/propdrive-v2-2836-1000kv-brushless-outrunner-motor.html
- Resolver ecuaciones diferenciales no rígidas; método de orden intermedio - MATLAB ode45 - MathWorks América Latina. (s.f.). <https://la.mathworks.com/help/matlab/ref/ode45.html>
- Russ, A., Russ, S. H., Spencer, E., & Frank, M. (2019). Motor Controller and Reaction Wheel for CubeSat. *2019 SoutheastCon*, 1-3. <https://doi.org/10.1109/SoutheastCon42311.2019.9113490>

- Sandeep, V. M. (2021). When and why to use P, PI, PD and PID Controller? - Sandeep VM. <https://medium.com/@svm161265/when-and-why-to-use-p-pi-pd-and-pid-controller-73729a708bb5>
- Spong, M. W., Corke, P., & Lozano, R. (2001). Nonlinear control of the Reaction Wheel Pendulum. *Automatica*, 37(11), 1845-1851. [https://doi.org/https://doi.org/10.1016/S0005-1098\(01\)00145-5](https://doi.org/https://doi.org/10.1016/S0005-1098(01)00145-5)
- Teoria PID explicada. (2006). <https://www.ni.com/es-cr/innovations/white-papers/06/pid-theory-explained.html>
- What are Brushless DC Motors [Recuperado: 7-1-2023]. (s.f.). <https://www.renesas.com/us/en/support/engineer-school/brushless-dc-motor-01-overview>