



Universidad Nacional de San Luis
Facultad de Ingeniería y Ciencias Agropecuarias

DISEÑO, IMPLEMENTACIÓN Y CONTROL DE UN PÉNDULO INVERTIDO LINEAL

ROMERO, Jorge Maximiliano

Trabajo final de Ingeniería Mecatrónica

Director: Dr. Ing. ASENSIO, Eduardo Maximiliano

Codirector: Ing. BERGOGLIO, Mario Federico

Villa Mercedes, San Luis

2024

DERECHO DE AUTOR

© 2024, Jorge Maximiliano Romero.

Se autoriza la reproducción total, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Se autoriza el cambio de cualquier componente de este proyecto para fines de optimización y mejora de este, excepto el del dispositivo enconder Eltra por su alto costo y fragilidad ante los golpes.

Le dedico este trabajo con profundo agradecimiento y amor eterno a mi madre Graciela y mi padre Jorge que han sido los pilares fundamentales para mi formación por su constante respaldo y mi fuente de inspiración de perseverancia. Su amor, sacrificio y sabias enseñanzas han sido el motor que me impulso a alcanzar cada meta.

AGRADECIMIENTOS

A Dios quien me ha dado la fortaleza necesaria para no perder el camino y perseverar hasta el final.

A mis padres, Graciela y Jorge que me brindaron educación, valores y todos los medios necesarios para ser lo que soy actualmente. Además de su apoyo incondicional para poder seguir adelante a pesar de la distancia. A mi hermana Romina que al igual que mis padres me aportó muchísimo afecto, contribuyó en los estudios de los finales de la carrera y me dio el mejor regalo junto a mi cuñado Diego que son mis dos sobrinas Guadalupe y Ludmila. También quiero agradecer a mi compañera Belén que ha sido muy importante en el final de esta etapa.

Por otro lado, quiero agradecer a mis directores, al Dr. Ing. Eduardo Maximiliano Asensio y al Ing. Mario Federico Bergoglio, los cuales estuvieron presente para aconsejarme y guiarme en este Trabajo Final. Pero por sobre todo a Maximiliano que me abrió las puertas del Laboratorio de Control Automático y nunca dejó de aconsejarme y reprenderme en el caso de que sea necesario para llegar a la meta. A la Universidad Nacional de San Luis por formarme en esta hermosa carrera y darme los medios para recibirme.

Por ultimo y no menos importante, muestro mi eterno agradecimiento a mis compañeros y amigos que me aportaron en todo lo que necesitara a lo largo de todos estos años de mi vida universitaria, me dieron su amistad y los mejores momentos de esta etapa. Mención destacada para mi amigo Dardo que además de haber aportado en todo lo anterior fue clave ya que sin el aporte de su impresora 3D este trabajo no se hubiera podido llevar a cabo.

Si una de estas piezas de este rompecabezas hubiera faltado, nada hubiera sido posible.

Simplemente gracias a todos

RESUMEN

Este trabajo final se centra en el diseño, implementación y control de un péndulo invertido lineal, que aborda el desafío de mantener el péndulo en posición vertical mediante un sistema de control preciso. Se desarrolló un modelo matemático del sistema y se simuló sus comportamientos utilizando herramientas como Matlab y Simulink para validar la respuesta del sistema bajo diversas condiciones. El proyecto incluye la programación de un microcontrolador STM32, configurando entradas, salidas e interrupciones para ejecutar el control en tiempo discreto. A través del software STM32CubeIDE, se implementó un algoritmo de control que regula el ángulo del péndulo. Además, se diseñó la estructura mecánica que soporta el sistema, que asegura la sincronización precisa entre el péndulo y el encoder. Se detallan los componentes como soportes, rodamientos, poleas y finales de carrera, necesarios para el funcionamiento del mecanismo. El informe también describe el proceso de ensamblaje y las pruebas experimentales que se realizaron para evaluar el desempeño del controlador. Se planteó la utilización de técnicas de control en tiempo discreto para la implementación en hardware, lo que permite explorar su aplicación en sistemas dinámicos complejos. El objetivo general es desarrollar un prototipo funcional que integre conocimientos de diversas áreas de la ingeniería, proporcionando una base sólida para el estudio y comprensión de sistemas de control y su aplicación en el campo de la robótica autónoma. Este proyecto representa un esfuerzo por combinar teoría y práctica en el diseño del sistema de control.

Palabras claves — Dinámica, péndulo invertido, sistema de control.

CONTENIDO

CAPITULO 1: Propuesta	9
1.1. Introducción.....	9
1.2. Objetivos	10
1.2.1. Objetivo general.....	10
1.2.2. Objetivo específico.....	10
1.3. Alcances y limitaciones	10
1.4. Marco teórico.....	10
1.4.1. Clasificación de péndulo invertido.....	10
1.4.2. Definición y descripción del péndulo invertido.....	12
1.4.3. Importancia y relevancia del estudio del péndulo invertido en ingeniería	13
1.4.4. Dinámica del péndulo invertido	14
1.4.5. Equilibrio.....	15
1.4.6. Sistemas de control	17
1.5. Justificación.....	21
1.6. Antecedentes	22
CAPITULO 2: Descripción del sistema	23
2.1. Aspectos físicos y geométricos.....	23
2.2. Componentes del Sistema.....	24
2.3. Hardware electromecánico	25
2.3.1. Motor CC	25
2.3.2. Final de carrera	26
2.3.3. Encoder	26
2.4. Hardware electrónico.....	27
2.4.1. Puente H	27
2.4.2. STM32 Blue Pill	28

2.5. Circuito electrónico	30
2.6. Hardware mecánico.....	31
2.6.1. Ensamblaje carro.....	32
2.6.2. Ensamblaje brazo de péndulo.....	34
2.6.3. Ensamblaje de varillas soportes	37
2.6.4. Rodamientos lineales	41
2.6.5. Correa	42
2.6.6. Poleas	44
2.6.7. Engranajes	46
2.6.8. Armado de dispositivo	47
CAPITULO 3: Modelado matemático de la planta	52
3.1. Función de transferencia	55
3.2. Simulación planta lineal	57
3.3. Simulación en planta no lineal	59
3.4. Discretización de la función de transferencia.....	60
CAPITULO 4: Diseño del controlador	62
4.1. Ecuación en diferencia finita.....	64
4.2. Simulación.....	65
4.3. Respuesta a una perturbación de 1°	67
CAPITULO 5: Configuración stm32	70
5.1. Código de STM32.....	74
CAPITULO 6: Implementación	79
CAPITULO 7: Conclusión.....	86
7.1. Trabajos a futuro	86
CAPITULO 8: Bibliografía.....	87

CAPITULO 9: Anexos	89
9.1. Anexo A – Código implementado en stm32	89
9.2. Anexo B – Código de Python para comunicación serial.....	100
9.3. Anexo C – Planos componentes.....	102

CAPITULO 1: Propuesta

1.1. Introducción

En el péndulo invertido lineal, a diferencia de un péndulo tradicional, donde el peso oscila libremente por debajo del punto de pivote, la masa se encuentra balanceándose sobre este punto. Este equilibrio inestable lo convierte en un sistema ideal para el estudio del control y la estabilización, con amplias aplicaciones en la ciencia, tecnología y educación.

El modelo físico se basa en la interacción entre la gravedad, la inercia de la masa y las fuerzas externas que actúan sobre el sistema. Su comportamiento se describe mediante ecuaciones diferenciales que capturan la dinámica no lineal del movimiento. A pesar de la complejidad inherente, el modelo permite analizar y comprender los principios fundamentales del control, la retroalimentación y la respuesta ante perturbaciones.

Las aplicaciones del péndulo invertido lineal son diversas y se extienden a diferentes campos:

- **Ciencia e ingeniería:** Se utiliza como plataforma para investigar y desarrollar algoritmos de control, probar teorías de control automático y experimentar con diferentes estrategias de estabilización. Su naturaleza inestable lo convierte en un banco de pruebas ideal para el desarrollo de controladores robustos y adaptativos.
- **Tecnología:** Los principios de control se aplican en robots bípedos, vehículos auto balanceados, sistemas de estabilización de imagen y plataformas de vuelo, entre otros. La capacidad de mantener el equilibrio en un sistema inestable es esencial para el desarrollo de tecnologías avanzadas.
- **Educación:** Es una herramienta educativa valiosa para ilustrar conceptos de física, control y matemáticas. Permite a los estudiantes experimentar con la dinámica de sistemas no lineales, comprender la importancia de la retroalimentación y desarrollar habilidades de diseño y control.

En las siguientes secciones de este informe, se profundizará en el modelo físico del péndulo invertido lineal, se analizarán las técnicas de control utilizadas para su estabilización y se explorarán con mayor detalle sus diversas aplicaciones.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollo de mecanismo didáctico de péndulo invertido con correcciones de ángulo lineal para la utilización en asignaturas de control y mecánica.

1.2.2. Objetivo específico

- Construir el prototipo.
- Diseñar del controlador más adecuado.
- Implementar el controlador con el modelo mecánico construido.
- Ensayar del dispositivo.

1.3. Alcances y limitaciones

Este trabajo tuvo como alcance el diseño, control e implementación de la estructura esencial para el funcionamiento del péndulo invertido. Se limitó a los dispositivos adquiridos mediante donaciones y al reacondicionamiento de estos. Adicionalmente, se diseñó un algoritmo capaz de controlar el ángulo del péndulo cuando este se encuentra en un ángulo cercano a 0° , sin considerar el control en estado de reposo a 180° . No se desarrolló una interfaz gráfica; la visualización se realizó mediante un monitor serial en el software Visual Studio Code.

1.4. Marco teórico

1.4.1. Clasificación de péndulo invertido

Lineal: Es un sistema dinámico que consta de una vara o péndulo que puede oscilar en un plano vertical alrededor de un punto fijo ilustrado en la Figura N° 1. Este sistema es un modelo

clásico en la teoría de control y la dinámica de sistemas y se utiliza comúnmente como una plataforma para enseñar y estudiar conceptos fundamentales de control automático [1].

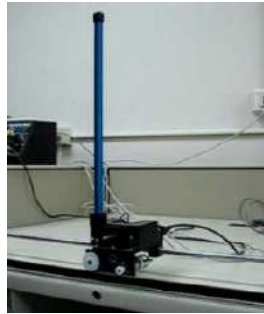


Figura N° 1: Péndulo invertido lineal

Fuente: tomado de [2]

Furuta: Es una variante avanzada que se utiliza en el campo de la teoría de control y la ingeniería robótica. Este sistema como se muestra en la Figura N° 2 es un modelo dinámico que combina un péndulo invertido con una base rotativa.

Consta de dos partes principales: un péndulo invertido y una base rotativa. El primero es una vara que puede oscilar en un plano vertical, similar al péndulo simple. La base rotativa permite que la varilla gire alrededor de un eje horizontal, lo que añade una dimensión rotacional al sistema [3].



Figura N° 2: Péndulo invertido furuta

Fuente: tomado de [4]

Rueda de inercia: Otra variante interesante que incorpora una rueda de inercia como parte integral del sistema. Este diseño mostrado en la Figura N° 3 agrega un componente

rotacional adicional, lo que afecta su dinámica y presenta desafíos únicos en términos de control y estabilización.

Este sistema consta de una vara que puede oscilar en un plano vertical y una rueda de inercia conectada a la misma. La rueda es una masa circular que puede girar alrededor de su eje, agregando un componente rotacional al sistema.

La rotación de la rueda de inercia afecta la distribución de la masa y, por lo tanto, la dinámica del péndulo invertido. Los cambios en la velocidad angular de la rueda de inercia pueden influir significativamente en la estabilidad y el comportamiento del sistema [5].



Figura N° 3: Péndulo invertido de rueda de inercia

Fuente: tomado de [6]

1.4.2. Definición y descripción del péndulo invertido

El péndulo invertido es un sistema dinámico que se utiliza comúnmente en ingeniería y control automático como un modelo desafiante y representativo de sistemas no lineales y sub actuados. Este sistema consiste en una varilla que puede oscilar libremente en un plano vertical y un carro desplazable horizontalmente (la “base”) las cuales están unidas a través de una articulación. El objetivo es mantener tal varilla en posición vertical, lo que implica la necesidad de controlar tanto el movimiento horizontal del carro.

La dinámica del sistema es altamente no lineal debido a la presencia de términos trigonométricos en las ecuaciones de movimiento. Estas ecuaciones describen cómo cambian las

posiciones, velocidades del carro y del péndulo en función del tiempo, y están influenciadas por factores como la gravedad, la longitud, la masa y la fricción. En su forma más simple, el modelo matemático se puede expresar mediante ecuaciones diferenciales.

La principal característica es su subactuación, lo que significa que hay menos grados de libertad de control disponibles que variables de estado que deben controlarse. En otras palabras, el sistema es inherentemente inestable y tiende a caerse si no se aplica un control adecuado.

El dispositivo se utiliza ampliamente como una plataforma de investigación y enseñanza en ingeniería de control debido a su complejidad y su capacidad para ilustrar conceptos de control, como la realimentación, la estabilidad y la respuesta transitoria. Su estudio es fundamental para desarrollar estrategias de control efectivas en sistemas dinámicos reales, como robots autónomos y sistemas de estabilización [7].

1.4.3. Importancia y relevancia del estudio del péndulo invertido en ingeniería

1. Ilustra conceptos de control complejos: El péndulo invertido es un sistema dinámico inherentemente inestable y no lineal. Su estudio permite a los ingenieros y científicos de control comprender conceptos avanzados de control, como la realimentación, la estabilidad, la linealización y la controlabilidad. Abordar un sistema tan desafiante proporciona una base sólida para abordar problemas de control más complejos en una amplia gama de aplicaciones.
2. Aplicación en robótica y automatización: Muchos sistemas robóticos y sistemas de automatización se asemejan al péndulo invertido en términos de subactuación y dinámica no lineal. El conocimiento adquirido al estudiar este dispositivo se puede aplicar directamente en el diseño y control de robots móviles, drones, brazos robóticos y otros dispositivos automatizados.
3. Educativo: El péndulo invertido es una herramienta educativa efectiva para enseñar a estudiantes y profesionales los principios del control automático. Es un sistema tangible y visualmente intuitivo que permite a los aprendices experimentar y comprender los conceptos de control en la práctica.
4. Investigación en control avanzado: El estudio del péndulo invertido sirve como plataforma de investigación para desarrollar y probar algoritmos de control avanzado. Estos algoritmos pueden incluir controladores PID, control óptimo,

control por retroalimentación de estados, control no lineal y técnicas de aprendizaje automático. Los avances en el control del péndulo invertido tienen aplicaciones más amplias en la estabilización de sistemas dinámicos complejos.

5. Control de sistemas subactuados: Los sistemas subactuados son comunes en la vida real, y su control efectivo es esencial en aplicaciones que van desde la industria aeroespacial hasta la robótica doméstica. Comprender cómo controlar sistemas subactuados es fundamental para mejorar la seguridad y el rendimiento en estas aplicaciones.
6. Aplicaciones prácticas: El control de un péndulo invertido tiene aplicaciones prácticas en la estabilización de objetos en movimiento, la suspensión de cargas en grúas, la navegación de vehículos autónomos, la corrección de postura en dispositivos portátiles, como los dispositivos de asistencia médica y el aterrizaje de módulos de cohetes espaciales.

1.4.4. Dinámica del péndulo invertido

El péndulo como se muestra en la Figura N° 4 cuenta con un brazo que puede oscilar libremente en un plano vertical alrededor de un punto de pivote. La longitud de este brazo se conoce como la longitud del péndulo (l), y su masa (m) contribuye significativamente a la dinámica del sistema. El ángulo de inclinación (θ) del péndulo con respecto a la vertical es una variable clave que determina su posición y movimiento.

La base o carro es la plataforma horizontal sobre la cual se encuentra montado el péndulo, y puede moverse horizontalmente a lo largo de un riel u otra superficie. La posición horizontal del carro se denota como (x), y su masa (M) también es un factor importante en la dinámica del sistema. La fuerza aplicada al carro (u) es un control que permite mover el carro y, por ende, afecta la posición y estabilidad del péndulo [8].

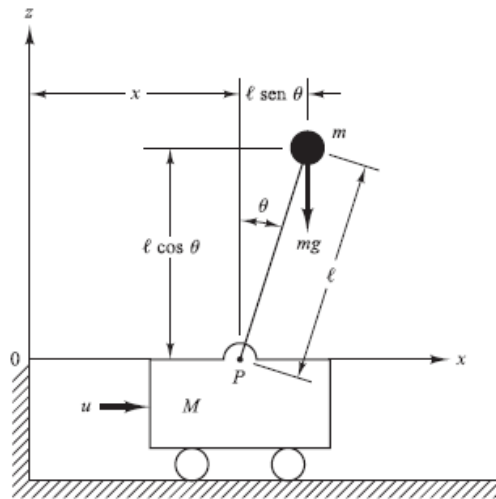


Figura N° 4: Sistema de péndulo invertido

Fuente: tomado de [9]

1.4.5. Equilibrio

El concepto de equilibrio en el contexto del péndulo invertido se refiere a un estado en el cual el sistema se encuentra en reposo y no experimenta cambios en sus variables relevantes con el tiempo. En otras palabras, cuando está en equilibrio, no hay aceleración ni movimiento neto en la posición angular (θ) ni en la posición horizontal (x) del carro que lo sostiene. Esto implica que el sistema se encuentra en una configuración estable donde todas las fuerzas y momentos que actúan sobre él se compensan mutuamente, lo que resulta en un estado de quietud relativa.

Para que el mecanismo esté en equilibrio, deben cumplirse ciertas condiciones específicas. Estas se relacionan con las fuerzas y momentos que actúan sobre el sistema, y varían según las características particulares del péndulo y las fuerzas externas aplicadas, como la gravedad y las fuerzas de control. En general, el equilibrio se logra cuando las fuerzas y momentos que actúan sobre la varilla y el carro se equilibran de manera que no haya tendencia a acelerar o desplazar el sistema en ninguna dirección.

1.4.5.1. Condiciones para el equilibrio estático

El equilibrio estático del sistema implica que este se encuentra en reposo, sin ningún tipo de movimiento ni aceleración. Para lograr este estado, deben cumplirse condiciones precisas en relación con las posiciones angular (θ) y horizontal (x). Estas condiciones pueden variar según las características particulares del sistema, pero en términos generales, las condiciones típicas para el equilibrio estático incluyen:

- **Ángulo de inclinación (θ):** La posición angular debe ser constante y no debe experimentar cambios con el tiempo. Esto significa que la derivada temporal del ángulo ($d\theta/dt$) debe ser igual a cero en el punto de equilibrio. En otras palabras, el péndulo no se está moviendo hacia adelante ni hacia atrás, y su ángulo de inclinación permanece constante.
- **Posición horizontal (x):** La posición horizontal del carro también debe ser constante. Al igual que con el ángulo, la derivada temporal de la posición horizontal (dx/dt) debe ser igual a cero en el punto de equilibrio. Esto significa que el carro no se está moviendo hacia la izquierda ni hacia la derecha, y su posición permanece constante.
- **Equilibrio de fuerzas y momentos:** Además de las condiciones relacionadas con las posiciones angular y horizontal, es esencial que todas las fuerzas y momentos que actúan sobre el sistema estén equilibrados en el punto de equilibrio. Esto implica que las fuerzas externas, como la gravedad y las fuerzas de control, deben compensarse de manera que no haya una fuerza neta actuando en ninguna dirección y que no haya un momento neto que cause una tendencia a girar el sistema.

1.4.5.2. Tipos de equilibrio

Equilibrio estable: Describe en detalle lo que sucede cuando el sistema recibe una perturbación y este vuelve a su estado estable luego de un tiempo. Esto implica que, si la varilla se desvía ligeramente de la posición vertical y se libera, eventualmente regresará a esa posición vertical sin alejarse aún más.

Equilibrio inestable: Contrasta el concepto de estabilidad con la inestabilidad. Explica que un punto de equilibrio es inestable si las perturbaciones lejos de ese punto tienden a alejar

aún más al sistema de dicho equilibrio. En este caso, esto podría significar que una pequeña inclinación resultaría en el péndulo cayendo completamente.

Equilibrio neutral: Refiere a una condición en la que el sistema, al sufrir una perturbación, no experimenta un cambio significativo hacia la estabilidad o la inestabilidad. En este estado, el sistema mantiene su posición desviada sin mostrar tendencia a retornar a su posición original ni a alejarse más de ella. Es una situación en la que el sistema no corrige su desviación ni la amplifica, permaneciendo en un estado intermedio y sin una respuesta definida frente a las perturbaciones externas [10].

1.4.6. Sistemas de control

Los sistemas de control son estructuras diseñadas para gestionar y regular el comportamiento de un sistema dinámico. Su función principal es ajustar las entradas al sistema para asegurar que la salida o el comportamiento del sistema se mantenga en un rango deseado o se ajuste a un objetivo específico. Estos sistemas emplean modelos matemáticos y algoritmos para prever y controlar las respuestas del sistema, con el fin de alcanzar una operación estable y eficiente. Son fundamentales en diversas aplicaciones tecnológicas e industriales, donde se busca automatizar y optimizar procesos para mejorar el rendimiento y la precisión por medio de lazos de control que pueden ser de lazo abierto o cerrados.

1.4.6.1. Lazo abierto

Los sistemas de control a lazo abierto son aquellos en los que la acción de control se aplica sin tener en cuenta el estado actual del sistema o su salida. En estos sistemas, la entrada o el comando se envía al proceso sin retroalimentación sobre el resultado de esa acción.

La principal característica de un sistema de control a lazo abierto es que la salida del sistema no se mide ni se utiliza para ajustar el comportamiento del controlador. Esto puede ser adecuado en situaciones donde el proceso es simple, predecible y no requiere corrección constante. La simplicidad y menor costo de implementación son ventajas de los sistemas a lazo abierto, pero pueden ser menos eficaces en entornos donde se presentan variaciones o perturbaciones que afectan el proceso [9].

1.4.6.2. Lazo cerrado

Los sistemas de control a lazo cerrado, a diferencia de los de lazo abierto, utilizan la retroalimentación para ajustar continuamente la acción de control en función del estado actual del sistema. En estos, se mide la salida del proceso y se compara con una referencia o valor deseado. La diferencia entre la salida medida y la referencia, conocida como error, se utiliza para ajustar las entradas de manera que el error se minimice y el sistema se mantenga dentro de los límites deseados.

La retroalimentación permite que el sistema responda a perturbaciones o variaciones en el entorno, adaptándose en tiempo real para mantener un rendimiento constante. Esto es especialmente útil en procesos complejos o variables, donde el control preciso es esencial para garantizar la estabilidad y eficiencia del sistema. Los sistemas de control a lazo cerrado ofrecen mayor precisión y adaptabilidad, aunque pueden ser más complejos y costosos de implementar en comparación con los de lazo abierto [9].

1.4.6.3. Lugar Geométrico de las Raíces

El lugar geométrico de las raíces (LGR) es una técnica gráfica fundamental en el análisis y diseño de sistemas de control, utilizada para explorar cómo varían las raíces del polinomio característico en función de un parámetro de ganancia. Las raíces, también conocidas como polos, juegan un papel crucial en la determinación de la estabilidad y del comportamiento dinámico de un sistema, influenciando aspectos como la respuesta transitoria y la estabilidad a largo plazo. El LGR traza la trayectoria de estos polos en el plano complejo conforme se ajusta la ganancia, permitiendo observar cómo se modifican las características dinámicas del sistema.

El uso del LGR es particularmente valioso en la ingeniería de control, ya que proporciona una representación visual clara de cómo los cambios en los parámetros de un controlador impactan la estabilidad y el rendimiento global. A través de esta técnica, los ingenieros pueden evaluar la viabilidad de diferentes ajustes y diseñar controladores que aseguren un desempeño óptimo y un comportamiento estable. Además, este facilita la comprensión de la relación entre la ganancia y la respuesta del sistema, lo cual es esencial para la optimización y el ajuste fino de los controladores. De esta manera, se convierte en una herramienta esencial para lograr un diseño de control efectivo y robusto, alineado con los requisitos del sistema en estudio [9].

1.4.6.4. Sistemas discretos

Los sistemas discretos son aquellos en los que las señales y los datos se procesan en intervalos de tiempo discretos, en lugar de ser continuos. Estos operan sobre secuencias de datos que se obtienen muestreando señales continuas o generando señales de forma digital.

En estos sistemas, el tiempo se divide en intervalos finitos, y las señales se representan como secuencias de valores en esos intervalos. Este enfoque es fundamental en el procesamiento digital de señales, control y comunicaciones digitales. Además, estos suelen analizarse utilizando herramientas matemáticas como la transformada Z y el análisis en el dominio de la frecuencia discreta.

El análisis y diseño involucra la aplicación de técnicas que permiten modelar y controlar procesos que se actualizan en pasos de tiempo específicos. Son especialmente útiles en aplicaciones que requieren procesamiento en tiempo real y alta precisión, como en el procesamiento de audio, video y datos en sistemas embebidos [9].

1.4.6.5. Transformada z

La transformada Z es una herramienta matemática utilizada en el análisis y diseño de sistemas discretos, que convierte secuencias temporales discretas en una representación en el dominio complejo. Es fundamental en el procesamiento digital de señales y en el control digital.

En el análisis de sistemas discretos es usada porque permite determinar la estabilidad y el comportamiento en frecuencia del sistema al examinar los polos y ceros de la función de transferencia en el plano Z . Además, facilita la implementación de algoritmos de control y filtrado digital, ya que las operaciones en el dominio Z a menudo son más manejables y directas que en el dominio del tiempo [11].

1.4.6.6. Lugar Geométrico de las Raíces en el plano Z

El lugar geométrico de las raíces en el dominio Z es una extensión del concepto de lugar geométrico de las raíces utilizado en el dominio continuo, pero aplicado a sistemas discretos. En el dominio Z , se estudia cómo las raíces del polinomio característico de un sistema discreto cambian en función de un parámetro de ganancia a medida que varía.

En un sistema discreto, el polinomio característico está dado por la función de transferencia del sistema en el dominio Z . El lugar geométrico de las raíces en este contexto muestra las trayectorias que siguen los polos del sistema en el plano complejo Z cuando el parámetro de ganancia varía.

Este análisis proporciona información crucial sobre la estabilidad y el comportamiento dinámico del sistema discreto. En particular, se puede observar cómo la ubicación de los polos afecta la respuesta transitoria y la estabilidad del sistema. La estabilidad de un sistema discreto está relacionada con la ubicación de los polos en el plano Z : si todos los polos están dentro del círculo unitario $|Z| < 1$, el sistema es estable; si alguno está fuera de este círculo, el sistema es inestable.

El lugar geométrico de las raíces en el dominio Z es una herramienta útil para diseñar y ajustar controladores digitales, ya que permite visualizar cómo los cambios en el parámetro de ganancia afectan la estabilidad y el desempeño del sistema [11].

1.4.6.7. Retenedor de orden cero

Un retenedor de orden cero es un circuito o dispositivo que mantiene el valor de una señal digital constante durante el intervalo de muestreo hasta que llega un nuevo valor de muestra. Esto significa que el valor de la señal se "sostiene" o se mantiene fijo durante el período de tiempo entre las actualizaciones, creando una señal continua a partir de la secuencia discreta de valores digitales. El retenedor de orden cero se usa comúnmente en la conversión entre sistemas digitales y analógicos.

Matemáticamente, un retenedor de orden cero se puede representar como una función de transferencia que tiene un comportamiento similar a un "muestreador" que no introduce ningún cambio en la señal durante el período de retención. Su función de transferencia es una función constante en el dominio Z , y en el dominio del tiempo, se comporta como un sistema que almacena y mantiene el valor de la señal de entrada.

En el análisis y diseño de sistemas discretos, el retenedor de orden cero ayuda a simplificar la representación de sistemas y facilita la implementación de algoritmos de control digital, ya que se encarga de mantener el valor de la señal entre los instantes de muestreo [11].

1.5. Justificación

Este proyecto tiene la particularidad de que su principio físico de funcionamiento sirve de base para muchos mecanismos importantes y en desarrollo en la actualidad. Entre ellos se encuentra lo que actualmente está en periodo de pruebas en SpaceX los aterrizajes de los propulsores de los cohetes, como lo muestra la Figura N° 5, evitando desperdicios económicos una vez cumplieron su función. Otra de las aplicaciones es para el sistema de estabilización de robots ilustrado en la Figura N° 6.



Figura N° 5: Aterrizaje de cohete

Fuente: tomado de [12]

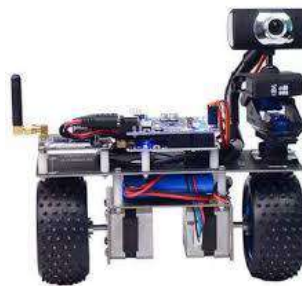


Figura N° 6: Robot con sistema de estabilización

Fuente: tomado de [13]

Con lo dicho anteriormente se quiere destacar que, a pesar de ser un sistema relativamente simple este puede ser adaptado y se puede usar para el control de otros mecanismos más complejos como los antes mencionados.

De aquí se desprende que este es un proyecto multidisciplinario, donde además de abordar tópicos de sistemas de control, se trabaja otros aspectos como:

- **Mecánica:** Al tratarse de un sistema el cual posee elementos que realizan un movimiento por medio de una fuerza, se debe diseñar un mecanismo el cual permita el correcto movimiento para la corrección del ángulo en cuestión.
- **Electrónica:** Aquí se diseña e implementa un circuito electrónico el cual permita la medición del ángulo de desviación y el control del motor que corrige dicha desviación.
- **Computación:** Se debe programar un microcontrolador el cual va a integrar todo el código del controlador seleccionado para que el sistema funcione de acuerdo con lo esperado.

De esta manera se estaría abordando todas las ramas que tiene la mecatrónica lo que permite integrar todos los conocimientos adquiridos a lo largo de la carrera.

1.6. Antecedentes

Hoy en día existen varios prototipos funcionales de péndulos entre ellos está el de tipo furuta desarrollado en la Universidad Politécnica de Valencia por A. Valera, M. Valles y M. Cardo [2]. Otro de los que se tuvo referencia es el de tipo rueda de inercia desarrollado también la misma universidad, pero por Iván Augusto Duarte Olivares [3].

Por otro lado, el tipo de péndulo desarrollado es en muchas universidades (uno de ellos desarrollado por Fernando Castaños Luna [4]) implementado como ejemplo de controles a sistemas mecánicos, pero no se aborda el ámbito didáctico al que se quiere llegar y no se realiza tampoco un estudio de los mecanismos del prototipo.

CAPITULO 2: Descripción del sistema

El sistema mostrado en la Figura N° 7 consiste en un carro que puede moverse horizontalmente a lo largo de un riel. Desde el carro, un brazo rígido está conectado a un eje de pivote, permitiendo que el brazo oscile libremente. La gravedad tiende a hacer que el péndulo caiga, por lo que el sistema es inherentemente inestable. El equilibrio del sistema se logra mediante movimientos controlados del carro, contrarrestando las tendencias gravitatorias que quieren llevar al péndulo a una posición estable (colgado hacia abajo).

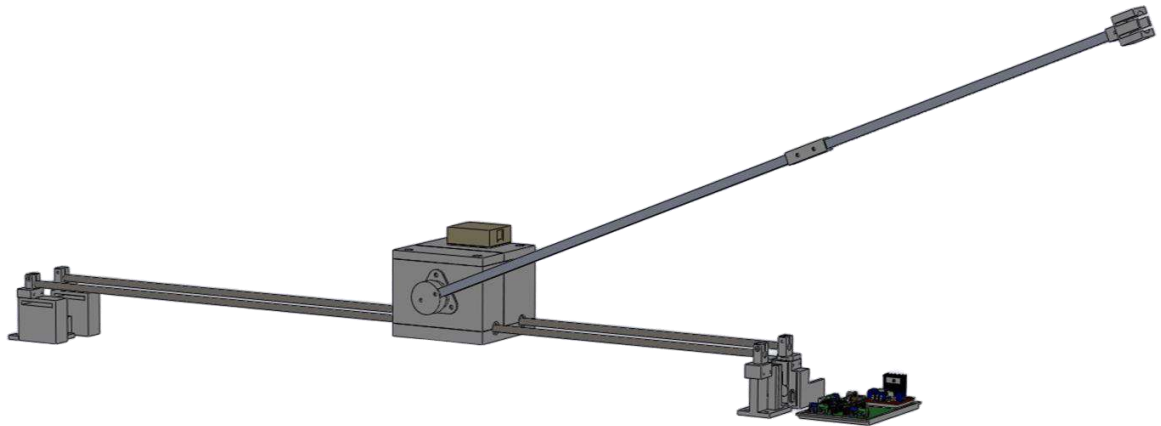


Figura N° 7: Péndulo invertido

2.1. Aspectos físicos y geométricos

El péndulo invertido tiene dos grados de libertad principales:

- Posición horizontal del carro (x): el desplazamiento del carro a lo largo del riel.
- Ángulo del péndulo (θ): la inclinación del brazo respecto a la vertical.

En equilibrio, el ángulo del péndulo es cero, es decir, el péndulo está completamente vertical. Cualquier pequeña desviación de esta posición tiende a aumentar por la acción de la gravedad, haciendo que el sistema sea inestable. El movimiento horizontal del carro es crucial para compensar esta tendencia y mantener el equilibrio del péndulo.

2.2. Componentes del Sistema

En este proyecto se utiliza varios componentes esenciales que permiten el funcionamiento y control de un sistema de péndulo invertido. A continuación, se enumeran junto con una breve descripción de su función principal:

1. Motor CC: Componente clave en el control del ángulo del péndulo, al girar mueve una polea conectada a la correa.
2. Puente H: Circuito electrónico que controla la dirección de giro del motor CC.
3. Encoder: Dispositivo que convierte el movimiento físico del carro en una señal eléctrica, proporcionando información sobre su ángulo, velocidad angular, aceleración angular y sentido de giro.
4. STM32 Blue Pill: Placa de desarrollo basada en el microcontrolador STM32F103C8T6, utilizada para la adquisición de datos y la salida de señales necesarias para controlar el péndulo invertido.
5. Carro: Es el elemento móvil que se desplaza a lo largo de un eje lineal, impulsado por un motor para ejercer una fuerza controlada.
6. Brazo del péndulo: Una barra rígida conectada al carro que puede rotar libremente alrededor de un eje horizontal.
7. Eje de pivote: El punto de conexión entre el carro y el brazo del péndulo, permitiendo la oscilación del brazo.
8. Varillas soportes: Mantienen el carro en el aire y guían su movimiento lineal.
9. Rodamientos lineales: Diseñados para soportar y facilitar el movimiento lineal del carro.
10. Soporte de varillas: Mantiene las varillas paralelas y a una distancia fija entre ellas.
11. Correa: Conectada al carro, permite su movimiento lineal y el control del ángulo mediante su tracción.

12. Poleas: Facilitan el movimiento de la correa.

Estos componentes, trabajando en conjunto, permitirán construir y controlar un sistema de péndulo invertido eficazmente, con un movimiento preciso y una estabilidad adecuada.

2.3. Hardware electromecánico

2.3.1. Motor CC

Un motor de corriente continua, ilustrado en la Figura N° 8, es un dispositivo electromecánico que convierte la energía eléctrica en energía mecánica mediante la interacción de campos magnéticos.

A fines del proyecto se van a tener en cuenta dos variables a controlar para llevar a cabo el control:

- Fuerza del rotor: A través de la polea dentada que está solidaria al eje del motor se puede controlar la fuerza que la correa ejerce sobre el carro y este sobre el péndulo. De esta manera, con el puente H y por medio de una salida PWM se puede controlar la fuerza a la que está sometido el péndulo para la corrección del ángulo.
- Dirección de rotación: La dirección de rotación del motor de CC puede invertirse invirtiendo la polaridad de la tensión de alimentación. Esto cambia la dirección del flujo de corriente en los devanados del rotor, invirtiendo el sentido del par motor.



Figura N° 8: Motor CC
Fuente: tomado de [14]

2.3.2. Final de carrera

Como lo muestra la Figura N° 9 los finales de carrera son dispositivos diseñados para establecer un límite en un mecanismo. Estos cuentan con un pulsador que se activa cuando el elemento controlado alcanza el final de su recorrido, emitiendo una señal eléctrica que indica la llegada al extremo. En este proyecto, se utilizarán para señalar cuando el carro del péndulo alcanza el final de las varillas, evitando que el sistema se dañe al continuar aplicando fuerza, dado que no se dispone de un control de posición.



Figura N° 9: Final de carrera

2.3.3. Encoder

Un encoder es un dispositivo electromecánico que convierte el movimiento físico de un objeto en una señal eléctrica que puede ser interpretada por un sistema de control. Este dispositivo es esencial para obtener información precisa sobre la posición, velocidad y dirección de un objeto en movimiento, y se utiliza ampliamente en aplicaciones de control de movimiento y automatización.

Se logró reacondicionar un Encoder Eltra ER63, ilustrado en la Figura N° 10, el cual estaba fuera de servicio debido a un daño en la ficha de conexión.



Figura N° 10: Encoder Eltra ER63

Fuente: tomado de [15]

Los datos técnicos son:

- Diámetro exterior: 63 mm
- Diámetro eje (máx.): 10 mm
- Resolución (máx.): 24.000 ppr
- Frecuencia de salida (máx.): 300 kHz
- Temperatura operativa (°C): -20° ... +70°
- Grado protección: IP 66

2.4. Hardware electrónico

2.4.1. Puente H

Un puente H es un circuito electrónico que permite controlar la dirección de giro de un motor de corriente continua (CC). Es fundamental en aplicaciones donde se requiere inversión de la dirección del motor, como en robots y sistemas de control de movimiento.

Para controlar el puente H se utiliza un PWM (Modulación por Ancho de Pulso), el cual al variar el ciclo de trabajo de la señal PWM aplicada a los transistores, se puede controlar la cantidad de potencia entregada al motor, ajustando así su velocidad.

Puente H – L298n

Las características del puente H, de la Figura N° 11, utilizado son las siguientes:

- Doble puente H
- Chip L298N (ST)
- Voltaje lógico: 5V
- Tensión de salida: 5V-35V
- Corriente de control: 0mA – 36mA
- Corriente máxima de salida: 2^a (puente simple)
- Potencia máxima de 25W
- Tamaño 43X43X27mm



Figura N° 11: Puente H – L298n

Fuente: tomado de [16]

2.4.2. STM32 Blue Pill

El STM32F103C8T6 mostrado en la Figura N° 12 [17] es un microcontrolador basado en la arquitectura ARM Cortex-M3 fabricado por STMicroelectronics. Es ampliamente utilizado en aplicaciones de desarrollo y prototipado debido a su bajo costo, alto rendimiento y amplia gama de funcionalidades.

El microcontrolador STM32 es extremadamente útil en el desarrollo del control del ángulo de un péndulo invertido debido a sus diversas características y capacidades que se alinean perfectamente con las necesidades de este tipo de sistema de control. A continuación, se detallan las razones específicas por las que este microcontrolador es adecuado para este proyecto:

- Alta velocidad de procesamiento
- Capacidad de manejo de sensores
- Control de actuadores
- Memoria adecuada para algoritmos de control

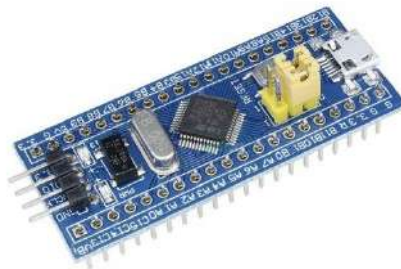


Figura N° 12: STM32F103pC8T6

Fuente: tomado de [18]

Especificaciones:

- ARM 32-bit Cortex-M3 CPU Core
- Corre a una frecuencia de 72 MHz (1.25 DMIPS/MHz)
- 64 Kbytes de memoria Flash
- 20 Kbytes de SRAM
- RTC integrado y entrada de batería de respaldo para el RTC, esto es un beneficio definitivamente ya que permite, por ejemplo, guardar datos con timestamp sin usar placas externas

- Modo Sleep, Stop y Standby
- 26 entradas y salidas digitales, la mayoría tolerantes a 5V
- Interrupciones en todas las I/O
- 2 conversores A/D de 12-bit de 1 us, (10 entradas analógicas)
- 7 temporizadores
- 2 interfaces I2C
- 3 interfaces USART
- 2 interfaces SPI
- Interfaz CAN
- Micro USB para alimentación de la placa y comunicaciones
- Dimensiones: 53 x 22 mm

2.5. Circuito electrónico

La placa electrónica (Anexo C Plano 01-04-03) ilustrada en la Figura N° 13 integra diversos componentes siendo el microcontrolador STM32, ubicado en el centro, el núcleo del sistema. Desde este microcontrolador se distribuyen todas las conexiones. Como se muestra en la Figura N° 14, en el extremo superior derecho de la placa se encuentran las entradas USB, utilizadas para la comunicación con la PC. Estas entradas permiten visualizar datos como el ángulo del péndulo, la acción de control y el estado del programa (encendido o apagado).

Junto a las entradas USB se ubican las señales de control y potencia del encoder, que se transmiten mediante un cable de comunicación UTP. En la esquina superior izquierda de la placa se localiza la alimentación del puente H, que opera a 12V, así como las salidas hacia el motor. Debajo de esta sección, se encuentran los finales de carrera, que previenen el choque del carro contra los soportes de las varillas. Finalmente, en el extremo inferior derecho, está el pulsador de “START”, que activa el funcionamiento del control.

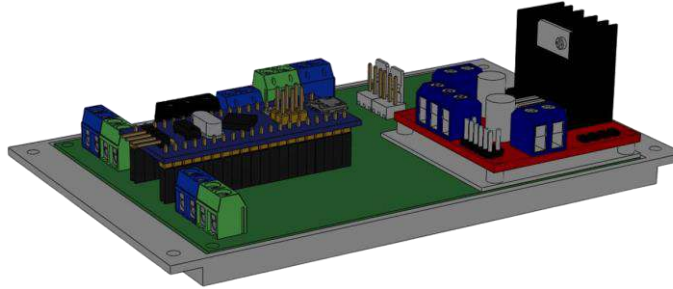


Figura N° 13: Esquema eléctrico de péndulo invertido

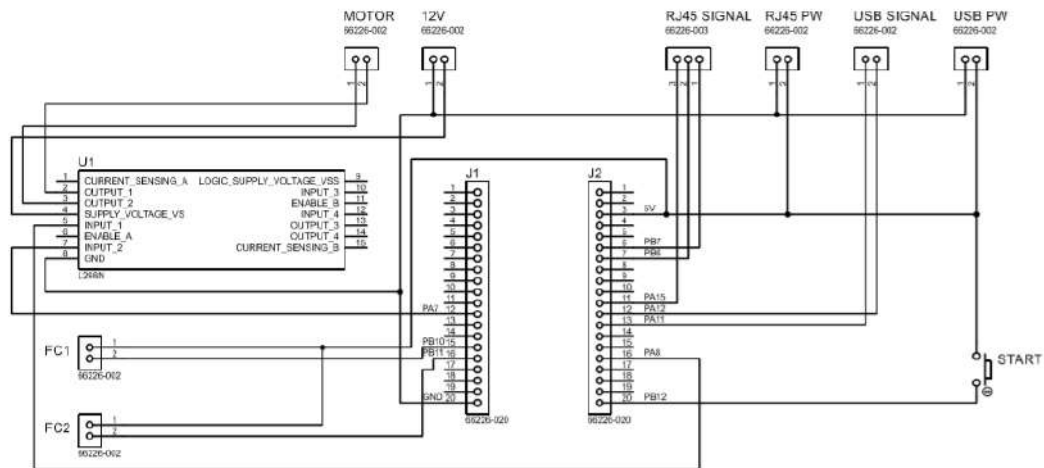


Figura N° 14: Esquema eléctrico de péndulo invertido

2.6. Hardware mecánico

Si se da una descripción detallada de los componentes del ensamblaje de la Figura N° 7 se puede dividir el dispositivo en cuatro partes:

- Ensamblaje carro
- Ensamblaje brazo de péndulo
- Ensamblaje de varillas soportes
- Ensamblaje de placa electrónica

2.6.1. Ensamblaje carro

El carro (Anexo C Plano 01-01-00) es una pieza clave del sistema, encargada de moverse a lo largo de un eje lineal para mantener el equilibrio del péndulo. Equipado con componentes como los rodamientos lineales y el encoder, el carro debe funcionar con alta precisión y rapidez. Su correcto funcionamiento, en coordinación con el microcontrolador, es esencial para lograr un control eficaz y estable del ángulo.

Como muestra la Figura N° 15 el cuerpo está conformada por las siguientes partes:

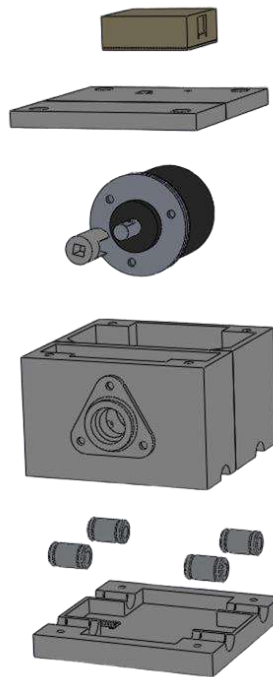


Figura N° 15: Carro

1. **Caja RJ45:** Este elemento es utilizado para la conexión entre la STM32 y el encoder a través de un cable UTP, facilitando la comunicación entre ambos. El enlace se realiza dentro de una caja que protege y organiza los componentes, asegurando que la transmisión de datos sea estable.
2. **Cubierta** (Anexo C Plano 01-01-04/Figura N° 16): Este elemento incluye las entradas para los tornillos que fijarán firmemente las tres partes entre sí. Además,

contará con la salida del cable que transmitirá las señales generadas por el encoder.

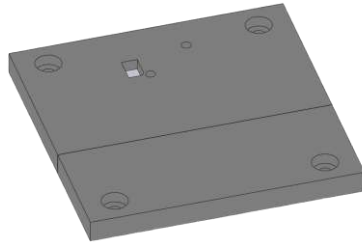


Figura N° 16: Cubierta carro

3. **Encoder Eltra:** Descripto en la sección “2.3.3. Encoder”.
4. **Eje encoder** (Anexo C Plano 01-01-01/Figura N° 17): Esta pieza sirve para acoplar el eje del encoder con el eje del péndulo, garantizando un movimiento sincronizado y preciso entre ambos. Ha sido diseñada cuidadosamente para eliminar cualquier juego entre las partes, lo que es crucial para el correcto funcionamiento del sistema. Su forma es mayormente cilíndrica, aunque cuenta con una superficie plana en uno de sus lados, permitiendo que el eje del encoder gire al unísono con el eje del péndulo.

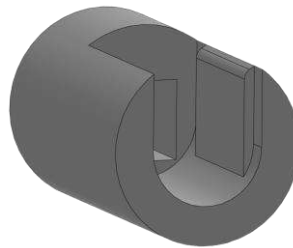


Figura N° 17: Eje encoder

5. **Cuerpo** (Anexo C Plano 01-01-02/Figura N° 18): Esta es la pieza principal, ya que soporta el encoder, como se muestra en la Figura N° 19, y un rodamiento que sostiene el brazo del péndulo. Además, junto con la tapa inferior, sujeta los cuatro rodamientos lineales. Los tres orificios situados alrededor del soporte del rodamiento permiten la inserción de un destornillador para fijar el encoder al carro.

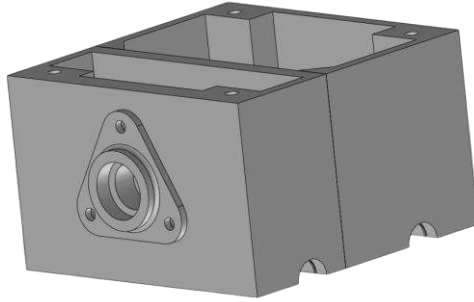


Figura N° 18: Cuerpo carro

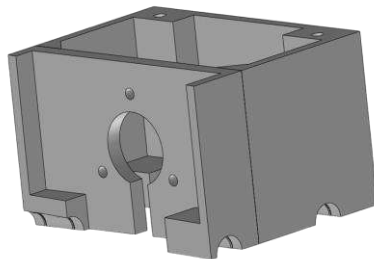


Figura N° 19: Corte de cuerpo carro

6. **Rodamientos lineales LM8UU:** Mencionados en el inciso “2.6.4. Rodamientos lineales”.
7. **Base carro** (Anexo C Plano 01-01-03/Figura N° 20): Su función es asegurar los rodamientos lineales al carrito, además de sujetar la correa, permitiendo así el movimiento del sistema.

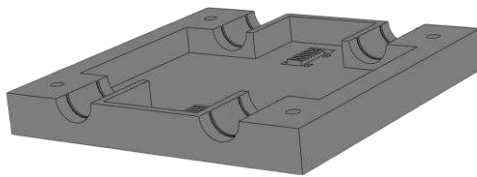


Figura N° 20: Base carro

2.6.2. Ensamblaje brazo de péndulo

El diseño y funcionamiento del brazo del péndulo (Anexo C Plano 01-02-00) mostrado en la Figura N° 21 son determinantes para la estabilidad y respuesta del sistema. Este ensamblaje

de componente proporciona la estructura necesaria para soportar la masa del péndulo y facilitar su movimiento oscilatorio, desempeñando un papel crucial en el control efectivo del sistema. Para lograr un rendimiento óptimo en diversas aplicaciones, es esencial que el diseño sea meticuloso y la construcción robusta.



Figura N° 21: Ensamblaje péndulo

Este ensamblaje cuenta con los siguientes componentes:

1. **Soporte pesas** (Anexo C Plano 01-02-02): Con el objetivo de optimizar la capacidad didáctica del prototipo, se diseñó un soporte ilustrado en la Figura N° 22 que permite ajustar distintas masas para analizar la respuesta del sistema frente a variaciones paramétricas. Este soporte fue concebido para mantener el equilibrio de la carga, evitando una distribución inadecuada del peso que podría desestabilizar el sistema. Así, se asegura que las pruebas reflejen con precisión el comportamiento del sistema bajo diferentes condiciones de carga.

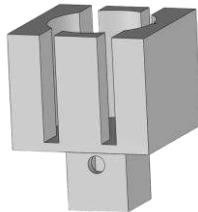


Figura N° 22: Soporte pesas

2. **Varillas** (Anexo C Plano 01-02-04/Figura N° 23): Las varillas están conectadas para incrementar la longitud y permitir variaciones paramétricas que enriquecen la didáctica del sistema. Esta conexión se realiza mediante dos uniones atornilladas: Una interior (Anexo C Plano 01-02-03/Figura N° 24) y otra exterior (Anexo C Plano 01-02-03/Figura N° 25). Esta configuración proporciona una unión rígida, eliminando el juego entre las mismas.



Figura N° 23: Varilla



Figura N° 24: Unión exterior



Figura N° 25: Unión interior

3. **Eje péndulo** (Anexo C Plano 01-02-01/Figura N° 26): El eje del péndulo actúa como un punto de conexión crucial entre el brazo del péndulo y la estructura base. El eje incluye dos orificios destinados para la fijación de las varillas. Un orificio está destinado a la principal, mientras que el otro permite la incorporación de una adicional para modificar la función de transferencia mediante la adición de un contrapeso, facilitando así la adaptación del sistema con fines didácticos en el desarrollo del control. Además, el eje está diseñado para apoyarse sobre un rodamiento y cuenta con una adaptación solidaria para acoplarse al encoder mediante el adaptador mostrado en la Figura N° 17.

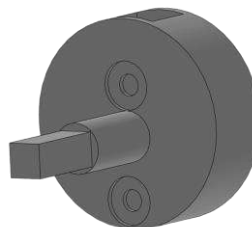


Figura N° 26: Eje péndulo

2.6.3. Ensamblaje de varillas soportes

Este ensamblaje (Anexo C Plano 01-03-00/Figura N° 27) tiene como función sostener el carro y permitir el movimiento tratando de evitar el menor esfuerzo del motor para moverlo. Está compuesto por soportes que están diseñados específicamente para fijar y mantener las varillas en una posición y orientación determinadas, proporcionando un punto de apoyo sólido que garantiza su alineación y posición relativa durante todo el funcionamiento del sistema. Además, contribuyen a la estabilidad estructural al distribuir uniformemente las cargas y fuerzas a lo largo de las varillas, fortaleciendo la estructura del sistema y minimizando la posibilidad de deformación o flexión bajo cargas dinámicas. Asimismo, los soportes de varillas desempeñan un papel importante en la minimización de vibraciones y resonancias no deseadas, absorbiendo y disipando las vibraciones que puedan surgir durante el funcionamiento del sistema y reduciendo la posibilidad de resonancias que podrían afectar la estabilidad y precisión. En conjunto, estos componentes contribuyen a un funcionamiento suave, estable y confiable del conjunto. Para ser más específico el ensamblaje tiene los siguientes componentes:

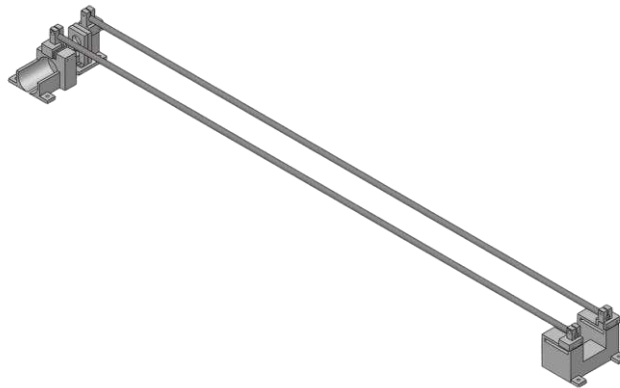


Figura N° 27: Ensamblaje soporte de varillas

1. **Soporte motor** (Anexo C Plano 01-03-10/Figura N° 28): Este contendrá las partes más importantes del accionamiento del mecanismo, como lo son:

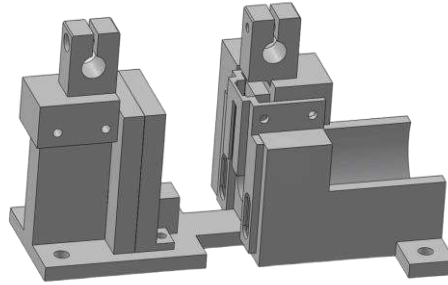


Figura N° 28: Soporte de varilla motor

- **Motor:** El motor será instalado en la parte lateral derecha del soporte, y contará con tres orificios en su extremo. Dos de estos orificios están destinados a la sujeción mediante tornillos, mientras que el tercero permitirá la salida del eje del motor.
- **Sujeción de polea motora** (Anexo C Plano 01-03-14): La polea motora estará montada sobre una rueda dentada, la cual se sostiene mediante un soporte, como se muestra en la Figura N° 29, que incluye un tornillo para ajustar la altura. A través de dos rodamientos, se evitará que la polea soporte la carga de la correa, lo que reduce el estrés en el sistema. Uno de los rodamientos estará soportado por dos estructuras que evitan el movimiento. La primera (Anexo C Plano 01-03-15/Figura N° 30) cuenta con un orificio para alojar el rodamiento, mientras que la segunda (Anexo C Plano 01-03-15/Figura N° 31) tiene la función de complementar la primera.

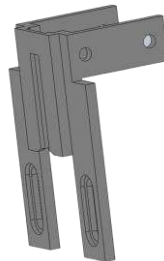


Figura N° 29: Soporte polea motora

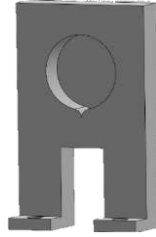


Figura N° 30: Soporte rodamiento de polea motora

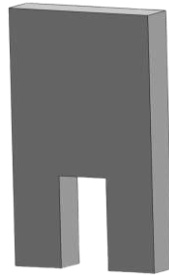


Figura N° 31: Suplemento

- **SK8:** Descripto en el punto 4.
 - **Orejas de sujeción:** Las orejas de sujeción servirán como anclas del sistema a una superficie específica, aseguradas mediante tornillos.
2. **Soporte polea** (Anexo C Plano 01-03-20/Figura N° 32): Posee este nombre ya que posee principalmente la polea conducida. Este posee orificio para el tornillo pasante adosado a la polea conducida, el soporte de las varillas y además las orejas de sujeción a la base.

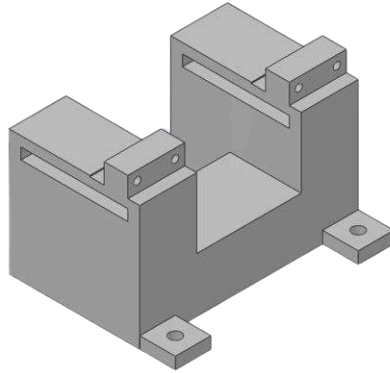


Figura N° 32: Soporte polea conducida

3. **Varillas** (Figura N° 33): Las varillas soportes son elementos estructurales las cuales tienen como función principal proporcionar una trayectoria lineal para el movimiento del carro a lo largo del eje horizontal. Las varillas seleccionadas tienen un diámetro de 8 mm y una longitud de 1 metro, fabricadas en acero inoxidable. Esta elección se fundamenta en la similitud del movimiento del carro con el de las impresoras CNC, donde las varillas de estas dimensiones son comúnmente empleadas. Además, su amplia disponibilidad en el mercado las hace una opción conveniente y práctica para la implementación en este proyecto.



Figura N° 33: Varillas soportes

4. **SK8** (Anexo C Plano 01-03-22/Figura N° 34): Los soportes para las varillas de 8 mm serán los encargados de sostener todo el peso del carro, impidiendo cualquier desplazamiento indeseado de las varillas. Estos soportes serán removibles en caso de rotura y estarán diseñados para funcionar como accesorios en ambos extremos del soporte.

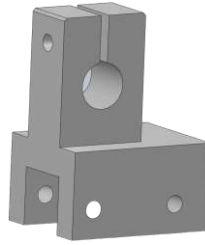


Figura N° 34: SK8

2.6.4. Rodamientos lineales

Los rodamientos lineales son componentes claves diseñados para proporcionar un movimiento lineal suave y sin fricción. Aquí se describen su función, importancia y consideraciones de diseño:

- **Movimiento lineal suave:** Están diseñados para permitir el movimiento lineal del carro a lo largo de las varillas soportes con una fricción mínima. Proporcionan una superficie de deslizamiento suave y uniforme que reduce la resistencia al movimiento y garantiza una respuesta dinámica rápida del sistema.
- **Soporte de cargas axiales y radiales:** Están diseñados para soportar cargas tanto axiales como radiales, proporcionando una capacidad de carga robusta y una estabilidad estructural óptima. Esto asegura que el sistema pueda soportar las fuerzas aplicadas durante el funcionamiento normal y mantener una trayectoria lineal estable para el movimiento del carro.
- **Reducción del desgaste y la vibración:** Diseñados para minimizar el desgaste y la vibración durante el movimiento del carro, lo que prolonga la vida útil del sistema y mejora su rendimiento general. Proporcionan un funcionamiento suave y silencioso, lo que es crucial para aplicaciones donde se requiere un nivel bajo de ruido y vibración.

Al igual que las varillas, se seleccionaron los rodamientos lineales LM8UU, mostrados en la Figura N° 35, debido a su alta disponibilidad en el mercado y su idoneidad para encajar externamente sobre las varillas. Estos rodamientos son ampliamente utilizados en aplicaciones

que requieren un movimiento lineal preciso, lo que los convierte en una opción adecuada para el sistema en desarrollo.



Figura N° 35: Rodamiento lineal LM8UU

Especificaciones técnicas:

- Tipo de rodamiento: Lineal de bolas
- Diámetro interior: 8 mm
- Diámetro exterior: 15 mm
- Longitud: 24 mm
- Material: Acero cromado
- Precisión: Alta precisión
- Cantidad de bolas: 4
- Lubricación: Grasa lubricante preinstalada
- Temperatura de funcionamiento: -20°C a 80°C

2.6.5. Correa

Las correas están diseñadas para transmitir la fuerza generada por el motor hacia el carro, permitiendo así el movimiento controlado a lo largo de las varillas soportes. Al estar vinculadas al carro, las correas se tensan en ambas direcciones para dirigir y controlar el ángulo del péndulo. Trabajando en conjunto con las poleas, las correas facilitan el movimiento del carro a lo largo de las varillas, asegurando una transmisión eficiente y precisa de la energía del motor. Su diseño y tensión adecuados son fundamentales para garantizar un funcionamiento suave y confiable del mecanismo, manteniendo el control preciso del ángulo del péndulo en todo momento.

El sistema de movimiento del péndulo invertido se diseñó considerando que el carro estará en constante desplazamiento, lo que requiere un elemento capaz de soportar el

estiramiento, económico, fácilmente disponible en el mercado y cuyos componentes de transmisión cumplan con los mismos criterios. Basado en estos requisitos, se seleccionó la correa GT2 de 6 mm ilustrada en la Figura N° 36, conocida por su durabilidad y eficiencia en aplicaciones que demandan precisión y resistencia al desgaste.



Figura N° 36: Correa motora GT2

Especificaciones técnicas:

- Tipo de correa: GT2 (diente trapezoidal de 2 mm de paso)
- Ancho de la correa: 6 mm
- Material: Neopreno con núcleo de fibra de vidrio o acero, lo que proporciona durabilidad y resistencia a la tracción.
- Paso de los dientes: 2 mm (la distancia entre los centros de dos dientes consecutivos)
- Espesor de la correa: Aproximadamente 1.38 mm
- Altura del diente: 0.75 mm
- Resistencia a la tracción: Alta, debido al refuerzo interno de fibra de vidrio o acero, lo que permite su uso en aplicaciones que requieren precisión y estabilidad.
- Rango de temperatura: Entre -20 °C a 80 °C.

2.6.6. Poleas

Las poleas permiten la transferencia eficiente de energía, asegurando un movimiento fluido y controlado del carro a lo largo de las varillas soportes. Su función principal radica en proporcionar un punto de contacto para las correas y garantizar su adecuada tensión, lo que permite un funcionamiento preciso y fiable del sistema. Además, las poleas contribuyen a la transmisión uniforme de la fuerza del motor, minimizando la posibilidad de deslizamiento o pérdida de tracción. En conjunto con las correas, las poleas forman un sistema de transmisión robusto y eficiente que desempeña un papel crucial en el control.

Aquí utilizaremos dos poleas:

- **Polea motora** (Figura N° 37): Esta es la encargada de transmitir el par generado por el motor a la correa motora. Esto lo hace posible por medio de una rueda dentada que tiene un eje y se acopla por medio de dos tornillos como se observa en la Figura N° .
- **Polea conducida** (Figura N° 38): Como lo indica este solamente funciona para mantener tensa la correa. La particularidad de esta polea es que posee incorporado un rodamiento que está ubicado en la varilla roscada en el soporte conducido.

Las poleas GT2 6mm fueron elegidas en concordancia con la correa GT2, y se optó por su adquisición debido a que, si se quiere realizar la impresión de estas, la polea motora debido a la impresión tiene mucho juego con la correa provocando que la misma patine y además la conducida hay que realizar la compra del rodamiento que va conectado en su eje el cual es de una dimensión muy chica aumentando su costo que si se compra directamente el cojinete. Estas poleas cuentan con un espárrago en el modelo dentado, lo cual facilita la fijación de la polea. Este diseño permite que la polea gire de manera sincrónica con la rueda dentada asociada.



Figura N° 37: Polea dentada motora



Figura N° 38: Polea conducida

Polea Dentada Gt2 6mm:

- Tipo de polea: GT2
- Ancho de la correa compatible: 6 *mm*
- Número de dientes: 16 dientes
- Diámetro interior del agujero (eje): 5 *mm*
- Material: Aluminio o acero (la mayoría de las poleas GT2 están fabricadas en estos materiales debido a su durabilidad y resistencia a la corrosión)
- Diámetro exterior de la polea: Aproximadamente 12.22 *mm* (puede variar ligeramente dependiendo del fabricante)
- Diámetro efectivo: 10.18 *mm*
- Paso de la correa: 2 *mm*
- Altura del diente: 0.75 *mm*
- Número de canales de la correa: 1
- Tipo de fijación: Normalmente incluye un tornillo de ajuste (grub screw) para fijar la polea al eje del motor de manera segura.

Polea Gt2 Rodamiento 20 Sin Dientes 6mm:

- Tipo de polea: Polea de guía (sin dientes)
- Diámetro exterior: 20 *mm*
- Ancho de la correa compatible: 6 *mm*
- Diámetro del agujero (eje): 3 *mm* (el agujero central está diseñado para ejes de 3 *mm* de diámetro)
- Rodamiento: Interno (generalmente un rodamiento de bolas de alta calidad para asegurar un giro suave y sin fricción)
- Material: Aluminio.

- Tipo de rodamiento: Rodamiento sellado para mantener la suavidad y reducir el mantenimiento.

2.6.7. Engranajes

En el proyecto se utilizarán dos engranajes con el propósito de transmitir el movimiento mecánico desde el motor hasta la polea motora. Se buscó mantener una relación de transmisión de 1:1, debido a que el motor tiene una potencia limitada. Una reducción en la transmisión resultaría en una pérdida significativa de velocidad en el desplazamiento del carro, lo cual complicaría el control del ángulo. Esta configuración asegura que el sistema mantenga una velocidad adecuada, facilitando un control más preciso y efectivo del ángulo.

Se tienen dos engranajes:

1. Rueda dentada motora (Figura N° 39): Esta rueda es responsable de transmitir la potencia del motor al engranaje asociado a la polea motora. Su función principal es transferir el movimiento rotacional del motor hacia la polea, garantizando la sincronización adecuada del sistema.
2. Rueda dentada conducida (Figura N° 40): La rueda dentada conducida se encarga de hacer girar la polea dentada mediante un eje que dispone de dos orificios para fijar la polea. Este diseño asegura el movimiento sincronizado entre el engranaje y la polea, manteniendo la precisión en la transmisión del movimiento.

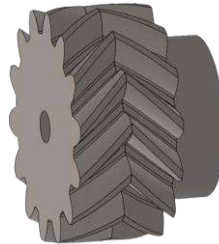


Figura N° 39: Engranaje motor

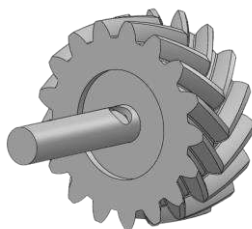


Figura N° 40: Engranaje conducido

2.6.8. Armado de dispositivo

Para ensamblar el péndulo invertido, se debieron seguir los pasos detallados a continuación:

1. Colocar el encoder en el cuerpo del carro y atornillar. Luego poner el eje del encoder (Figura N° 41).

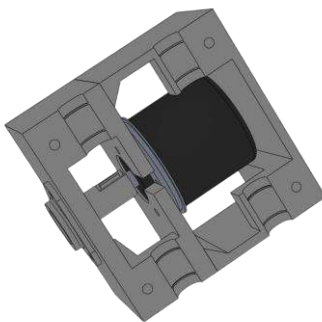


Figura N° 41: Paso N° 1

2. En la base del carro colocar los extremos de la correa en los soportes dentados y apretar con los precintos. Luego colocar los rodamientos lineales (Figura N° 42).

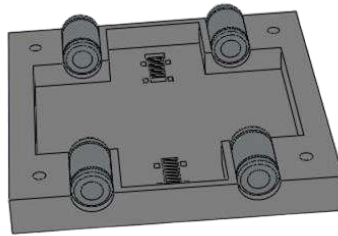


Figura N° 42: Paso N° 2

3. En la cubierta atornillar la caja RJ45 y conectar los cables del encoder a la misma (Figura N° 43).

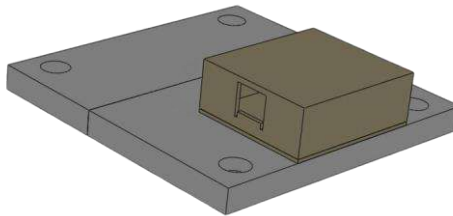


Figura N° 43: Paso N° 3

4. Ensamblar el carro con tornillos y tuerca (Figura N° 44).

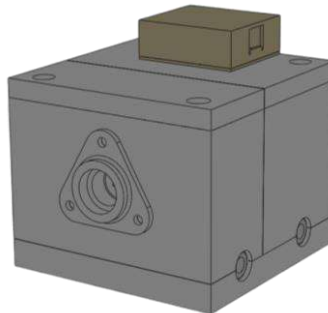


Figura N° 44: Paso N° 4

5. Se colocan los soportes de varillas, agregar los soportes de varillas SK8 y finalmente en el soporte motor agregarle el motor en el orificio y atornillar (Figura N° 45).

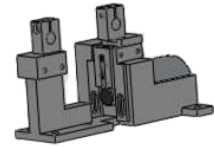
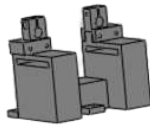


Figura N° 45: Paso N° 5

6. Atravesar los rodamientos lineales con las varillas y luego sujetar con los soportes SK8. Atornillar los soportes en una base estable (Figura N° 46).

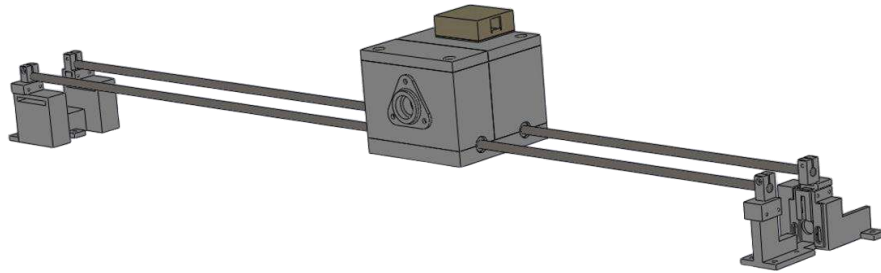


Figura N° 46: Paso N° 6

7. Rodear con la correa las poleas y poner los soportes de rodamiento y suplemento (Figura N° 47).

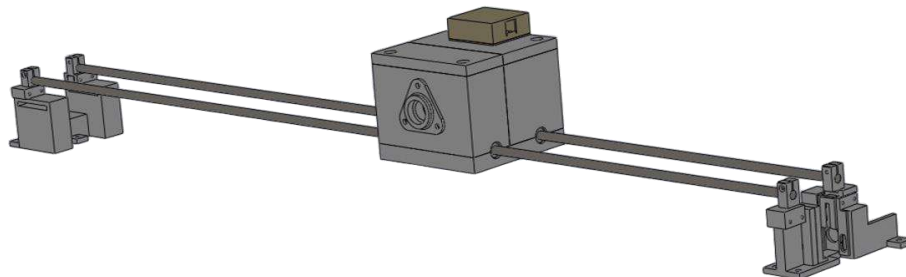


Figura N° 47: Paso N° 7

8. Introducir en la varilla de aluminio la unión interior. Luego en el otro extremo el eje del péndulo y atornillar ambos elementos (Figura N° 48).



Figura N° 48: Paso N° 8

9. Colocar la otra varilla en la unión interior y por el otro extremo introducir la unión exterior y llevarla hasta la mitad para atornillar (Figura N° 49).



Figura N° 49: Paso N° 9

10. Poner el soporte de pesas en el extremo libre del péndulo, atornillar y luego con el péndulo ensamblado colocar el eje del péndulo dentro del adaptador del eje del encoder (Figura N° 50).

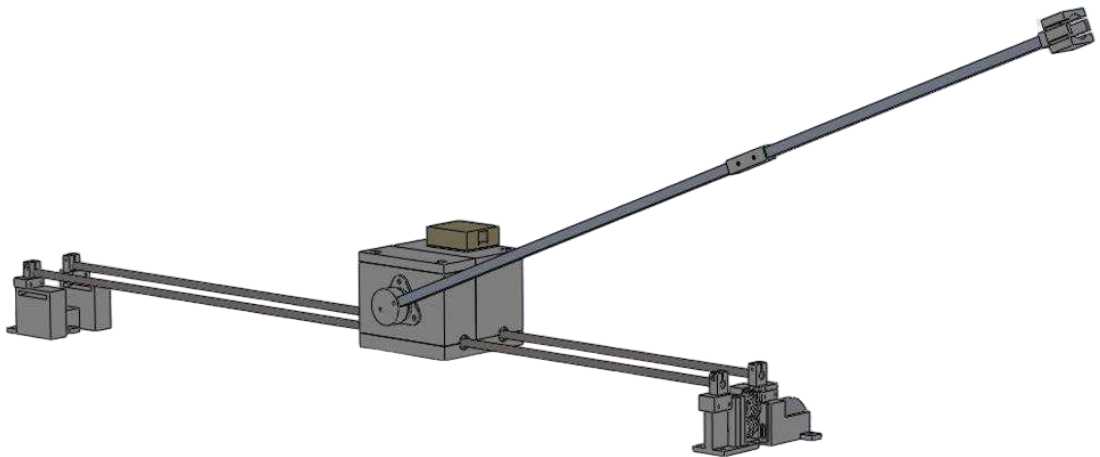


Figura N° 50: Paso N° 10

11. Conectar la placa electrónica como la Figura N° 14. Y finalmente se encuentra ensamblado el péndulo en su totalidad como lo muestra la Figura N° 51.

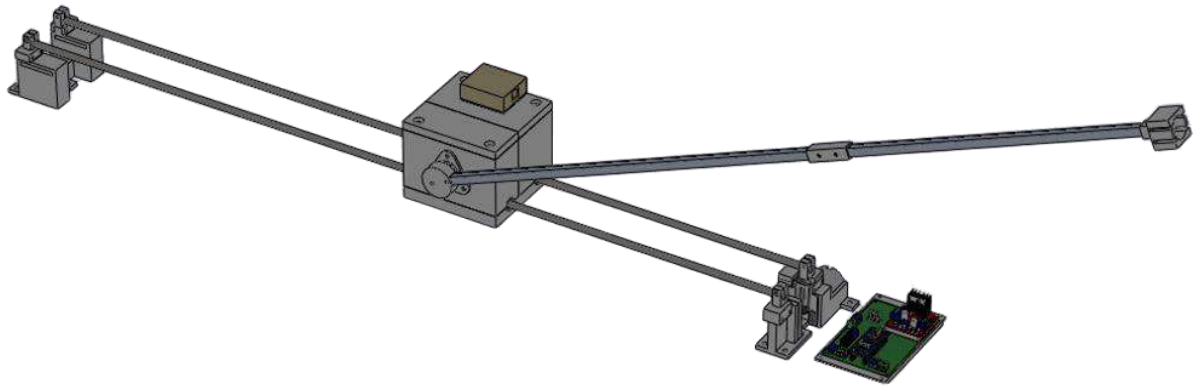


Figura N° 51: Ensamblaje de péndulo

CAPITULO 3: Modelado matemático de la planta

Modelar la planta es crucial para implementar un control efectivo porque proporciona una comprensión aproximada del comportamiento del sistema y permite diseñar un controlador que pueda gestionar adecuadamente el proceso. El modelo facilita la simulación y prueba del controlador en un entorno virtual, asegurando la estabilidad y el rendimiento del sistema antes de la implementación real. Además, permite optimizar el diseño del controlador y adaptarlo a cambios en la planta. Sin un modelo, sería difícil prever cómo el controlador interactuará con la planta y cumplir con los objetivos de desempeño. Para ello, se considera el sistema de la Figura N° 52 que consta de un carro que se mueve sobre un riel horizontal y un péndulo unido al carro por un pivote. El péndulo puede oscilar hacia adelante y hacia atrás, y el carro puede moverse en la dirección horizontal. Los parámetros principales que describen este sistema son:

- x : Posición del carro a lo largo del riel.
- \dot{x} : Velocidad lineal del carro.
- \ddot{x} : Aceleración lineal del carro.
- θ : Angulo del péndulo respecto a la vertical.
- $\dot{\theta}$: Velocidad angular del péndulo.
- $\ddot{\theta}$: Velocidad angular del péndulo.
- u : Fuerza aplicada en el carro.
- T : Fuerza resultante del eje del péndulo a la varilla
- F : Fuerza resultante del carro.
- M : Masa del carro.
- m : Masa del péndulo.
- l : Longitud del péndulo.

- b_c : Fricción viscosa del carro.
- b_p : Fricción viscosa del péndulo.
- τ : Momento en el punto de unión en el carro.
- I_p : Momento de inercia del péndulo.
- g : Aceleración de la gravedad.

El carro puede ser empujado o tirado para controlar el movimiento del péndulo y mantenerlo en posición vertical.

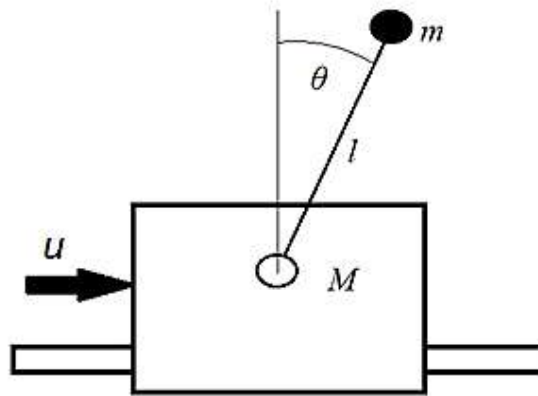


Figura N° 52: Esquemático péndulo invertido

La segunda ley de Newton para el carro se puede expresar como:

$$M\ddot{x} = u - F_x - b_c\dot{x} \quad (1)$$

Si se tiene en cuenta las fuerzas radiales y tangenciales en el péndulo se tiene:

$$T\cos(\theta) - mg\cos(\theta) = m(\ddot{x}\cos(\theta) + l\dot{\theta}^2) \quad (2)$$

$$T\sin(\theta) - mg\sin(\theta) = m(l\ddot{\theta} + \ddot{x}\sin(\theta)) \quad (3)$$

Si se analiza el momento respecto al punto de unión con el carro:

$$\sum \tau = I_p \ddot{\theta} \quad (4)$$

Considerando los momentos debido a la gravedad y la tensión:

$$mgl\sin(\theta) = ml^2\ddot{\theta} + ml\ddot{x}\cos(\theta) \quad (5)$$

Si se relaciona la fuerza F_x y la aceleración del carro \ddot{x} :

$$F_x = T\sin(\theta) = m(\ddot{x}\cos(\theta) + l\dot{\theta}^2) \quad (6)$$

Utilizando la ecuación (5) de momentos y organizando los términos:

$$mgl\sin(\theta) = ml^2\ddot{\theta} + ml\ddot{x}\cos(\theta) \quad (7)$$

Se puede despejar la aceleración angular $\ddot{\theta}$:

$$\ddot{\theta} = \frac{mgl\sin(\theta) - ml\ddot{x}\cos(\theta)}{ml^2} \quad (8)$$

Simplificando:

$$\ddot{\theta} = \frac{g\sin(\theta) - \ddot{x}\cos(\theta)}{l} \quad (9)$$

Usando la ecuación (1) del carro:

$$M\ddot{x} = u - m(\ddot{x}\cos(\theta) + l\dot{\theta}^2) - b_c\dot{x} \quad (10)$$

Simplificando:

$$(M + m)\ddot{x} = u - ml(\dot{\theta}^2\sin(\theta) + \ddot{\theta}\cos(\theta)) - b_c\dot{x} \quad (11)$$

Finalmente se obtiene \ddot{x} :

$$\ddot{x} = \frac{u - ml(\dot{\theta}^2\sin(\theta) - \ddot{\theta}\cos(\theta)) - b_c\dot{x}}{(M + m)} \quad (12)$$

Sustituyendo en la ecuación (9):

$$\ddot{\theta} = \frac{g \sin(\theta) - \left(\frac{u - ml(\dot{\theta}^2 \sin(\theta) + \ddot{\theta} \cos(\theta)) - b_c \dot{\theta}}{(M + m)} \right) \cos(\theta)}{l} \quad (13)$$

Acomodando se obtiene:

$$\ddot{\theta} = \frac{g \sin(\theta) - \left(\frac{u + ml(\dot{\theta}^2 \sin(\theta))}{(M + m)} \right) \cos(\theta) - b_c \dot{\theta}}{l \left(\frac{4}{3} - \frac{m \cos^2(\theta)}{M + m} \right)} \quad (14)$$

A partir de la ecuación (12) y la ecuación (14) queda descrito el comportamiento dinámico que tiene el carro y el péndulo respectivamente. De tal manera ya se está en condiciones de realizar pruebas mediante simulaciones para saber si las ecuaciones obtenidas responden igual que la planta real.

3.1. Función de transferencia

Para relacionar el ángulo θ que forma la varilla respecto al eje vertical y la fuerza u , la cual será responsable de corregir dicho ángulo mediante una correa, es necesario determinar la función de transferencia correspondiente. Esta función es fundamental, ya que establece la relación entre la entrada (fuerza u) y la salida (ángulo θ) del sistema, permitiendo comprender y predecir cómo la fuerza aplicada influirá en la posición del péndulo.

Debido a la falta de linealidad e invariancia en el tiempo, la aplicación directa de la transformada de Laplace presenta una gran dificultad, ya que no se cumplen las condiciones necesarias para realizar las transformaciones y suposiciones requeridas. Por lo tanto, se deben emplear métodos indirectos para aplicar la transformada de Laplace. Con este fin, se considera que el sistema realiza pequeñas variaciones en el ángulo en periodos muy breves de tiempo. Por lo tanto, se realizan las siguientes suposiciones para linealizar el sistema:

$$\cos(\theta) \cong 1 \quad (15)$$

$$\sin(\theta) \cong \theta \quad (16)$$

Por lo tanto, se obtiene las ecuaciones de sumatoria de fuerzas y momento:

$$\ddot{\theta} = \frac{g\theta - \left(\frac{u + ml(\dot{\theta}^2 \theta)}{(M + m)} \right) - b_c \dot{\theta}}{l \left(\frac{4}{3} - \frac{m}{M + m} \right)} \quad (17)$$

$$\ddot{x} = \frac{u + ml(\dot{\theta}^2 \theta - \ddot{\theta}) - b_p \dot{x}}{M + m} \quad (18)$$

Si se ignora los términos de orden superior obtenemos:

$$\ddot{\theta} = \frac{g\theta - \frac{u}{M + m} - b_c \dot{\theta}}{l \left(\frac{4}{3} - \frac{m}{M + m} \right)} \quad (19)$$

$$\ddot{x} = \frac{u - ml\ddot{\theta} - b_p \dot{x}}{M + m} \quad (20)$$

Se aplica transformada de Laplace a ecuación (19) y ecuación (20):

$$s^2 \Theta(s) = \frac{g\Theta - \frac{U(s)}{M + m} - sb_c \Theta(s)}{l \left(\frac{4}{3} - \frac{m}{M + m} \right)} \quad (21)$$

$$s^2 X(s) = \frac{U(s) - s^2 ml \Theta(s) - sb_p X(s)}{M + m} \quad (22)$$

Se reorganiza la ecuación (21):

$$s^2 \Theta(s) l \left(\frac{4}{3} - \frac{m}{M + m} \right) = g\Theta - \frac{U(s)}{M + m} - sb_c \Theta(s) \quad (23)$$

Se agrupa los términos con $\Theta(s)$:

$$\Theta(s) \left[s^2 l \left(\frac{4}{3} - \frac{m}{M + m} \right) + b_c s - g \right] = - \frac{U(s)}{M + m} \quad (24)$$

Entonces:

$$\theta(s) = -\frac{U(s)}{(M+m) \left[s^2 l \left(\frac{4}{3} - \frac{m}{M+m} + b_c s - g \right) \right]} \quad (25)$$

Se halla la función de transferencia:

$$\frac{\theta(s)}{-U(s)} = \frac{1}{(M+m) \left[s^2 l \left(\frac{4}{3} - \frac{m}{M+m} + b_c s - g \right) \right]} \quad (26)$$

Finalmente simplificando y teniendo la fricción viscosa del péndulo se obtiene:

$$G(s) = \frac{\theta(s)}{-U(s)} = \frac{1}{Mls^2 - (M+m)g + b_c s + \frac{b_p s}{l}} \quad (27)$$

3.2. Simulación planta lineal

Cabe aclarar que la función de transferencia no representa la planta por que al estar linealizada esta no captura adecuadamente todas las dinámicas complejas y no lineales del sistema.

Dada la anterior función de transferencia G de la ecuación (27) se procede a hacer la simulación por medio del software Matlab con las siguientes parametrizaciones:

- $M = 751.3 \text{ g}$
- $m = 329.73 \text{ g}$
- $l = 0.93 \text{ m}$

La simulación se llevó a cabo utilizando la función Sisotool, que proporciona gráficos de la respuesta del sistema a un escalón unitario como lo muestra la Figura N° 53. Esto permite analizar cómo reacciona el sistema ante una variación en la entrada o la presencia de perturbaciones. Además, la herramienta muestra el lugar geométrico de las raíces del sistema como la Figura N° 54, lo que facilita la comprensión de su comportamiento dinámico.

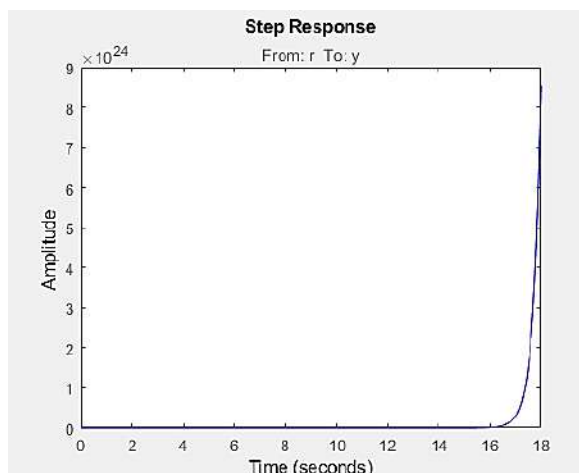


Figura N° 53: Respuesta del sistema al escalón unitario

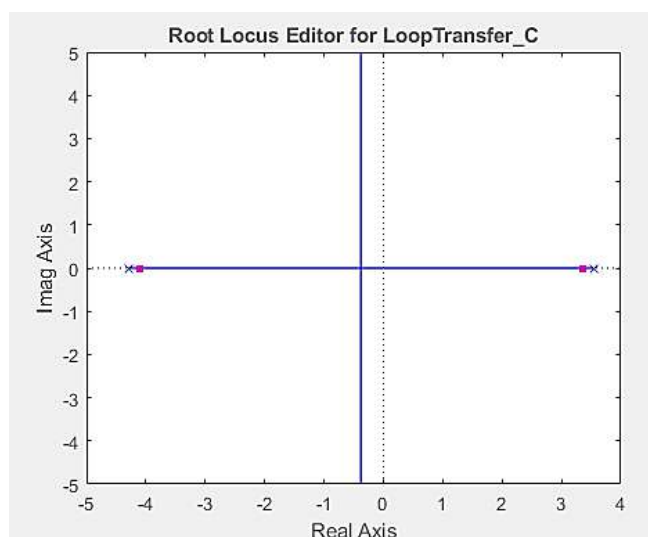


Figura N° 54: Lugar geométrico de las raíces del sistema

En el lugar geométrico de las raíces, se observa que el sistema presenta dos polos: uno ubicado en -4.2936 y el otro en 3.535. La presencia de una raíz positiva indica que el sistema es inherentemente inestable, ya que el péndulo tiende a caer en lugar de estabilizarse en 0° , que es la posición deseada.

El análisis del lugar geométrico entre los polos sugiere que es posible estabilizar el sistema mediante un incremento de la ganancia. Sin embargo, dado que se trata de un sistema de tipo 0, este enfoque no corregiría el error en estado estable. Además, implicaría una ganancia

proporcional muy elevada. Como consecuencia, el sistema podría volverse inestable frente a cualquier perturbación, resultando en un comportamiento de control similar a un sistema de tipo on/off, lo cual no es ideal para un control preciso y continuo.

La inestabilidad se refleja en el gráfico de la respuesta al impulso, donde la respuesta diverge exponencialmente a medida que $(t \rightarrow \infty)$.

3.3. Simulación en planta no lineal

Se lleva a cabo el mismo procedimiento para la planta no lineal. Partiendo desde 0° , que indica que el péndulo se encuentra en posición vertical, y aplicando una perturbación de 1° , se obtiene la respuesta mostrada en la Figura N° 55. En esta respuesta, se observa que, en ausencia de fricción, el péndulo rotaría indefinidamente.

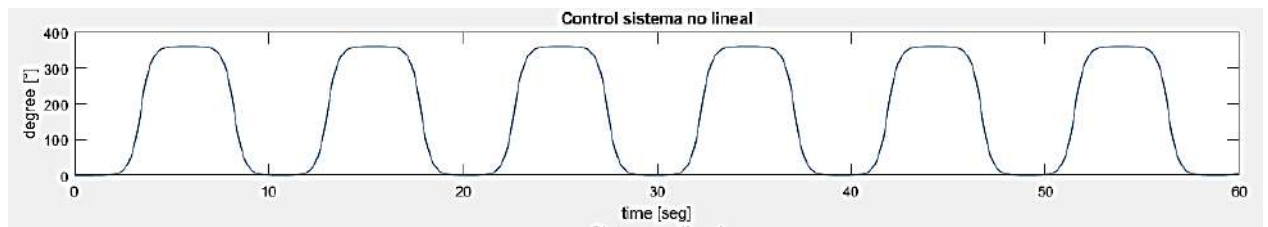


Figura N° 55: Simulación de comportamiento del ángulo de péndulo en sistema no lineal

Al incorporar la fricción según la ecuación (22), se observa que el sistema comienza a converger hacia 180° . Cabe destacar que los valores de la fricción viscosa del péndulo y del carro, definidos mediante ensayos con el péndulo real, son $b_c = 0.01$ y $b_p = 0.01$. Como se puede observar en la Figura N° 56, el sistema alcanza su estado de equilibrio estable alrededor de los 50 segundos, lo cual es consistente con el comportamiento observado en la realidad.

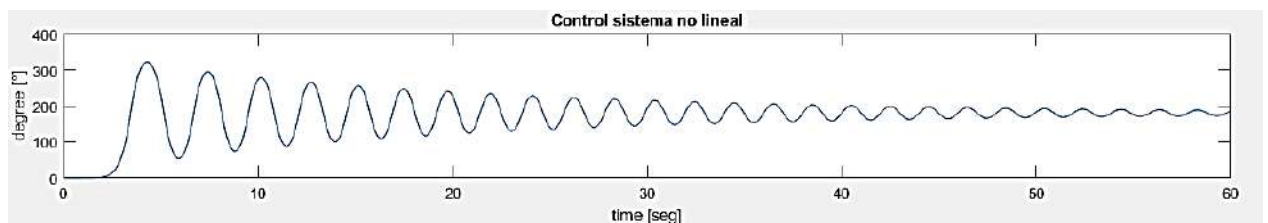


Figura N° 56: Simulación de planta considerando fricción viscosa de péndulo y carro

3.4. Discretización de la función de transferencia

El objetivo de este trabajo fue desarrollar un control e implementarlo en un microcontrolador, utilizando herramientas en tiempo discreto que facilitaron dicha implementación. Para ello, fue necesario discretizar la función de transferencia. Al convertir la función a tiempo discreto y graficar el lugar geométrico de las raíces (LGR) correspondiente a la Figura N° 57, se utilizó como técnica de discretización la aplicación de un retenedor de orden cero. El cual se rige por la siguiente ecuación [11]:

$$G(z) = (1 - z^{-1})Z \left\{ \mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} \right\} \quad (28)$$

Teniendo en cuenta una frecuencia de muestreo de 20 kHz , se obtiene el siguiente resultado con la ecuación (28):

$$G(z) = \frac{1.789 \times 10^{-9}z + 1.798 \times 10^{-9}}{z^2 - 2z + 1} \quad (29)$$

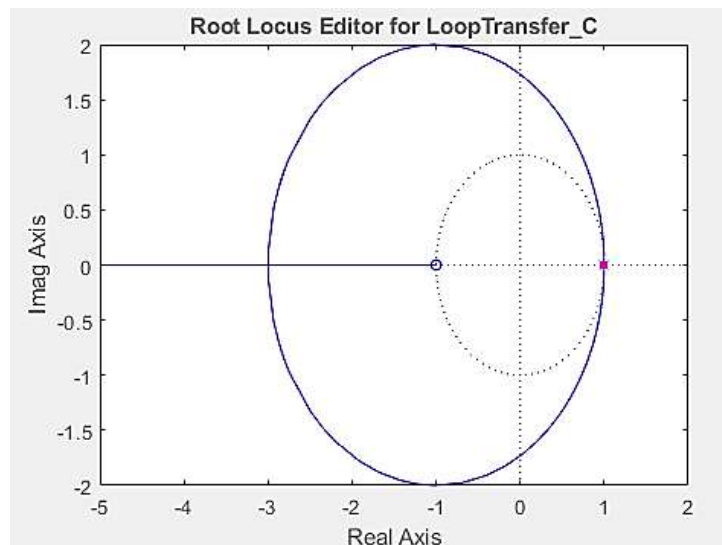


Figura N° 57: LGR sistema discretizado

En el gráfico de la Figura N° 58, se observa que el sistema es inestable debido a la presencia de un polo fuera del círculo unitario ubicado en 1.0001768. Este hecho indica que la estabilidad del sistema no está garantizada en su configuración discreta.

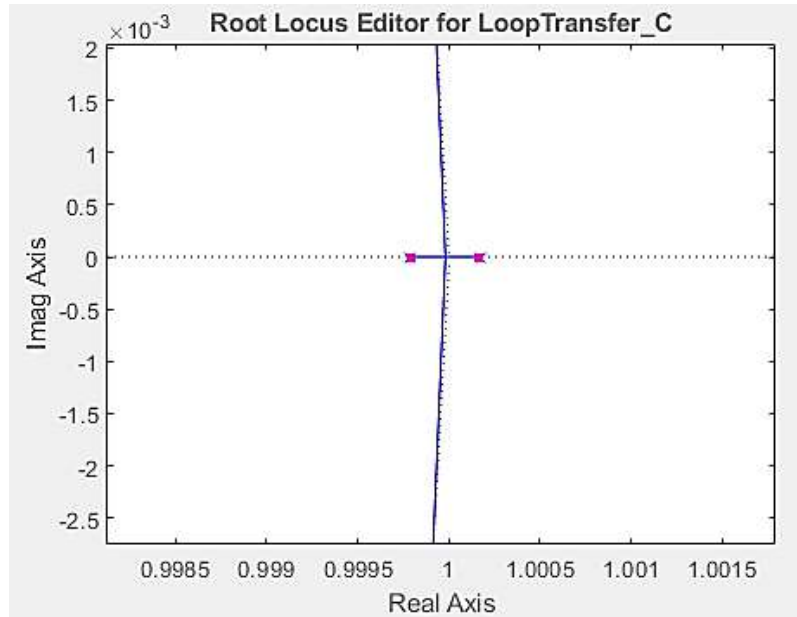


Figura N° 58: Ubicación polo inestable

CAPITULO 4: Diseño del controlador

El diagrama de bloques del sistema, mostrado en la Figura N° 59, se compone de tres bloques principales: El bloque G_p representa la planta del sistema, el bloque *Encoder* es el que recoge los datos del ángulo del péndulo y cierra el lazo de control. Finalmente, el bloque G_c es el controlador, que calcula la diferencia entre el ángulo actual y el ángulo de referencia. Basado en esta diferencia, el controlador genera una señal de control, en este caso una salida PWM. Esta señal es enviada al motor, el cual ajusta su velocidad y sentido para ejercer una fuerza que contrarresta el aumento del ángulo del péndulo.

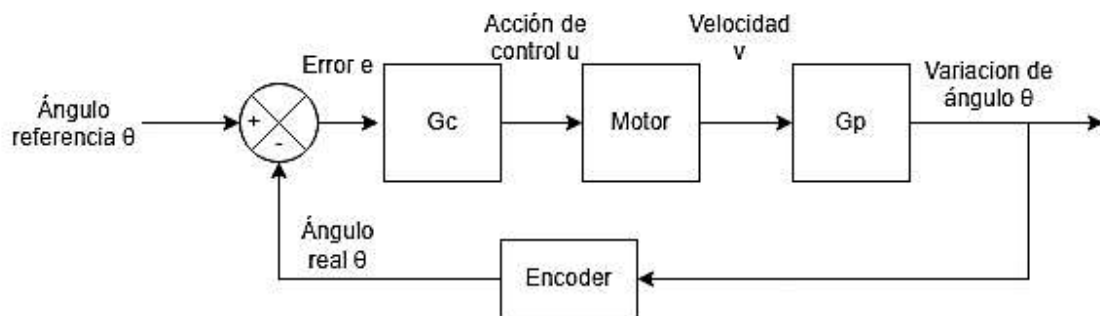


Figura N° 59: Diagrama de bloques a lazo cerrado de la planta

Para el diseño del controlador, se utilizó el lugar geométrico de las raíces (LGR) mediante la herramienta Sisotool propia de Matlab, con el fin de equilibrar la rapidez en la acción del control con la capacidad de fuerza del motor. El controlador fue configurado para alcanzar un tiempo de asentamiento aproximado de 0.5 segundos, asegurando que no se exceda la fuerza disponible.

En el LGR de la Figura N° 60, se colocó un cero en 0.99995, muy próximo al polo de la planta, con el objetivo de atraer el lugar geométrico de las raíces dentro del círculo y estabilizar la respuesta del sistema. Además, para que el controlador sea realizable se añadió un polo en 0.9 resultando en la implementación de un control por adelanto de fase. Esto permitió obtener un controlador con alta ganancia que tiene como ventaja la estabilización del sistema y mantener un error en estado estable despreciable. Con estas modificaciones, la función de transferencia del controlador se establece como sigue:

$$G_c(z) = \frac{6.0002 \cdot 10^6 (z - 0.9995)}{z - 0.9} \quad (30)$$

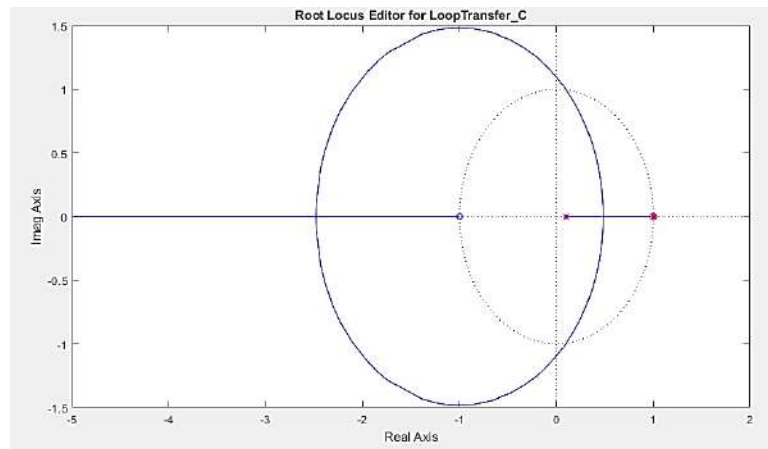


Figura N° 60: LGR de sistema con el controlador

Teniendo en cuenta lo anterior, se somete al sistema a la respuesta al escalón y se obtiene los resultados mostrados en la Figura N° 61 donde la respuesta es aceptable con:

- $pr = 35.4\%$
- $st = 0.0037 [s]$

Es importante considerar que, aunque se observa un pico considerable en la respuesta del sistema, este se mantiene por debajo del nivel mínimo de lectura del encoder, que cuenta con una resolución de 0.25° . Debido a que este pico se manifiesta en un breve intervalo de tiempo, no debería presentar problemas significativos al momento de implementar el control. Es relevante señalar que, a pesar de la ausencia de un integrador en el controlador, el sistema es capaz de disminuir el error en estado estable a tal punto que llega a ser imperceptible.

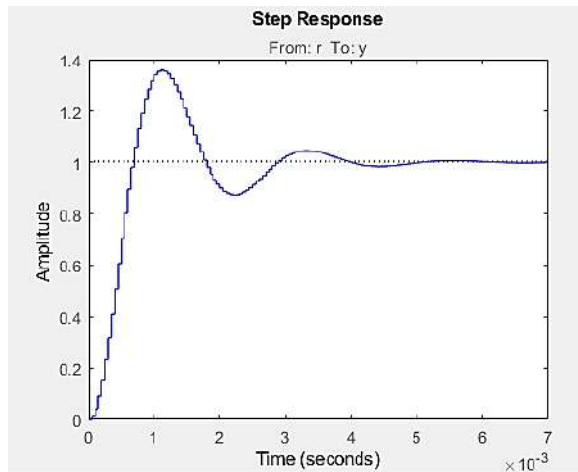


Figura N° 61: Respuesta al impulso sistema controlado

4.1. Ecuación en diferencia finita

Para implementar el controlador es necesario expresar este en términos de ecuaciones en diferencias para poder ser programado en la STM32 debido a la naturaleza discreta del hardware digital. Es por eso por lo que aparece la necesidad de convertir el modelo continuo en discreto para procesar las señales en intervalos de tiempo definidos por la frecuencia de muestreo.

Se parte de la ecuación (30) se obtiene:

$$G_c(z) = \frac{6.0002 \times 10^6 (z - 0.99995)}{z - 0.9} = \frac{U(z)}{E(z)} \quad (31)$$

Se reordenan los términos y se obtiene:

$$U(z)(1 - 0.9z^{-1}) = 6.0002 \times 10^6 E(z)(1 - 0.99995z^{-1}) \quad (32)$$

Despejando $u(k)$ se logra finalmente la acción de control a implementar:

$$u(k) = 0.9u(k - 1) + 6.0002 \times 10^6 (e(k) - 0.99995e(k - 1)) \quad (33)$$

La ecuación (33) será la implementada en el microcontrolador donde:

- $e(k)$ muestra actual de la diferencia entre en ángulo de referencia y el actual.
- $e(k - 1)$ muestra anterior de la diferencia entre en ángulo de referencia y el actual.
- $u(k)$ acción de control actual.
- $u(k - 1)$ acción de control pasada.

4.2. Simulación

El objetivo de la simulación fue evaluar el desempeño de los controles diseñados y, principalmente, verificar si las ecuaciones obtenidas durante el modelado de la planta correspondían con el comportamiento del dispositivo diseñado.

Para desarrollar el diagrama de bloques de la Figura N° 59, se utilizó el entorno Simulink de Matlab. El bloque principal contiene el controlador, cuya entrada se configura con un ángulo de referencia de cero, conectado directamente a la función de transferencia del sistema. La salida de este bloque proporciona el ángulo, el cual se retroalimenta al controlador para calcular el error.

En el bloque inferior, se incluyó el modelo no lineal del sistema con el objetivo de evaluar el impacto real del controlador. A través de los visualizadores de Matlab, se analizaron los efectos sobre las diferentes variables involucradas. Esta disposición permite realizar una evaluación exhaustiva del rendimiento y la eficacia del controlador bajo diversas condiciones operativas. En el programa de Simulink existen dos configuraciones posibles:

- **Planta lineal** (Figura N° 62): Este componente se emplea para verificar que la planta responde conforme a lo diseñado utilizando la herramienta Sisotool. Para lograrlo, se conecta la salida del controlador a la función de transferencia de la

planta, y se aplica una entrada escalón. Esto permite observar y analizar la respuesta de la planta en función de las expectativas del diseño original.

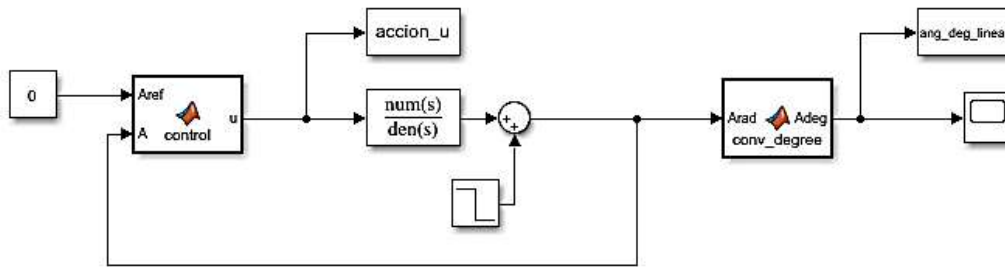


Figura N° 62: Diagrama de bloques para planta lineal

- **Planta no lineal** (Figura N° 63): Esta configuración es crucial, ya que proporciona una aproximación realista de cómo responderá el control en condiciones prácticas. Para lograr esto, la salida del controlador se conecta a un bloque que incorpora las ecuaciones definidas en la ecuación (12) y la ecuación (14). Este bloque recibe como entradas el ángulo θ , la acción de control u , la velocidad angular $\dot{\theta}$ y la velocidad lineal \dot{x} , y produce como salidas la aceleración angular $\ddot{\theta}$ y la aceleración lineal \ddot{x} .

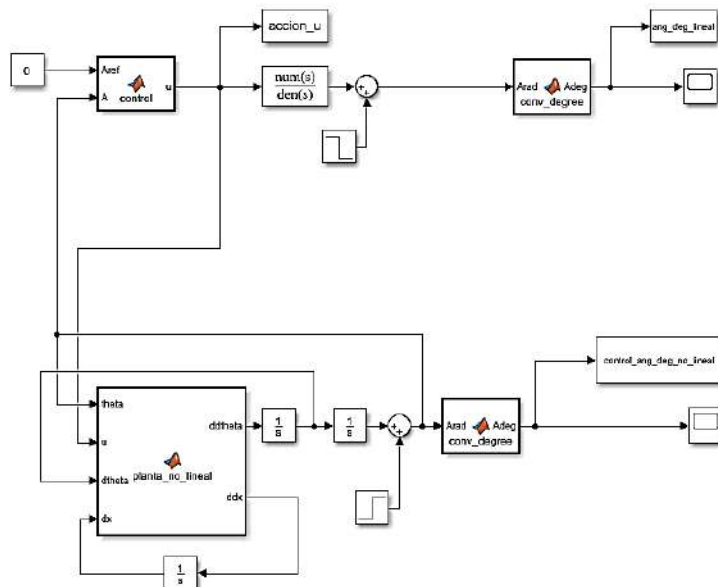


Figura N° 63: Diagrama de bloques para planta no lineal

4.3. Respuesta a una perturbación de 1°

- **Planta lineal:** La Figura N° 64 muestra el comportamiento del ángulo del péndulo en respuesta a una perturbación de 1°. Se observa un pico inicial debido a la perturbación, seguido por una rápida corrección del sistema con un contra pico de aproximadamente 0.39°. Posteriormente, se presentan oscilaciones que amortiguan la perturbación, llevando el ángulo finalmente al objetivo de 0°. Este comportamiento es resultado de la acción del controlador, que trabaja para corregir la desviación generada. Todo el proceso ocurre en aproximadamente 0.01 segundos, lo que refleja la eficacia del controlador en ajustar rápidamente la referencia y estabilizar el sistema en un corto período de tiempo.

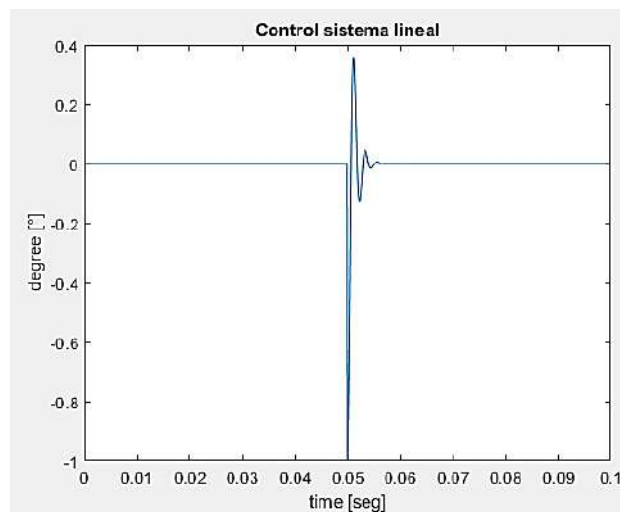


Figura N° 64: Respuesta de planta lineal ante una perturbación de 1°

- **Planta no lineal:** Utilizando el diagrama de bloques mostrado en la Figura N° 52, se realizó un procedimiento similar al aplicado para la planta lineal. En este caso, como se observa en la Figura N° 65, el sistema muestra un mejor comportamiento, ya que no se presentan subamortiguamientos. Esto indica que el sistema es menos oscilatorio y, por lo tanto, resulta más conveniente para mantener una respuesta más estable lo que resulta beneficioso para el sistema mecánico debido al juego presente en las piezas mecánicas ante pequeños movimientos.

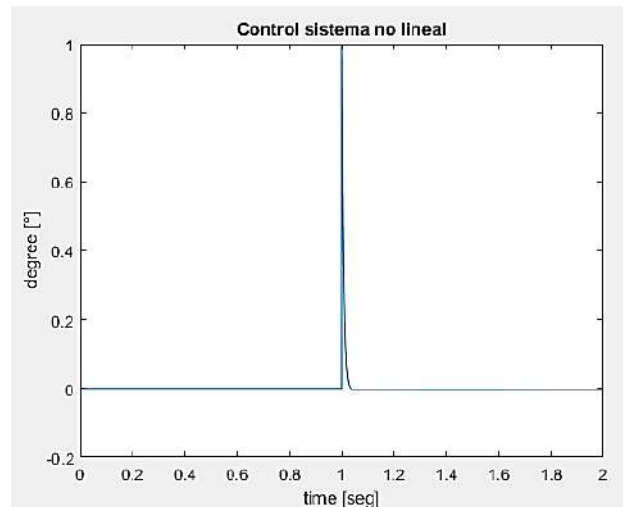


Figura N° 65: Respuesta del sistema ante una perturbación de 1°

Si se analiza la acción de control del sistema como se muestra la Figura N° 66 se puede apreciar que la misma es elevada y se requiere una gran cantidad de energía para lograr los objetivos de control. Esto se traduce en que el motor nunca podría alcanzar ese valor. Por lo cual se procede a la restricción de la acción de control, suponiendo mediante ensayos que el motor tiene una potencia de 35[W] y llega a 500 [rpm] el par ofrecido por el mismo es de [0,66 Nm] por lo cual si lo convertimos a una unidad más apta para nuestro sistema nos quedaría 66 [Ncm]. Se procede entonces a la restricción para esos valores lo cual la acción de control será la mostrada por la Figura N° 56 donde se puede apreciar el mismo comportamiento que la Figura N° 67 pero más realista.

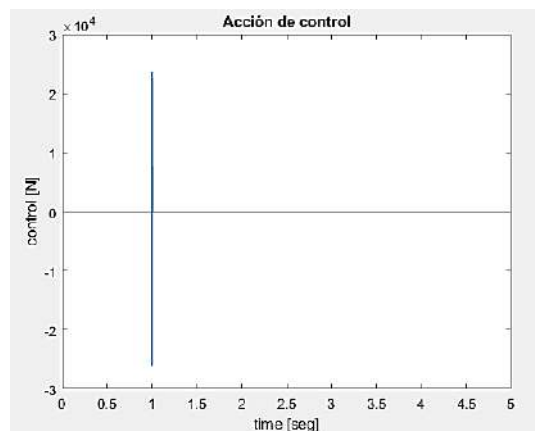


Figura N° 66: Acción de control ante una perturbación de 1°

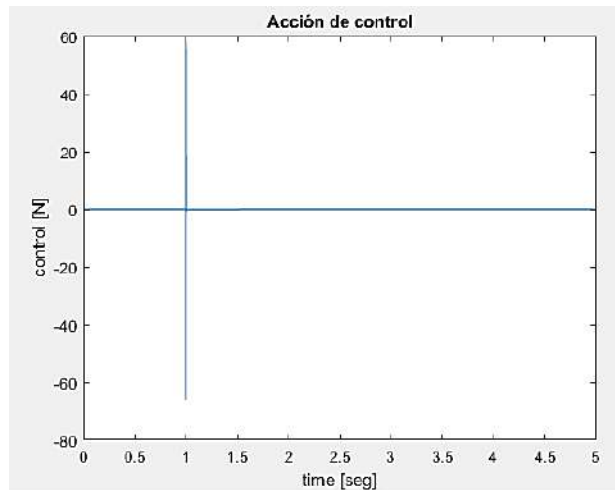


Figura N° 67: Acción de control restringida

Por otra parte, el ángulo también es afectado por esta restricción como lo muestra la Figura N° 68 donde el ángulo llega a corregirse en un tiempo de 5 segundos esto quiere decir que si bien se corrige el ángulo esta demora mucho tiempo lo cual denota la falta de potencia del motor.

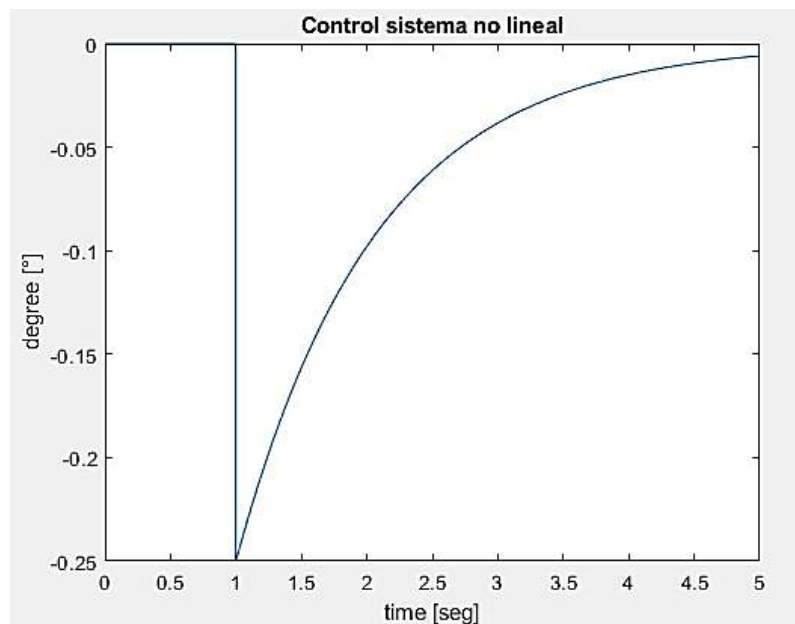


Figura N° 68: Angulo resultante ante la restricción de la acción de control

CAPITULO 5: Configuración stm32

Para la programación de la STM32, se utilizó el software STM32CubeIDE, desarrollado por la misma empresa del microcontrolador. Dentro de este entorno, se configuraron las entradas necesarias para alcanzar los objetivos propuestos, como se ilustra en la Figura N° 69. La disposición de las entradas se estableció de la siguiente manera:

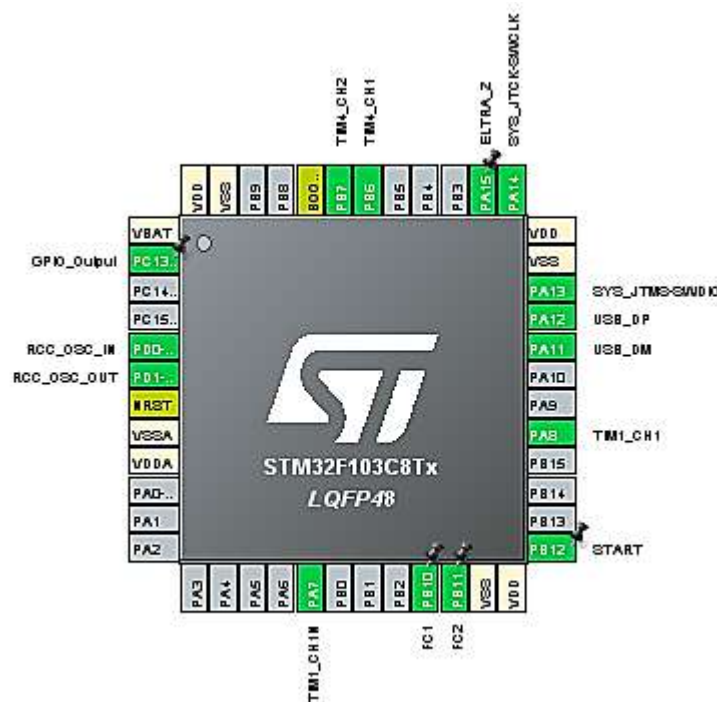


Figura N° 69: Pinout STM32

- PA7 - Timer 1 – PWMN: Salida negada del PWM hacia uno de los canales del puente H.
- PA8 - Timer 1 – PWM: Salida del PWM hacia el otro canal del puente.
- PA11 - USB-DP: Pin de comunicación.
- PA12 – USB-DM: Pin de comunicación.
- PA15 – Eltra-Z: Salida de marca de revolución del encoder.
- PB6 – Timer 4 – Eltra-S1: Sensor 1 perteneciente a encoder.
- PB7 – Timer 4 – Eltra S2: Sensor 2 perteneciente a encoder.
- PB10 – FC1 – Final de carrera 1.
- PB11 – FC2 – Final de carrera 2.
- PB12 – START – Inicio de control

Una vez configuradas las entradas y salidas, es necesario configurar los temporizadores de tal manera que:

TIMER 1: Encargado del funcionamiento del motor se lo configuro para obtener una salida de 20kHz y que no sea audible el funcionamiento. Para ello se tiene:

$$f_{timer} = \frac{f_{clk}}{(P + 1)(CP + 1)} \quad (34)$$

Donde:

- f_{timer} : Frecuencia buscada de 20kHz
- f_{clk} : Frecuencia de la STM32
- P : Prescaler
- CP : Counter Period

Configurando la STM a la frecuencia máxima que puede entregar como muestra la Figura N° 70, es decir $f_{clk} = 72MHz$, un $P = 0$ y despejando CP :

$$CP = \frac{f_{clk}}{(P + 1)f_{timer}} - 1 \quad (35)$$

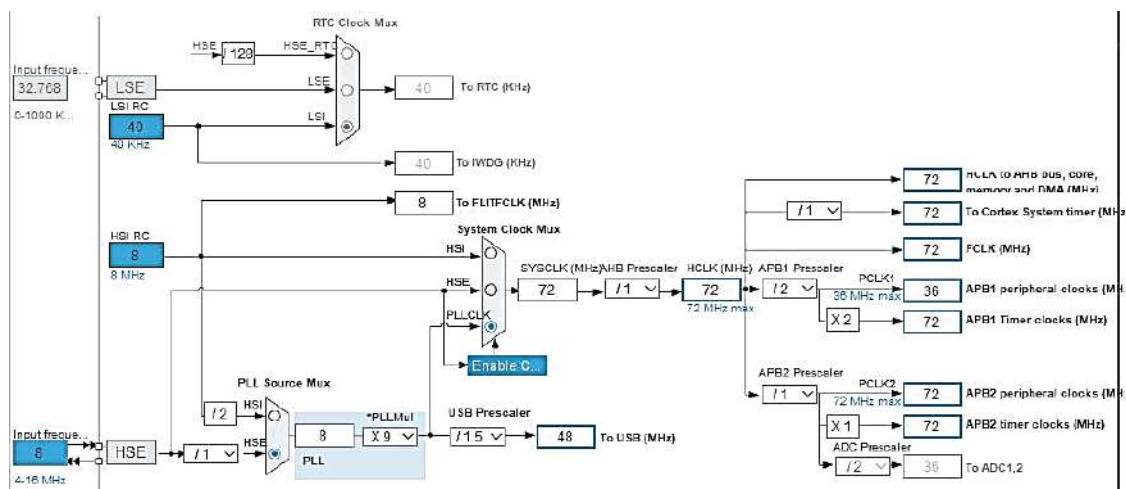


Figura N° 70: Configuración de clock

Se reemplaza los datos en la ecuación (34) se tiene:

$$CP = \frac{72MHz}{20kHz} - 1 = 3599 \quad (35)$$

Para obtener una salida PWM incluyendo el canal negado igual se redondea el valor del Counter Period a 3600.

Se necesitan interrupciones para el correcto funcionamiento del automatismo del control discreto diseñado. Para ello, y tal como se muestra en la Figura N° 71, se utilizó la siguiente disposición de interrupciones:

- PA15-Eltra-Z: Se la dispuso de esta manera por si se llegaba a dar una utilidad en algún momento.
- PB10-FC1 y PB11-FC2: Interrumpe el programa si el carro llego al final de las varillas.
- PB12 – START: Da inicio al control del ángulo.

Pin Name	Signal on Pin	GPIO output...	GPIO mode
PA15	n/a	n/a	External Interrupt Mode with Rising edge trigger detection
PB10	n/a	n/a	External Interrupt Mode with Falling edge trigger detection
PB11	n/a	n/a	External Interrupt Mode with Falling edge trigger detection
PB12	n/a	n/a	External Interrupt Mode with Falling edge trigger detection
PC13-TAMP...	n/a	Low	Output Push Pull

Figura N° 71: Configuración de interrupciones

Por otra parte, se configuraron los dos timers restantes como interrupciones de la siguiente manera:

- TIMER 2: Este temporizador fue configurado para definir el tiempo de ejecución del control discreto a la frecuencia requerida, permitiendo así calcular la acción de control necesaria para corregir la desviación de ángulo presente en cada instante.

- **TIMER 4:** Se utilizó como medio para calcular el ángulo, incrementándose o decrementándose en función de las variaciones del ángulo obtenido por el encoder conforme transcurre el tiempo.

Finalmente, se configura la comunicación de la STM. La cual cuenta con dos tipos:

- **SYS:** Necesaria para cargar el programa al microcontrolador. Como se muestra en la Figura N° 72 se la configura en “Serial Wire”.

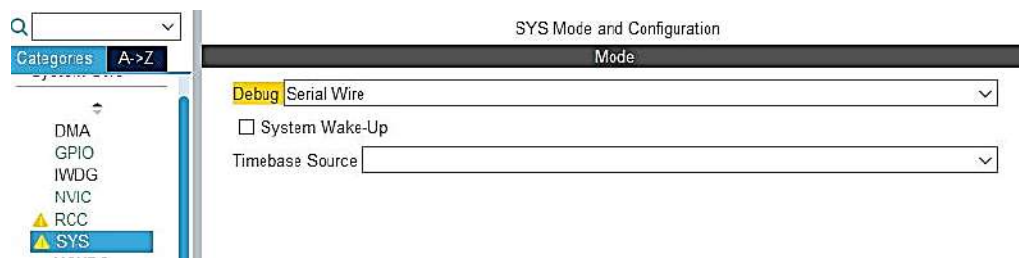


Figura N° 72: Configuración de SYS

- **USB:** Necesaria para ver los datos desde un monitor serie. Para ello se tilda la opción de “Device (FS)” como muestra la Figura N° 73.

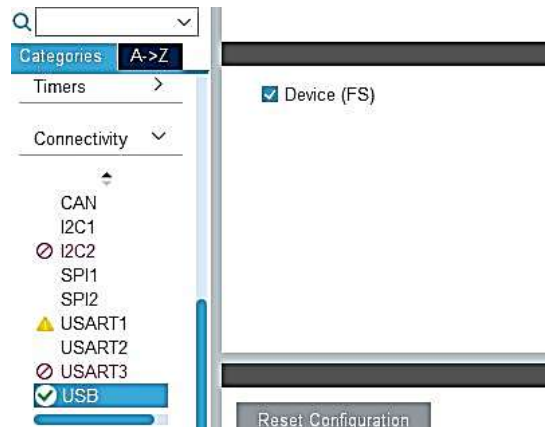


Figura N° 73: Configuración de USB

Haciendo estas configuraciones ya se puede proseguir a la codificación del control en el microcontrolador.

5.1. Código de STM32

Lo primero que se necesita realizar en el código es incluir las librerías necesarias para que el programa y la comunicación funcione de manera adecuada:

```
#include "main.h"
#include "usb_device.h"
```

Luego se declaran las variables del sistema las cuales son:

```
uint16_t CONTADOR = 0;           //CONTADOR DE PULSOS DE ENCODER ELTRA
float CONTADOR_ANG = 0;          //ANGULO REAL
float ANG= 0;                    //ANGULOS EN RADIANTES
float ek_1 = 0;                  //ERROR MUESTRA PASADA
int32_t uk_1 = 0;                //ACCIÓN DE CONTROL PASADA
float ek = 0;                    //ERROR ACTUAL
float u = 0;                     //ACCIÓN DE CONTROL ACTUAL
int Aref = 180;                  //ANGULO DE REFERENCIA
int porc = 0;                    //PORCENTAJE DE PWM
int PWM = 0;                     //PWM
float pi = 3.14159265359;
int sec = 0;                     //SECUENCIA
```

Se tiene:

- **CONTADOR:** Esta variable es necesaria para contar la cantidad de pulsos enviados por el encoder.
- **CONTADOR_ANG:** Este convierte el valor de los pulsos en un valor en ángulos.
- **ANG:** Necesario para el control, ya que el mismo utiliza los ángulos en radianes.
- **Aref:** Angulo de referencia. En este caso es 0.
- **ek:** Error actual entre el ángulo actual con el ángulo de referencia.
- **ek_1:** Muestra pasada del error.

- u: Acción de control actual.
- uk_1: Acción de control pasada.
- Porc: De la acción de control se estandarizo un máximo y un mínimo para sacar un porcentaje de este y poder calcular en base a esto la salida PWM.
- PWM: Control de velocidad y sentido de giro del motor de corriente continua.
- Sec: Se creo una secuencia para el manejo del automatismo.

Se procede al envío de datos vía USB para el monitoreo de las variables requeridas para la depuración de la acción de control. Aquí se tienen dos códigos:

1. Se programa la STM32 para el envío de datos. En este caso se necesitaba el ángulo para graficarlos en una hoja de cálculo Excel.

```
while (1)
{
    sprintf(TxBuf, "%.2f \n\r", ANG);
    CDC_Transmit_FS((uint8_t *)TxBuf, sizeof(TxBuf));
    HAL_Delay(500);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
```

2. Por otra parte, se tiene el código programado en el Visual Studio Code, que se encuentra en el Anexo B, para tener una lectura en tiempo real del ángulo y la creación de un archivo .xlsx para poder graficar los datos

A continuación, se detallan las tres partes más importantes del código del Anexo A:

1. **Interrupción del encoder para realizar el cálculo del ángulo:** Esta interrupción es utilizada cuando uno de los sensores del encoder detectó una rotación del eje de este y a partir de ello se realiza la suma o resta al contador de ángulo en uno.

Por otra parte, se multiplica ese valor por 25 ya que pulso por rotación significan un cuarto de grado.

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    CONTADOR = __HAL_TIM_GET_COUNTER(htim);
    CONTADOR_ANG = (int16_t)CONTADOR * 25;
}
```

2. **Cálculo y restricción de la acción de control:** Esta sección se encarga de codificar la acción de control de la ecuación (33) implementada en Matlab y hacer las restricciones necesarias para poder normalizar y escalar la acción de control previamente mencionada. Esto se realiza para tener una salida correspondiente a la necesidad de respuesta del sistema para controlar el ángulo. De manera que, normalizando el ángulo se lo puede escalar a un valor de salida PWM que tiene un margen mínimo de 0 a 3600 como máximo. Estos márgenes indican sentido y velocidad siendo los valores extremo superior e inferior las velocidades máximas que puede alcanzar el motor en sus dos sentidos de giro.

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim == &htim2)
    {

        ANG=Aref-(CONTADOR_ANG/100);
        ek=(ANG*pi)/(180);

        u=0.9*uk_1+6.0002e06*(ek-0.99995*ek_1);

        if (u < -75)
        {
            u = -75;
        }

        if (u > 75)
        {
            u = 75;
        }

        porc = round(((u+75)/150)*100);
        PWM = 36 * porc;
        TIM1 -> CCR1 = PWM;
        uk_1=u;
        ek_1=ek;
    }
}

```

3. **Interrupciones de finales de carrera e inicialización del PWM:** Esta sección de código es utilizada para los pulsadores de inicio y finales de carrera. Donde se establece por secuencia si el control debe empezar o terminar de actuar. Si es activado el pulsador de “START” la secuencia pasa a uno y se inicializa el PWM. Ahora bien, si estando en la secuencia 1 se detecta una interrupción de los finales de carrera la secuencia pasará a 0 y se detendrá la generación de la señal para el motor PWM.

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == START_Pin)
    {
        HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
        HAL_TIMEx_PWMN_Start(&htim1, TIM_CHANNEL_1);
        sec=1;
    }

    if(GPIO_Pin == FC1_Pin)
    {
        HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_13);
        if (sec == 1)
        {
            HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
            HAL_TIMEx_PWMN_Stop(&htim1, TIM_CHANNEL_1);
            sec = 0;
        }
    }
    if(GPIO_Pin == FC2_Pin)
    {
        if (sec == 1)
        {
            HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
            HAL_TIMEx_PWMN_Stop(&htim1, TIM_CHANNEL_1);
            sec = 0;
        }
    }
}

```

CAPITULO 6: Implementación

Antes de comenzar con la implementación del control primero se tuvo que imprimir las piezas necesarias diseñadas en SolidWorks para que el dispositivo funcione, para ello se utilizó una impresora 3D Ender-5 pro, ilustrada en la Figura N° 74. A través del software de diseño mencionado se guarda la pieza en un formato especial llamado “.stl” ¹ y desde otro programa llamado Creality Slicer se agrega la pieza, se configura los parámetros de impresión y se imprime. A continuación, se detalla los parámetros implementados:

1. Calidad:

- Espesor de capa: 0.15 *mm*
- Espesor de pared: 1.2 *mm*

2. Relleno:

- Espesor superior e inferior: 1.2
- Densidad de relleno: 10%

3. Velocidad:

- Velocidad de impresión: 50 *mm/s*
- Temperatura de impresión: 200 °C
- Temperatura de la cama: 60 °C

A partir de aquí, se procede a guardar la pieza desde el programa, como muestra la Figura N° 75, mediante una memoria SD y se procede a imprimir la pieza.

¹ stl (STereoLithography) es un formato de archivo informático de diseño asistido por computadora que define geometría de objetos 3D [27].



Figura N° 74: Impresora Ender Dragon 5 pro

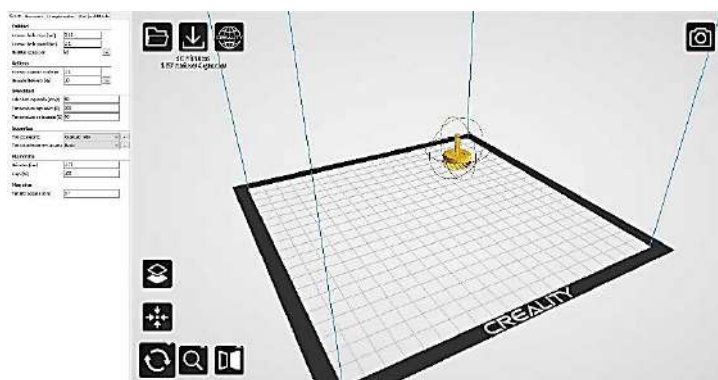


Figura N° 75: Impresión de rueda dentada motora en el software Creality Slider

Una vez impresas cada una de las piezas se procede a el armado del dispositivo como se mencionó en los pasos de la sección “**1.7. Armado de dispositivo**”. Luego de completarlos se puede observar las siguientes imágenes:

1. En la Figura N° 76 el ensamblaje del dispositivo ya con las conexiones de alimentación y monitoreo.



Figura N° 76: Foto de ensamble del dispositivo

2. La Figura N° 77 el sistema de transmisión de potencia motora y el final de carrera.

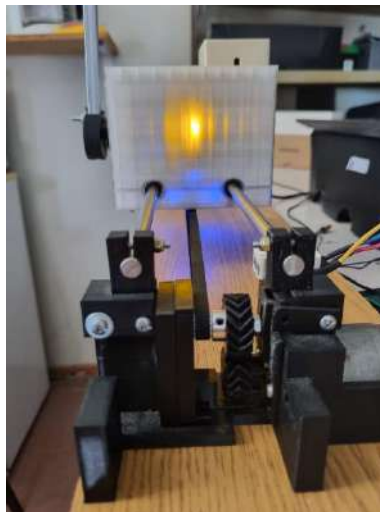


Figura N° 77: Foto de soporte de varillas motor

3. Del otro lado se encuentra el soporte de varillas con la polea conducida, capturado en la Figura N° 78, donde se puede apreciar la polea mencionada y a la izquierda de la imagen el soporte SK8 con la varilla y por debajo el otro final de carrera.



Figura N° 78: Foto de soporte de varillas conducido

4. Por otro lado, en la Figura N° 79 se encuentra el carro con el péndulo ya montado y en la cubierta de este la caja con el cable UTP conectado.



Figura N° 79: Foto de carro

5. Finalmente, se encuentra el ensamblaje de la placa electrónica conectada como se ilustra en la Figura N° 80. En esta imagen se puede apreciar que está indicado los bornes positivo y negativo donde se debe colocar la alimentación de $12\text{ V} - \text{CC}$ para evitar equivocaciones a la hora del conexionado.

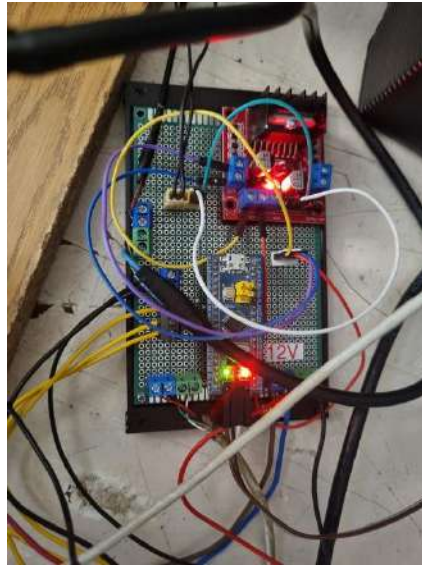


Figura N° 80: Foto de ensamblaje de placa electrónica.

A partir de lo anterior se procede a la implementación del control, el primer paso es descargar el código a la STM32 (Anexo A). Luego, con el péndulo en reposo, se conecta la alimentación a través del puerto USB y se gira el péndulo hasta alcanzar un ángulo aproximado de 0° . A partir de ese punto, se debe presionar el botón de “START” para que el controlador comience a operar.

En ese momento, se comenzó a controlar el ángulo, y mediante la ejecución de un código en Python (Anexo B), se pudieron recopilar datos para su posterior análisis. Después de realizar las pruebas, se obtuvo el gráfico de la Figura N° 81, donde se observó que el sistema respondió adecuadamente. Sin embargo, se identificó un leve inconveniente, ya que debido a la disposición de los sensores del encoder, el ángulo de 0° nunca se alineó con precisión sobre uno de estos, lo que provocó un error en estado estable. Además, al haber una mayor diferencia angular entre los sensores en un lado que en el otro, el péndulo tendió a caer siempre hacia el mismo lado. Este problema podría haberse mitigado aumentando la potencia del motor, lo que habría permitido compensar la diferencia entre los ángulos detectados por los sensores.

El ensayo se llevó a cabo durante un período de 372 segundos, en el cual, a los 340 segundos, se introdujo una perturbación significativa al péndulo, generando un gran ángulo de desviación. Como resultado, el carro alcanzó el final de carrera, lo que ocasionó que el control

dejara de actuar. Esto provocó que el péndulo cayera como se muestra en la Figura N° 82 y oscilara hasta finalmente llegar a su estado de reposo en 180° .



Figura N° 81: Grafico de prueba de control en un periodo de 372 segundos

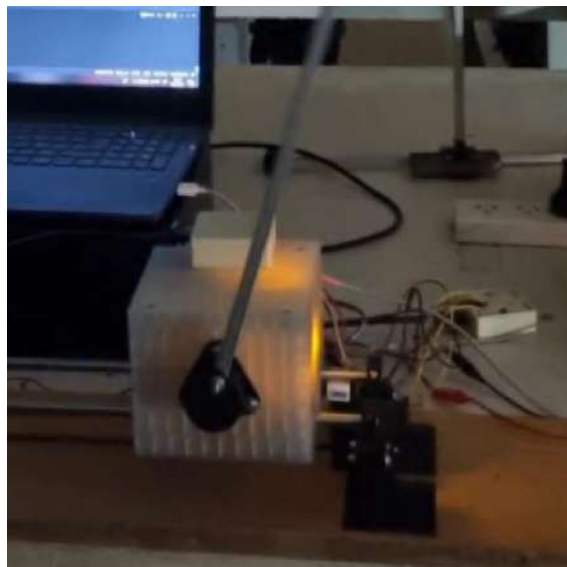


Figura N° 82: Actuación de final de carrera

Al realizar un acercamiento al gráfico de la Figura N° 81, se observó en la Figura N° 83 que la mayoría de los datos se encuentran en 0.25° , lo que indica que el sistema se mantuvo estable en esa posición. Esto se debe a que la acción de control no fue lo suficientemente fuerte como para superar la inercia del carro. Sin embargo, cuando los valores tomaron magnitudes negativas, fue necesario realizar una corrección física, ya que el sistema se volvió inestable bajo esas condiciones, como se mencionó anteriormente.

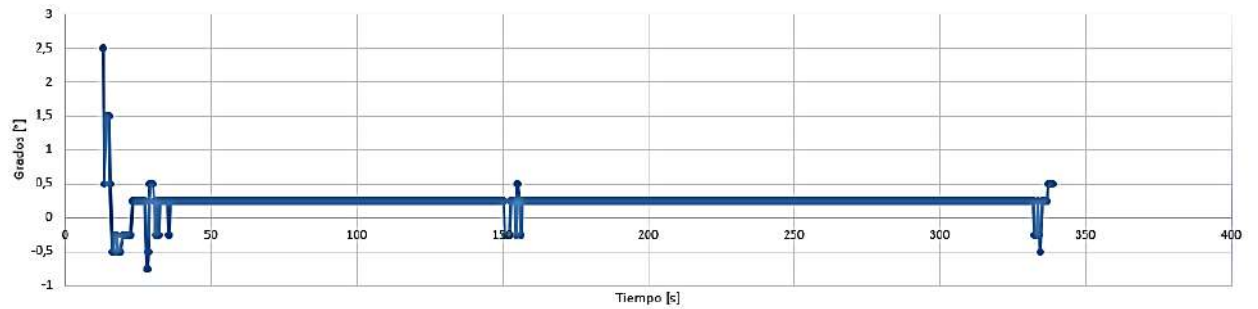


Figura N° 83: Grafico zoom de línea continua de Figura N° 63

A partir de este prototipo se llevaron a cabo las diversas pruebas para ver la efectividad del sistema, las cuales son:

- Ya que no se posee una hoja de datos del motor se procedió a realizar una prueba donde se suministró la tensión máxima a este y mediante fórmulas dinámicas se obtuvo la velocidad máxima y fuerza que el motor puede proporcionar. De estos ensayos se obtuvieron los siguientes resultados:
- $V_{max} = 7.44[km/h]$
- $F_{max} = 1.2 [N]$
- El peso del carro tiene un peso capaz de flexionar las varillas de apoyo del péndulo por lo cual desequilibran al propio péndulo.
- Se observa un cierto juego en los engranajes de transmisión de potencia hacia la correa del carro.

CAPITULO 7: Conclusión

En este trabajo se diseñó diseñar un péndulo invertido mediante la modelación de las partes en el software SolidWorks lo que permitió hacer la fabricación de estas en una impresora 3D abaratando costos en la construcción ya que el insumo utilizado para la fabricación es PLA que destaca por ser un material barato y que se encontraba disponible en la facultad. Además, se implementó un control del tipo adelanto de fase utilizando herramientas definidas en tiempo discreto y un microcontrolador como lo es la STM32. A través del diseño por Matlab del controlador, la modelación y la simulación, se evaluó la respuesta del sistema ante perturbaciones y se comprobó que, aunque existían limitaciones en la potencia del motor, el sistema podía estabilizar el péndulo en un ángulo cercano a 0° . Las pruebas demostraron la efectividad del controlador en la mayoría de los escenarios evaluados, aunque surgieron desafíos relacionados con la respuesta ante variaciones paramétricas como lo fueron la longitud del péndulo y la masa de este. Pero a pesar de esto el proyecto presento una alta viabilidad con los parámetros definidos y ante un desperfecto en alguna de las piezas impresas, las mismas pueden ser reemplazadas con facilidad ya que la mayoría de las piezas son modulares.

7.1. Trabajos a futuro

Se propone como trabajo a futuro:

1. Implementación de un nuevo motor de mayor potencia para que el sistema pueda corregir el ángulo monitoreado en todo momento y en cualquier condición. Para ello, se debe realizar un nuevo diseño 3D para las nuevas dimensiones de este.
2. Eliminación de juego entre las piezas mediante la modificación la disposición y dimensiones de las piezas intervinientes en los mecanismos de transmisión de potencia en el ensamblaje.
3. Fabricación 3D de cobertura de placa electrónica.
4. Pruebas con distinto materiales de impresión para encontrar el más apto para cada uno de los elementos intervinientes en los mecanismos del proyecto.
5. Implementación de diversos tipos de controles para probar su efectividad.
6. Creación de una interfaz gráfica para monitoreo y modificación del control sin la necesidad de modificar el código en el microcontrolador.

CAPITULO 8: Bibliografía

- [1] R. Dorf y R. Bishop, Modern Control Systems, Prentice Hall, 2016.
- [2] inin4010clavado, 17 12 2008. [En línea]. Available: https://www.youtube.com/watch?app=desktop&v=f4kOPxpRGrk&NR=1&ab_channel=inin4010clavado.
- [3] K. Furuta, M. Yamakita y S. Kobayashi, «Swing-up control of inverted pendulum using pseudo-state feedback,» *IEEJ Journal of Industry Applications*, vol. 6, nº 1, pp. 55-61, 1992.
- [4] O. Gonzalez, «BricoGeek,» 30 06 2023. [En línea]. Available: <https://blog.bricogeek.com/noticias/robotica/espectacular-pendulo-invertido-de-escritorio/>.
- [5] B. Wie, Space Vehicle Dynamics and Control.
- [6] GUI_UNO, 9 4 2020. [En línea]. Available: https://www.youtube.com/watch?v=PgXYd8fkJEo&ab_channel=GUI_UNO.
- [7] K. Furuta y K. Åström, «Swinging up a pendulum by energy control,» *Automatica*, vol. 36, nº 2, pp. 287-295, 2000.
- [8] I. Postlethwaite y S. Skogestad, Multivariable Feedback Control: Analysis and Design, 2005.
- [9] K. Ogata, Ingeniería de control moderna, Madrid: Pearson, 2010.
- [10] G. Franklin y M. Workman, Digital Control of Dynamic Systems, Addison-Wesley, 1997.
- [11] K. Ogata, Sistemas de control en tiempo discreto, México: Pearson.
- [12] M. Redondo, «hipertextual,» 29 11 2017. [En línea]. Available: <https://hipertextual.com/2017/11/nasa-utilizara-falcon-9-spacex-carga-espacial>.
- [13] «XiaoRGeek,» [En línea]. Available: <https://www.xiaorgeek.net/es/products/xiaor-geek-stm32-self-balancing-robot-car>.
- [14] «Bühler,» [En línea]. Available: <https://www.directindustry.es/prod/buehler-motor-gmbh/product-7076-1883955.html>.
- [15] «Inw,» [En línea]. Available: <https://www.tap-part.com/article/129/eltra-encoder-italy-3>.
- [16] I. S. I. «Mercado Libre,» [En línea]. Available: https://articulo.mercadolibre.com.ar/MLA-809797110-modulo-l298n-doble-puente-h-driver-l298n-motor-dc-arm-avr-l298-arduino-_JM.
- [17] «ST,» [En línea]. Available: https://www.st.com/content/st_com/en.html.
- [18] «Ubuy,» [En línea]. Available: <https://www.ubuy.com.ar/sp/product/48LWXHL6C-5pcs-stm32f103c8t6-io-io-arm-stm32-32-cortex-m3-swd-minimum-system-development-board-module-mini-usb-interface-for-arduino>.
- [19] A. Valera, M. Valles y M. Cardo. [En línea]. Available: <https://intranet.ceautomatica.es/old/actividades/jornadas/XXIV/documentos/ro/77.pdf>.
- [20] I. A. Duarte Olivares. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/126347/Duarte%20-%20Dise%C3%B1o%20implementaci%C3%B3n%20y%20control%20de%20un>

%20p%C3%A9ndulo%20invertido%20con%20rueda%20de%20reacci%C3%B3n..pdf
?sequence=2.

- [21] F. Castaños Luna. [En línea]. Available:
<https://www.ctrl.cinvestav.mx/~fcastanos/mios/bachelorCastanos.pdf>.
- [22] «Pololu,» [En línea]. Available: <https://www.pololu.com/category/12/rc-servo-controllers>.
- [23] «MOTESA Automatismos,» [En línea]. Available:
<https://www.motesa.com/product/encoder-eltra/>.
- [24] E. A. C. M. A. Parra, «Jovenes en la ciencia,» 22 02 2018. [En línea]. Available:
<https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/2294>.
- [25] J. Garcia, L. Ramirez, X. Siordia y T. Martinez, «Ecorfan,» 12 2016. [En línea]. Available:
https://www.ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol3num9/Revista_Tecnologia_e_Innovacion_V3_N9_16_2.pdf.
- [26] L. G. T. Marcias, «UAB,» 09 2020. [En línea].
- [27] 14 11 2023. [En línea]. Available: <https://es.wikipedia.org/wiki/STL>.

CAPITULO 9: Anexos

9.1. Anexo A – Código implementado en stm32

```
/* USER CODE BEGIN Header */
/**

*****
 * @file           : main.c
 * @brief          : Main program body
*****

 * @attention
 *
 * Copyright (c) 2024 STMicroelectronics.
 * All rights reserved.
 *
 * This software is licensed under terms that can be found in the LICENSE
file
 * in the root directory of this software component.
 * If no LICENSE file comes with this software, it is provided AS-IS.
*

*****
 */
/* USER CODE END Header */
/* Includes -----
*/
#include "main.h"
#include "usb_device.h"

/* Private includes -----
*/
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----
*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */
```

```

/* Private macro -----
*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----
*/
TIM_HandleTypeDef htim1;
TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim4;

/* USER CODE BEGIN PV */
char TxBuf[84];
/* USER CODE END PV */

/* Private function prototypes -----
*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM1_Init(void);
static void MX_TIM4_Init(void);
static void MX_TIM2_Init(void);
/* USER CODE BEGIN PFP */
extern uint8_t CDC_Transmit_FS(uint8_t* Buf, uint16_t Len);
/* USER CODE END PFP */

/* Private user code -----
*/
/* USER CODE BEGIN 0 */
uint16_t CONTADOR = 0; //CONTADOR DE PULSOS DE ENCODER
ELTRA
float CONTADOR_ANG = 0; //ANGULO REAL
float ANG= 0; //ANGULOS EN RADIANES
float ek_1 = 0; //ERROR MUESTRA PASADA
int32_t uk_1 = 0; //ACCIÓN DE CONTROL
PASADA
float ek = 0; //ERROR ACTUAL
float u = 0; //ACCIÓN DE CONTROL
ACTUAL
int Aref = 180; //ANGULO DE REFERENCIA
int porc = 0; //PORCENTAJE DE PWM
int PWM = 0; //PWM
float pi = 3.14159265359;
int sec = 0; //SECUENCIA
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int

```

```

*/
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuración-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_TIM1_Init();
    MX_USB_DEVICE_Init();
    MX_TIM4_Init();
    MX_TIM2_Init();
    /* USER CODE BEGIN 2 */
    HAL_TIM_Encoder_Start_IT(&htim4, TIM_CHANNEL_ALL);
    HAL_TIM_Base_Start_IT(&htim2);
    HAL_TIM_Base_Start(&htim2);
    HAL_TIM_Base_Start(&htim1);
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        sprintf(TxBuf, "%.2f \n\r", ANG);
        CDC_Transmit_FS((uint8_t *)TxBuf, sizeof(TxBuf));
        HAL_Delay(500);
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
    }
    /* USER CODE END 3 */
}

```

```

}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /** Initializes the CPU, AHB and APB buses clocks
     */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                   |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
    {
        Error_Handler();
    }
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USB;
    PeriphClkInit.UsbClockSelection = RCC_USBCLKSOURCE_PLL_DIV1_5;
    if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    {
        Error_Handler();
    }
}

/**
 * @brief TIM1 Initialization Function

```

```

    * @param None
    * @retval None
    */
static void MX_TIM1_Init(void)
{
    /* USER CODE BEGIN TIM1_Init 0 */

    /* USER CODE END TIM1_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};
    TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig = {0};

    /* USER CODE BEGIN TIM1_Init 1 */

    /* USER CODE END TIM1_Init 1 */
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 0;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 3600;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_TIM_PWM_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 0;
    sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
    sConfigOC.OCNPolarity = TIM_OCNPOLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;

```

```

sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_DISABLE;
sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_DISABLE;
sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
sBreakDeadTimeConfig.DeadTime = 0;
sBreakDeadTimeConfig.BreakState = TIM_BREAK_DISABLE;
sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
sBreakDeadTimeConfigAutomaticOutput = TIM_AUTOMATICOUTPUT_DISABLE;
if (HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM1_Init 2 */

/* USER CODE END TIM1_Init 2 */
HAL_TIM_MspPostInit(&htim1);

}

/**
 * @brief TIM2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM2_Init(void)
{
    /* USER CODE BEGIN TIM2_Init 0 */

    /* USER CODE END TIM2_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM2_Init 1 */

    /* USER CODE END TIM2_Init 1 */
    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 0;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 65535;
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
    {
        Error_Handler();
    }

```

```

    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM2_Init 2 */

    /* USER CODE END TIM2_Init 2 */

}

/**
 * @brief TIM4 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM4_Init(void)
{
    /* USER CODE BEGIN TIM4_Init 0 */

    /* USER CODE END TIM4_Init 0 */

    TIM_Encoder_InitTypeDef sConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM4_Init 1 */

    /* USER CODE END TIM4_Init 1 */
    htim4.Instance = TIM4;
    htim4.Init.Prescaler = 0;
    htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim4.Init.Period = 65535;
    htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim4.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    sConfig.EncoderMode = TIM_ENCODERMODE_TI12;
    sConfig.IC1Polarity = TIM_ICPOLARITY_FALLING;
    sConfig.IC1Selection = TIM_ICSELECTION_DIRECTTI;
    sConfig.IC1Prescaler = TIM_ICPSC_DIV1;
    sConfig.IC1Filter = 0;
    sConfig.IC2Polarity = TIM_ICPOLARITY_FALLING;
    sConfig.IC2Selection = TIM_ICSELECTION_DIRECTTI;
    sConfig.IC2Prescaler = TIM_ICPSC_DIV1;

```

```

sConfig.IC2Filter = 0;
if (HAL_TIM_Encoder_Init(&htim4, &sConfig) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM4_Init 2 */

/* USER CODE END TIM4_Init 2 */

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);

    /*Configure GPIO pin : PC13 */
    GPIO_InitStruct.Pin = GPIO_PIN_13;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : FC1_Pin FC2_Pin START_Pin */
    GPIO_InitStruct.Pin = FC1_Pin|FC2_Pin|START_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
    GPIO_InitStruct.Pull = GPIO_PULLDOWN;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    /*Configure GPIO pin : ELTRA_Z_Pin */
    GPIO_InitStruct.Pin = ELTRA_Z_Pin;

```



```

GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(ELTRA_Z_GPIO_Port, &GPIO_InitStruct);

/* EXTI interrupt init*/
HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);

}

/* USER CODE BEGIN 4 */
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    CONTADOR = __HAL_TIM_GET_COUNTER(htim);
    CONTADOR_ANG = (int16_t)CONTADOR * 25;
}

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim == &htim2)
    {
        ANG=Aref-(CONTADOR_ANG/100);
        ek=(ANG*pi)/(180);

        u=0.9*uk_1+6.0002e06*(ek-0.99995*ek_1);

        if (u < -75)
        {
            u = -75;
        }

        if (u > 75)
        {
            u = 75;
        }

        porc = round(((u+75)/150)*100);
        PWM = 36 * porc;
        TIM1 -> CCR1 = PWM;
        uk_2=uk_1;
        uk_1=u;
        ek_2=ek_1;
        ek_1=ek;
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{

```

```

    if(GPIO_Pin == START_Pin)
    {
        HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
        HAL_TIMEx_PWMN_Start(&htim1, TIM_CHANNEL_1);
        sec=1;
    }

    if(GPIO_Pin == FC1_Pin)
    {
        HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_13);
        if (sec == 1)
        {
            HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
            HAL_TIMEx_PWMN_Stop(&htim1, TIM_CHANNEL_1);
            sec = 0;
        }
    }
    if(GPIO_Pin == FC2_Pin)
    {
        if (sec == 1)
        {
            HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
            HAL_TIMEx_PWMN_Stop(&htim1, TIM_CHANNEL_1);
            sec = 0;
        }
    }
}

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.

```

```

    * @param file: pointer to the source file name
    * @param line: assert_param error line source number
    * @retval None
    */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line
    number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
    */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```

9.2. Anexo B – Código de Python para comunicación serial

```
import serial
import time
import pandas as pd

puerto = 'COM8'
bit_rate = 9600
output_file = 'C:/Users/jmxro/Desktop/STM/datos_serie.xlsx' # Define la ruta
y nombre del archivo Excel

try:
    ser = serial.Serial(puerto, bit_rate)
    print(f"Se abrió el puerto {puerto} correctamente.")
except Exception as e:
    print(f"No se pudo abrir el puerto {puerto}: {e}")
    exit()

# Lista para almacenar los datos (tiempo, ángulo)
datos = []

try:
    start_time = time.time()
    buffer = "" # Buffer para acumular los datos del ángulo

    while True:
        data = ser.read().decode('utf-8') # Lee un carácter y lo decodifica
        if data.isdigit() or data == '.' or data == '-': # Verifica si el
carácter es un dígito, un punto o un signo menos
            buffer += data # Acumula el carácter en el buffer
        elif data == '\n': # Si llega el carácter de nueva línea (fin del
dato)
            if buffer: # Si el buffer no está vacío
                try:
                    angulo = float(buffer.strip()) # Convierte el buffer a
flotante
                    tiempo = time.time() - start_time # Calcula el tiempo
transcurrido
                    datos.append((tiempo, angulo)) # Guarda el tiempo y el
ángulo en la lista
                    print(f"Tiempo: {tiempo}, Ángulo: {angulo}")
                except ValueError:
                    print(f"Error al convertir {buffer} a flotante")
                buffer = "" # Resetea el buffer para el siguiente dato

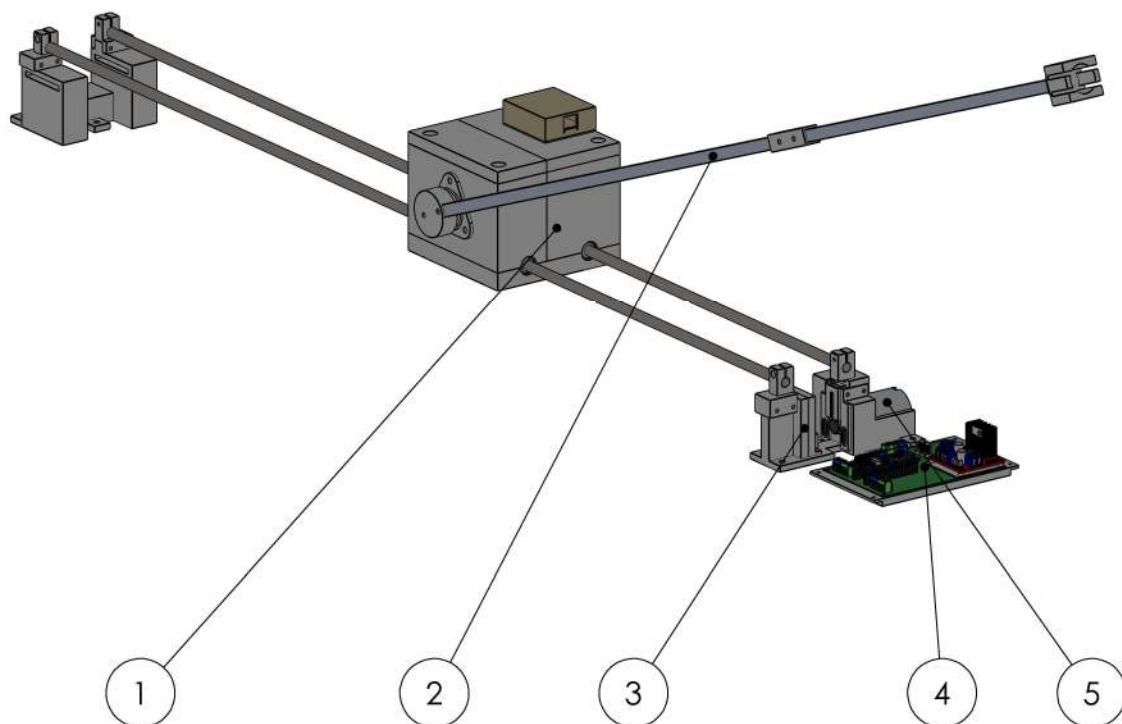
except KeyboardInterrupt:
    ser.close()
    print("\nSe cerró el puerto serial.")
```

```
# Convertir la lista de datos a un DataFrame de pandas
df = pd.DataFrame(datos, columns=['Tiempo', 'Ángulo'])

# Guardar el DataFrame en un archivo Excel
df.to_excel(output_file, index=False)
print(f"Datos guardados en {output_file}")
```

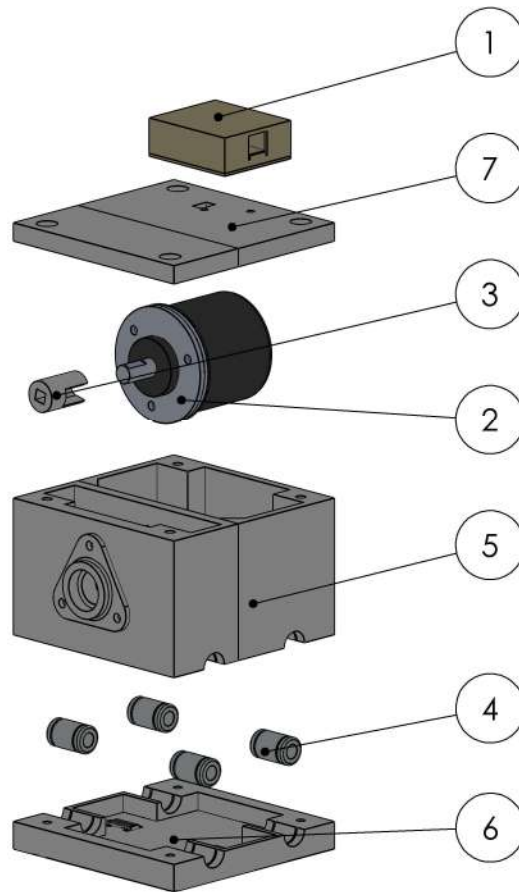
9.3. Anexo C – Planos componentes

A continuación, se adjuntan la representación gráfica de los componentes intervinientes en este trabajo.



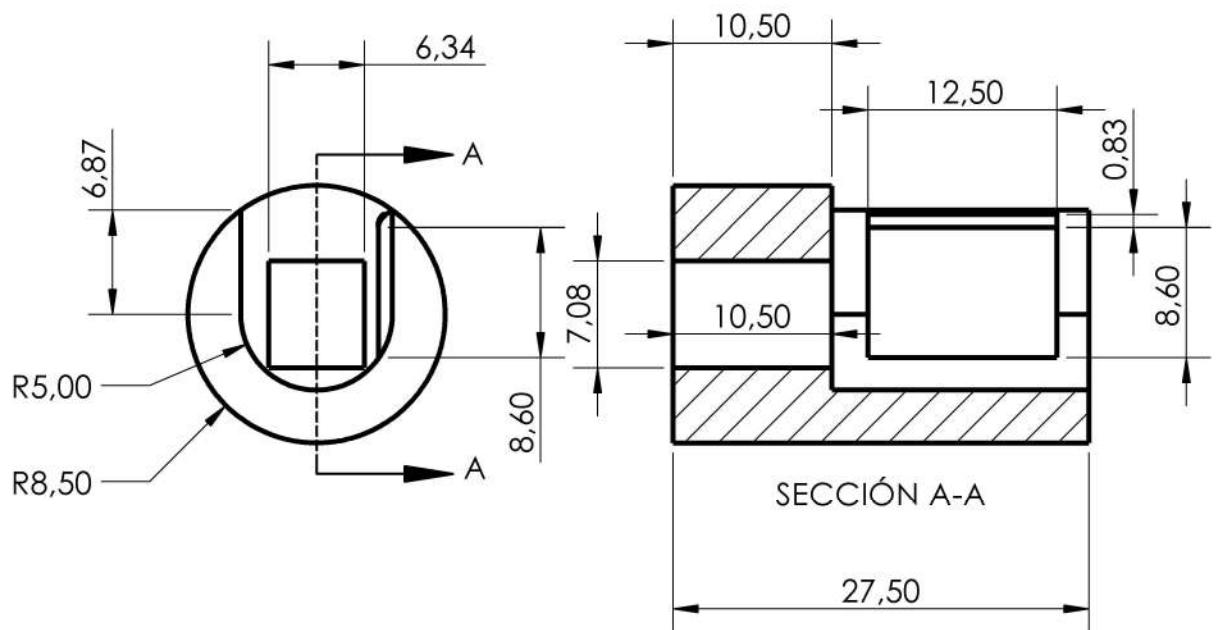
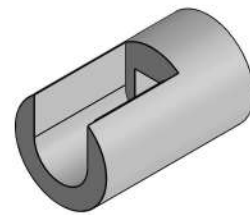
5	1	Motor					CC - Buheler
4	1	Placa electrónica	01-04-00				Ensamblaje
3	1	Varillas soportes	01-03-00				Ensamblaje
2	1	Péndulo	01-02-00				Ensamblaje
1	1	Carro	01-01-00				Ensamblaje
Pos.	Cant.	Denominacion	Nº plano	Nº de pieza	Material	Masa (gr)	Observaciones

	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:6	PENDULO INVERTIDO LINEAL			Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
	Formato A4				01-00-00	
					1/25	

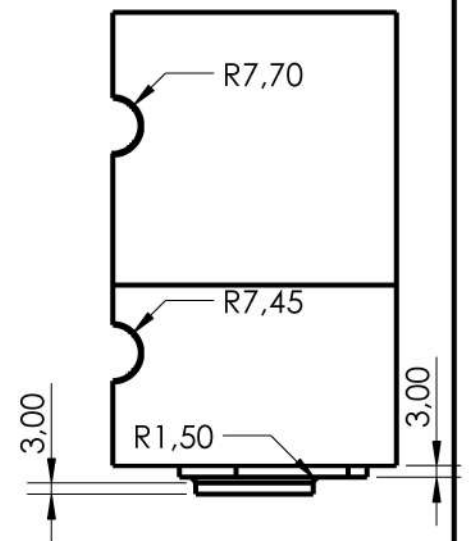
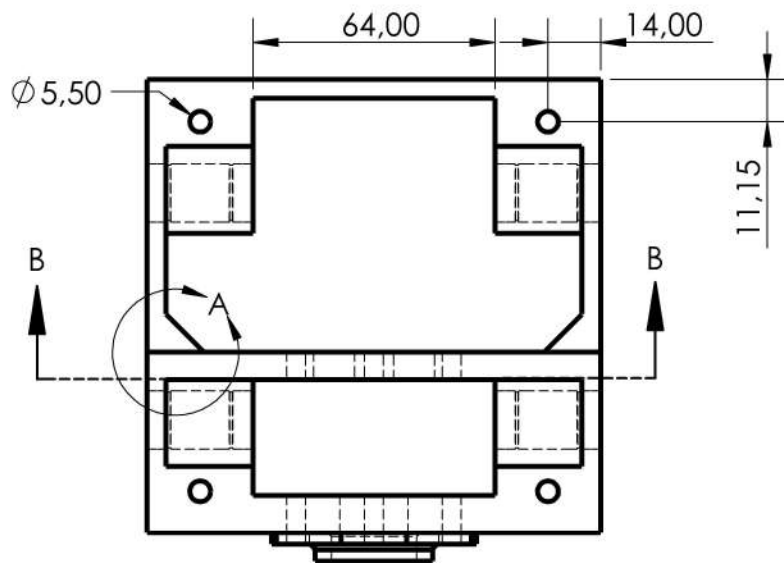
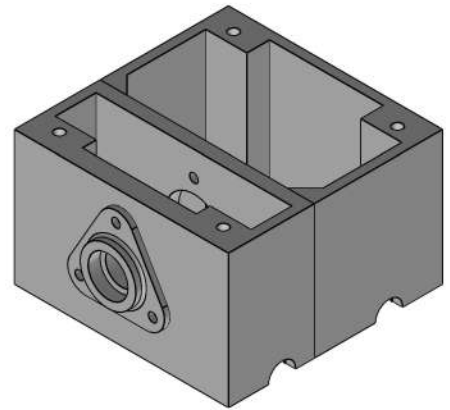
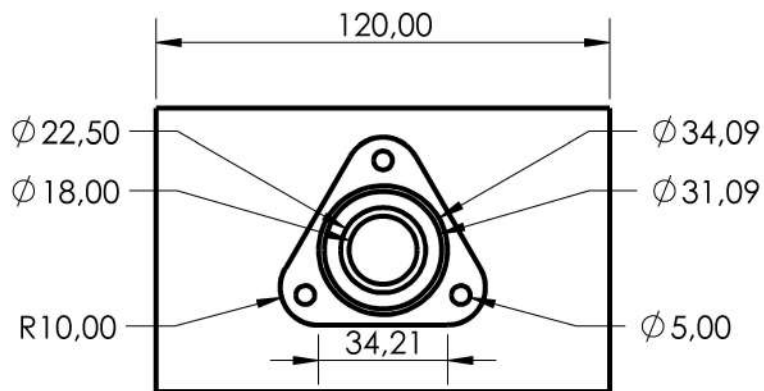


7	1	Cubierta carro	01-01-04	7	PLA		
6	1	Base carro	01-01-03	6	PLA		
5	1	Cuerpo carro	01-01-02	5	PLA	469.08	
4	4	Rodamiento lineal		4			LM8UU
3	1	Eje encoder	01-01-01	3	PLA	4.99	
2	1	Encoder		2			Eltra ER 63
1	1	Caja rj45		1			Genérico
Pos.	Cant.	Denominacion	Nº plano	Nº de pieza	Material	Masa (gr)	Observaciones

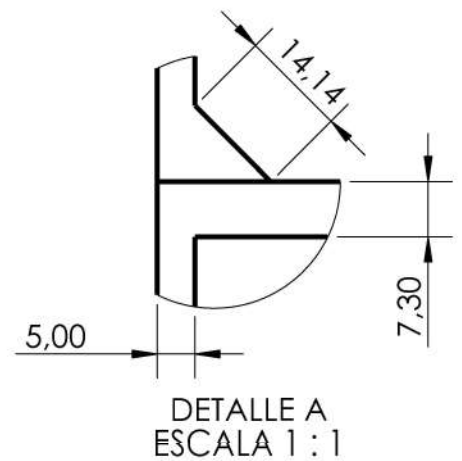
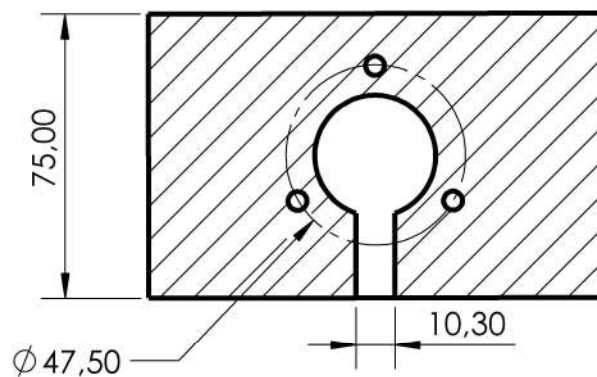
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:4	CARRO			Diseño, implementación y control de un péndulo invertido lineal	
	Trabajo Final de Grado					
Formato A4	01-01-00					
	2/25					



	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 2:1	EJE ENCODER			Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
	Formato A4				01-01-01	
					3/25	



SECCIÓN B-B



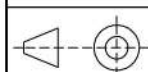
DETALLE A
ESCALA 1 : 1

Proyectó	26/08/2024	J.M.ROMERO
Dibujó	26/08/2024	J.M.ROMERO
Revisó	27/08/2024	M.F.BERGOGLIO
Aprobó		

UNIVERSIDAD
NACIONAL DE
SAN LUIS

Facultad de
Ingeniería y Ciencias
Agropecuarias

Esc.
1:2



Formato
A4

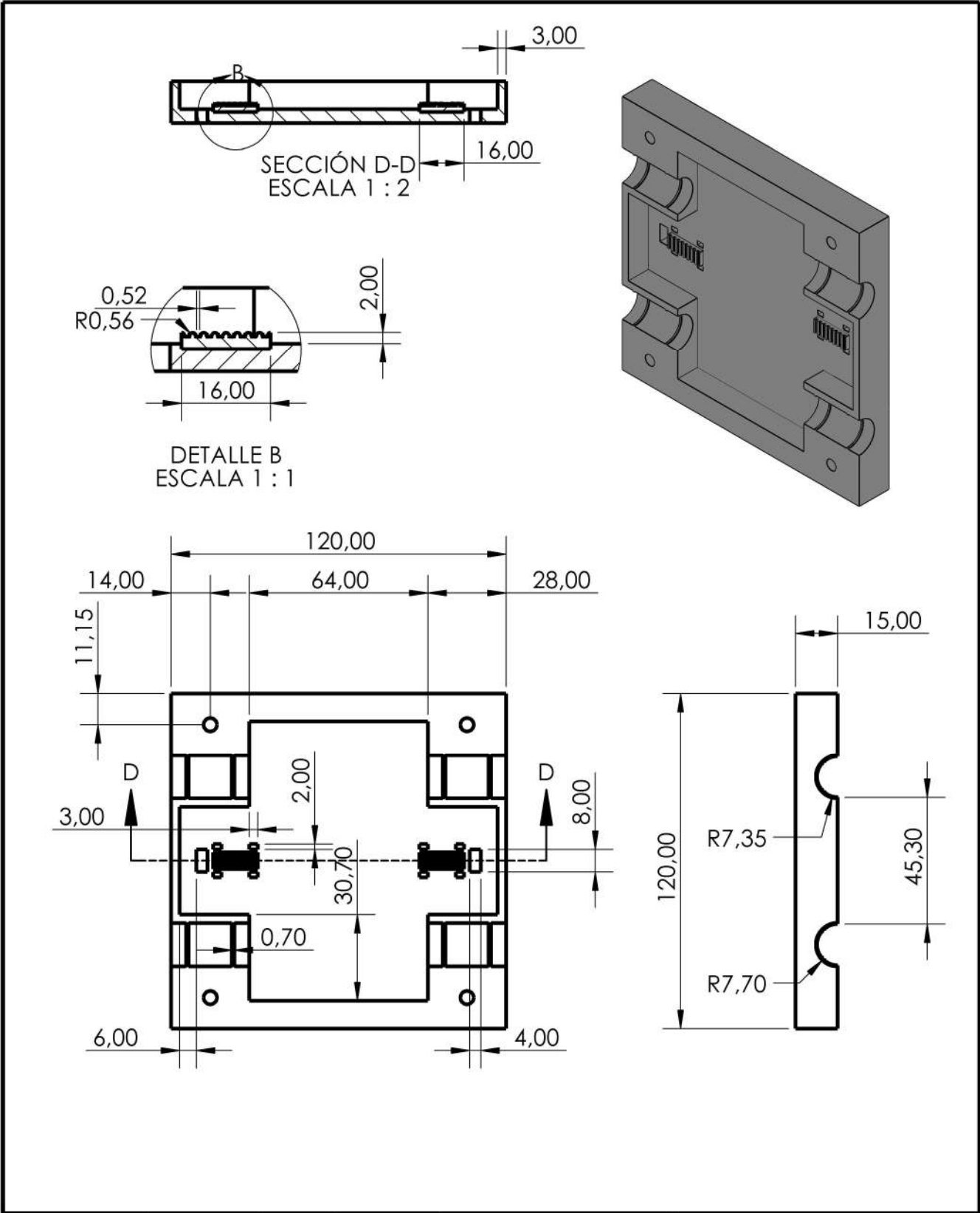
CUERPO CARRO

Diseño, implementación y
control de un péndulo
invertido lineal

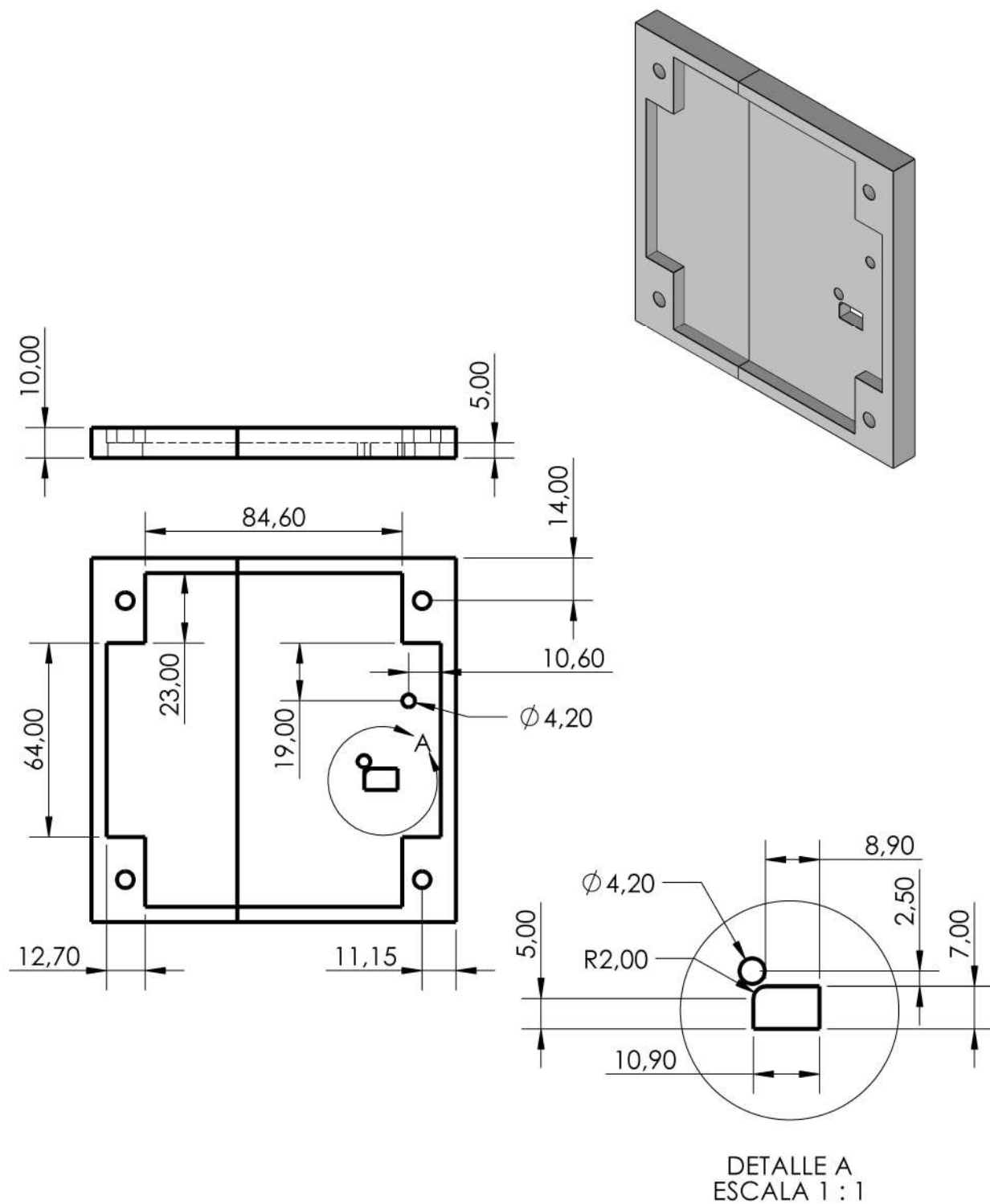
Trabajo Final de Grado

01-01-02

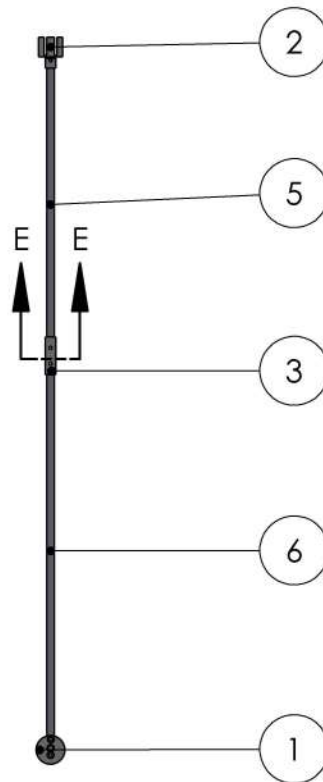
4/25



	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
Esc. 1:2	BASE CARRO				Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
Formato A4					01-01-03	



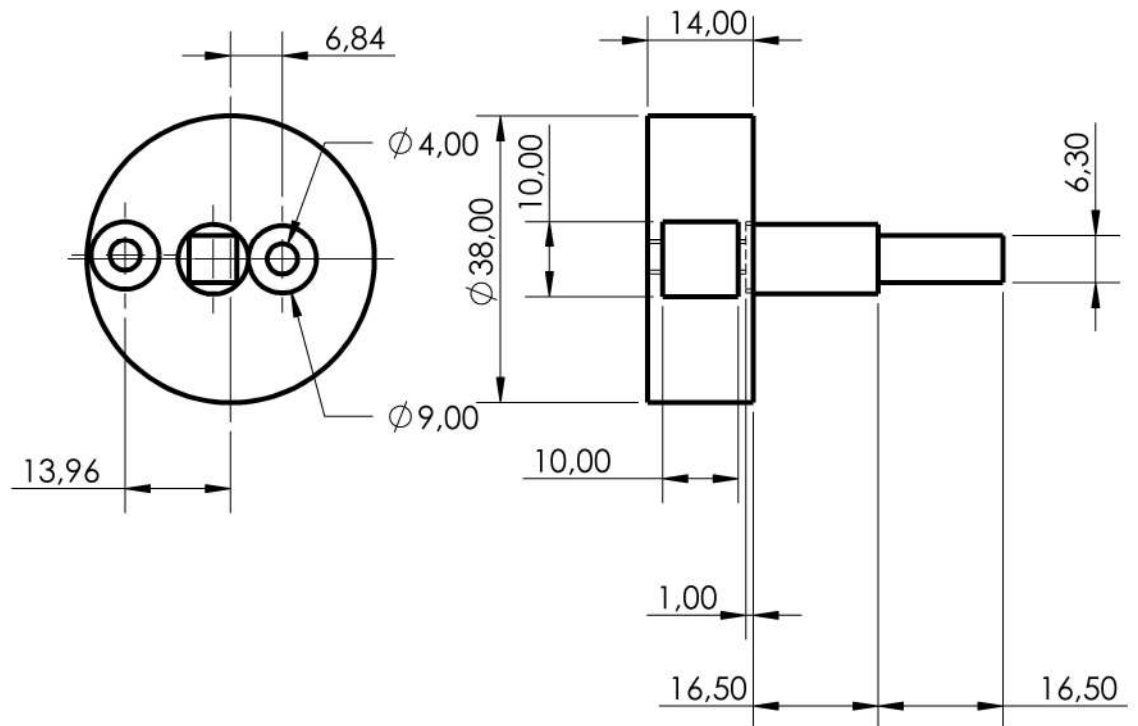
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
<div>Esc. 1:2</div> <div></div> <div>Formato A4</div>	CUBIERTA CARRO				Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
					01-01-04	



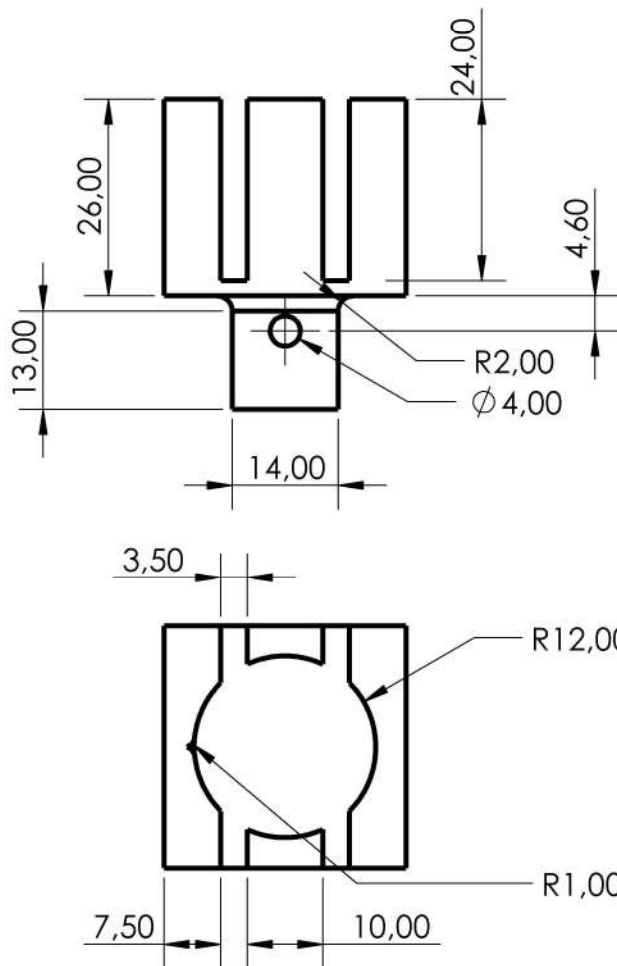
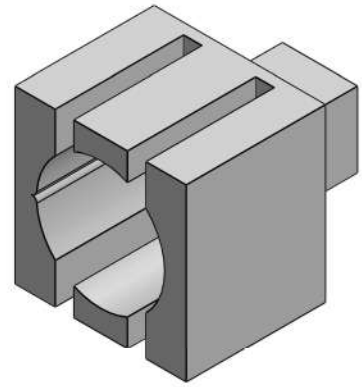
SECCIÓN E-E
ESCALA 1 : 4

6	1	Varilla 2	01-02-04	13	Aluminio		
5	1	Varilla 1	01-02-04	12	Aluminio		
4	1	Unión varillas	01-02-03	11	PLA	2.68	
3	1	Unión varillas exterior	01-02-03	10	PLA	6.11	
2	1	Soporte pesas	01-02-02	9	PLA	19.76	
1	1	Eje péndulo	01-02-01	8	PLA	17.80	
Pos.	Cant.	Denominacion	N° plano	N° de pieza	Material	Masa (gr)	Observaciones

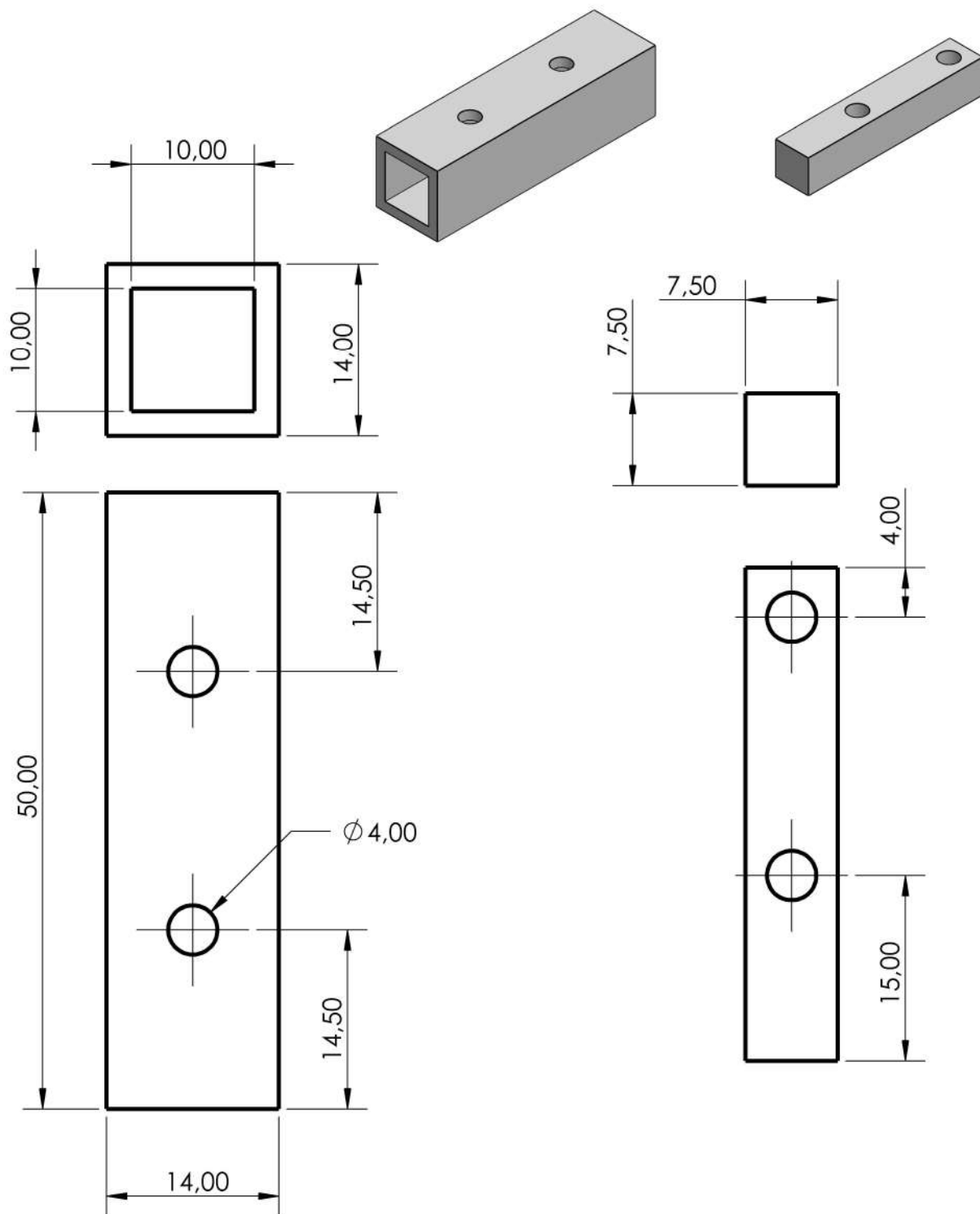
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias		
	Dibujó	26/08/2024	J.M.ROMERO				
	Revisó	27/08/2024	M.F.BERGOGLIO				
	Aprobó						
Esc. 1:10		PÉNDULO			Diseño, implementación y control de un péndulo invertido lineal		
					Trabajo Final de Grado		
Formato A4					01-02-00		7/25



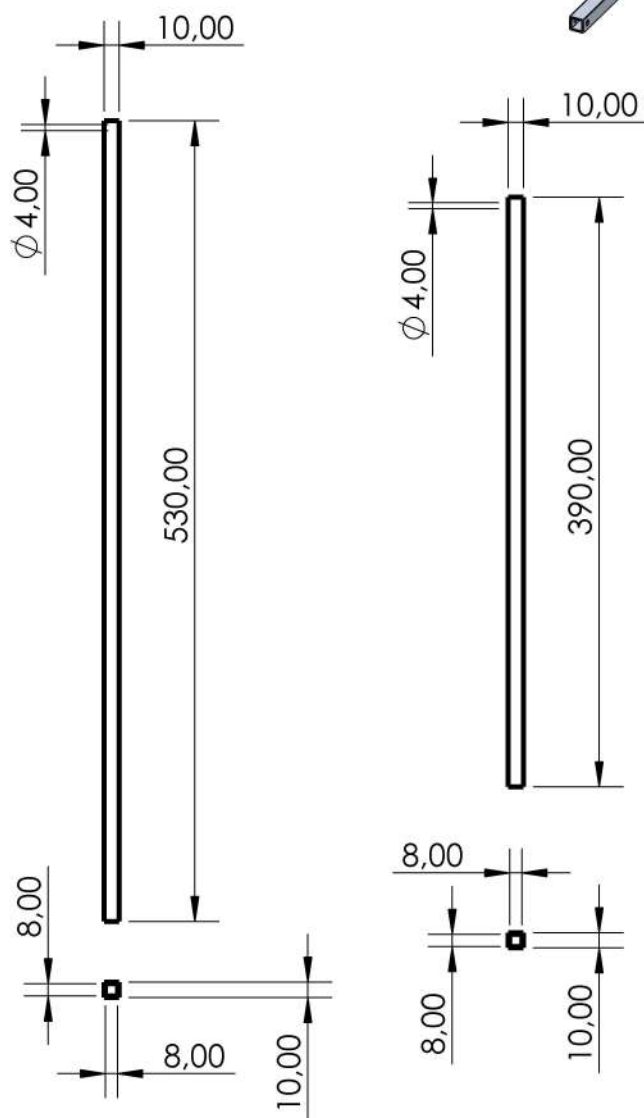
Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
Dibujó	26/08/2024	J.M.ROMERO		Diseño, implementación y control de un péndulo invertido lineal	
Revisó	27/08/2024	M.F.BERGOGLIO			
Aprobó					
Esc. 1:1	EJE PÉNDULO			Trabajo Final de Grado	
				01-02-01	
Formato A4				8/25	



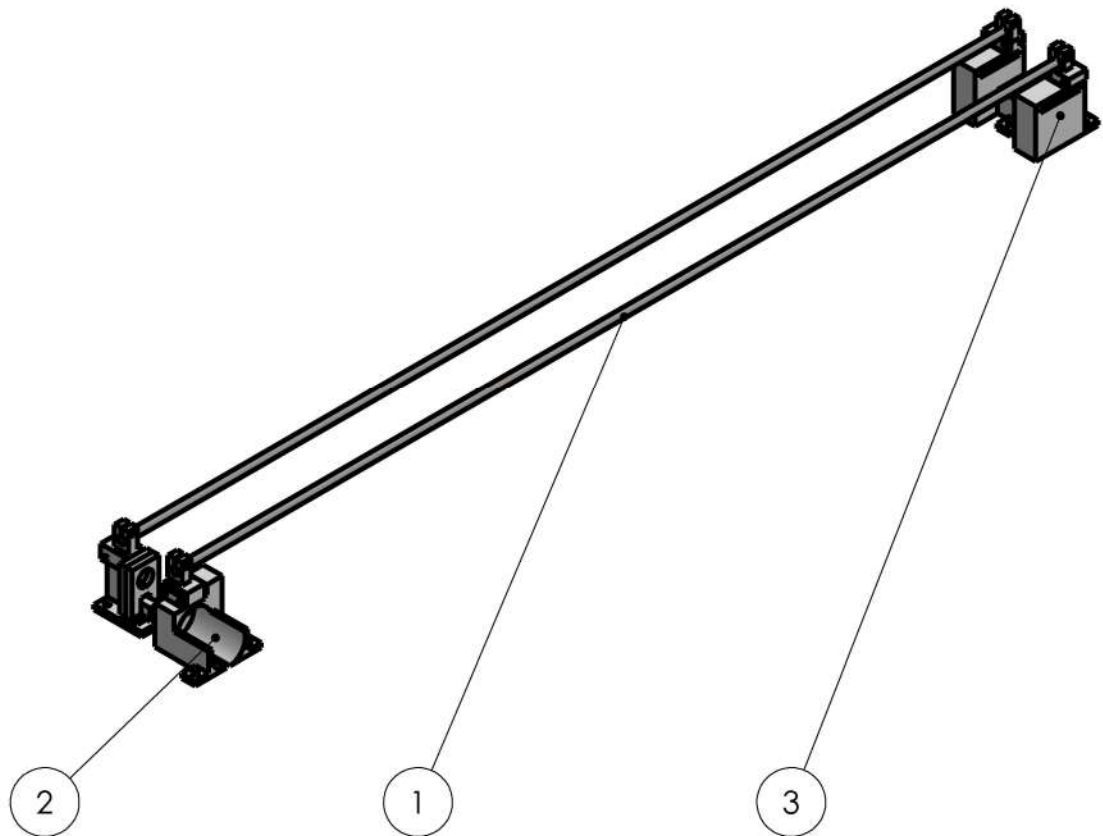
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:1	SOPORTE PESAS			Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
	Formato A4				01-02-02	
					9/25	



	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingenieria y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO		Diseño, implementación y control de un péndulo invertido lineal	
	Aprobó					
	Esc. 2:1	UNIONES VARILLAS EXTERIOR E INTERIOR			Trabajo Final de Grado	
						
Formato A4	01-02-03				10/25	

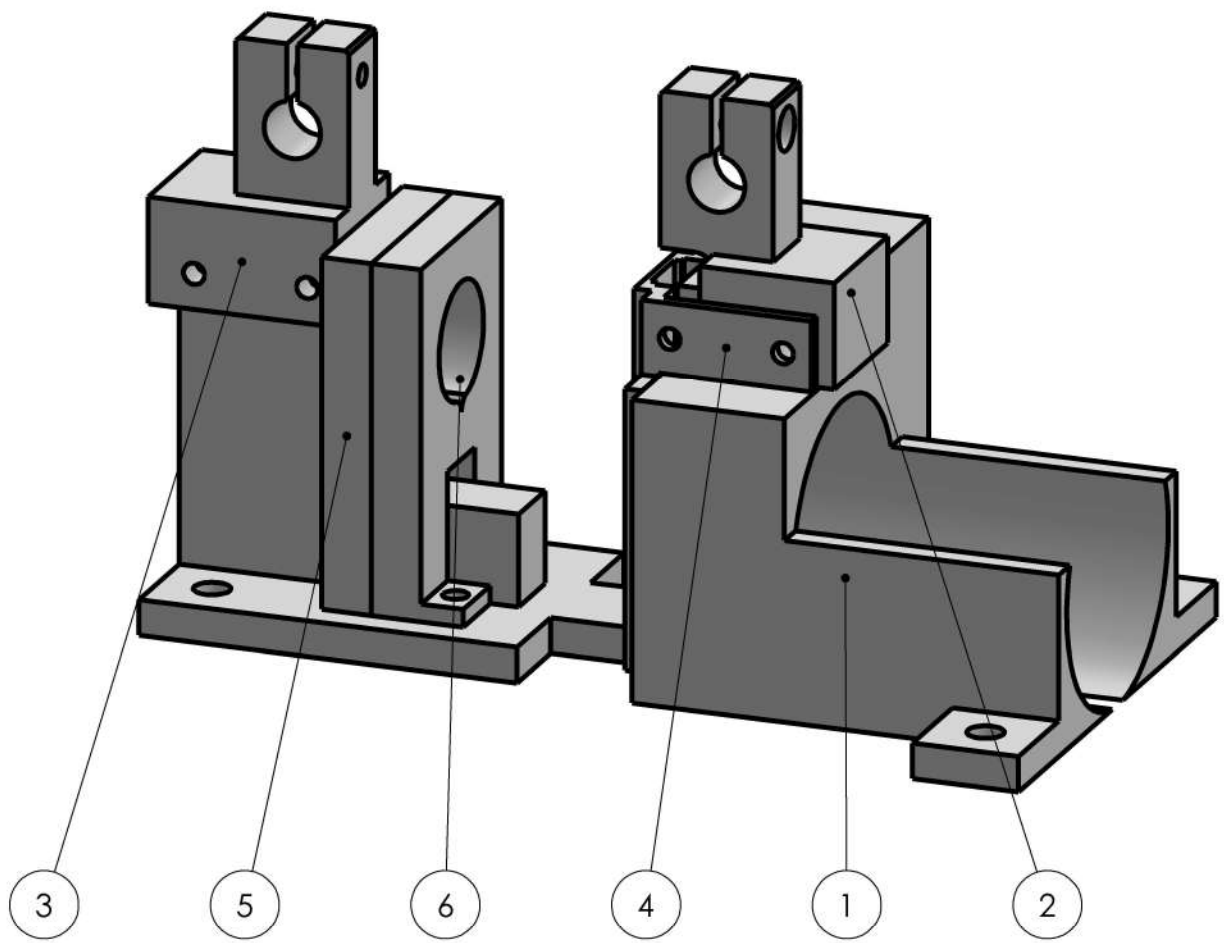


	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:5	VARILLAS			Diseño, implementación y control de un péndulo invertido lineal	
	Trabajo Final de Grado					
Formato A4	01-02-04					
	11/25					



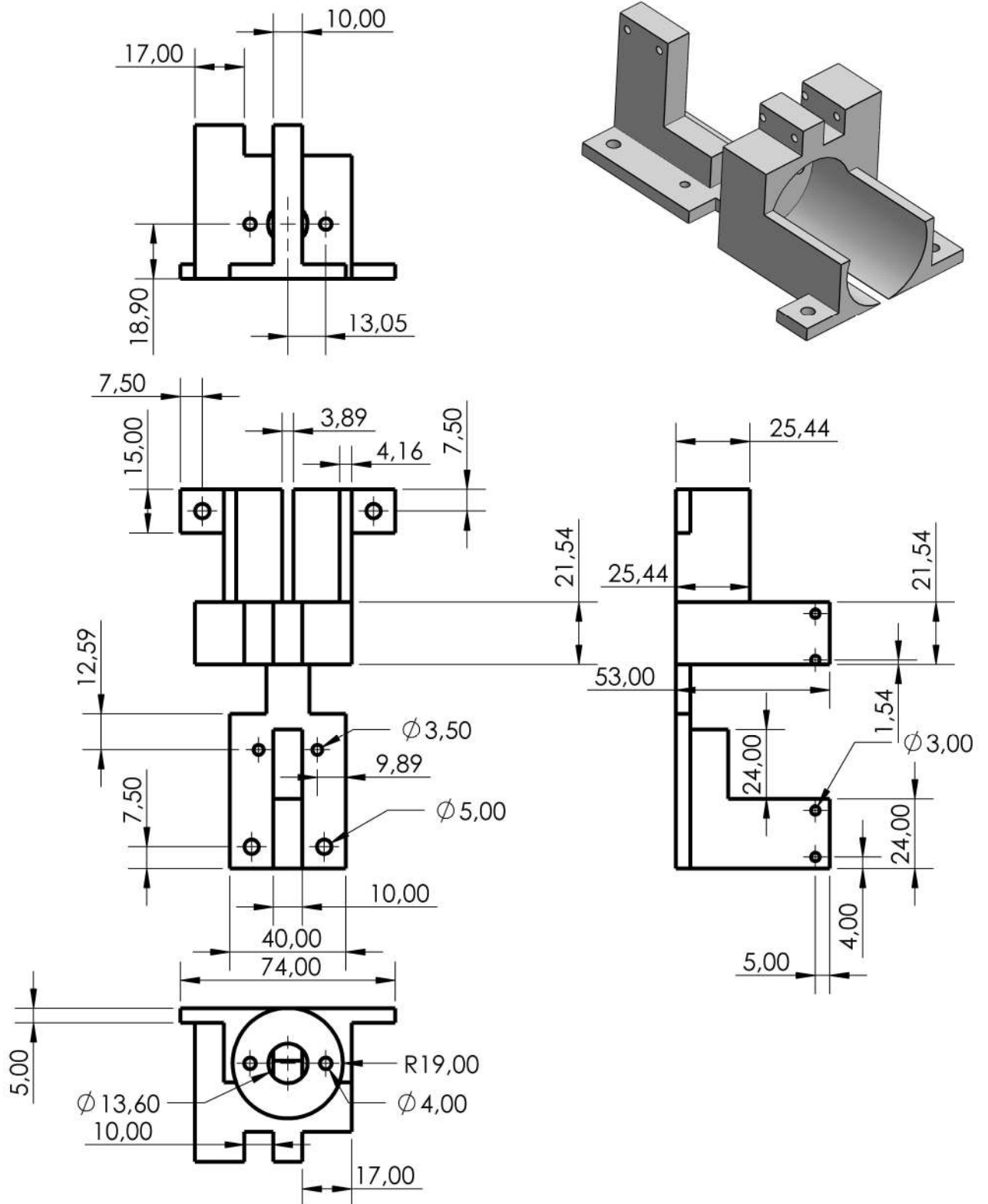
3	1	Soporte polea	01-03-20			229.60	Ensamblaje
2	1	Soporte motor	01-03-10			142.06	Ensamblaje
1	2	Varilla soporte		14	Acero rectificado templado		
Pos.	Cant.	Denominacion	N° plano	N° de pieza	Material	Masa (gr)	Observaciones

	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:6	SOPORTES			Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
	Formato A4				01-03-00	
					12/25	

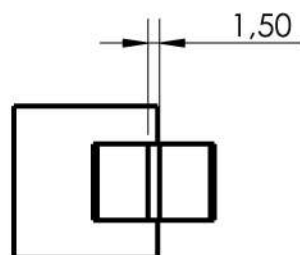
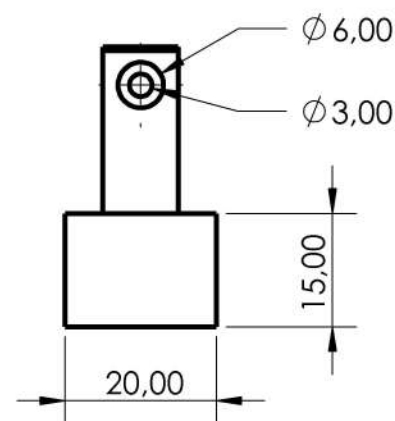
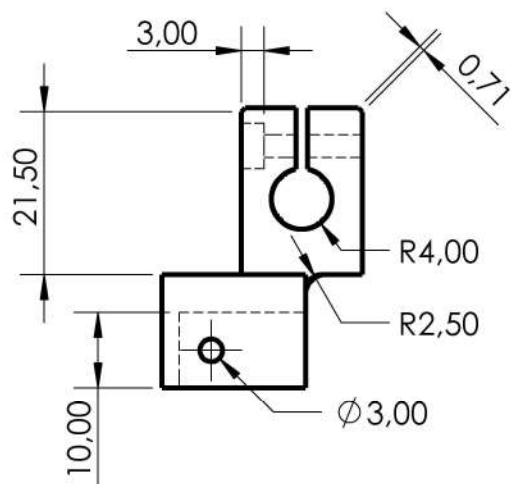
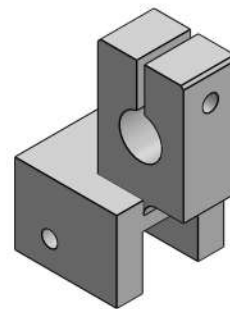
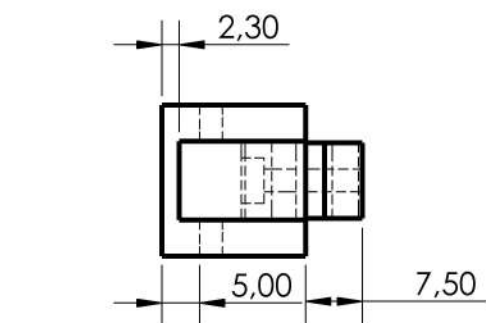


6	1	Soporte eje polea motora	01-03-15	20	PLA	10.58	
5	1	Suplemento soporte eje polea motora	01-03-15	19	PLA	11.95	
4	1	Soporte rodamiento	01-03-14	18	PLA	4.57	
3	1	sk8 1	01-03-13	17	PLA	10.43	
2	1	sk8 1 motor	01-03-12	16	PLA	8.63	
1	1	Paredes	01-03-11	15	PLA	95.89	
Pos.	Cant.	Denominacion	N° plano	N° de pieza	Material	Masa (gr)	Observaciones

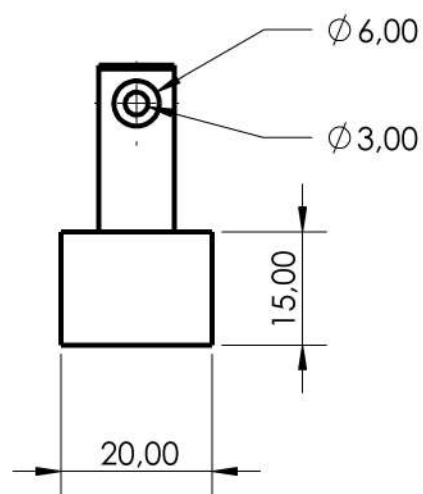
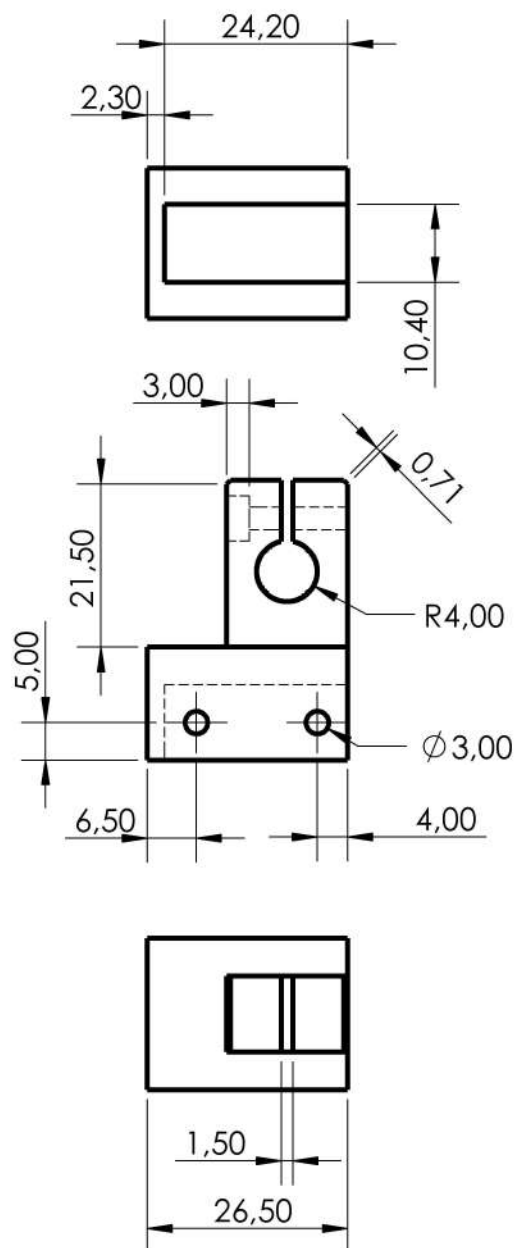
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingenieria y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:1	SOPORTE MOTOR			Diseño, implementación y control de un péndulo invertido lineal	
	Trabajo Final de Grado					
Formato A4	01-03-10				13/25	



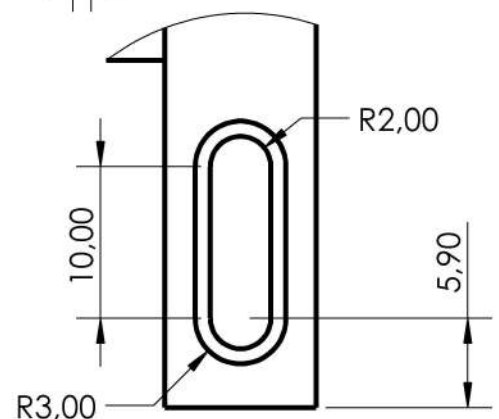
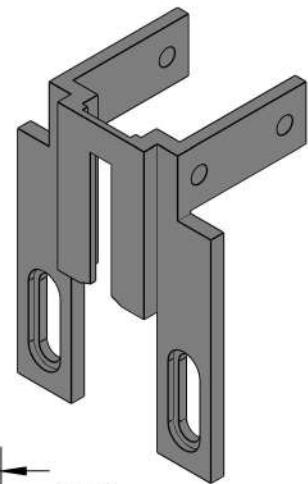
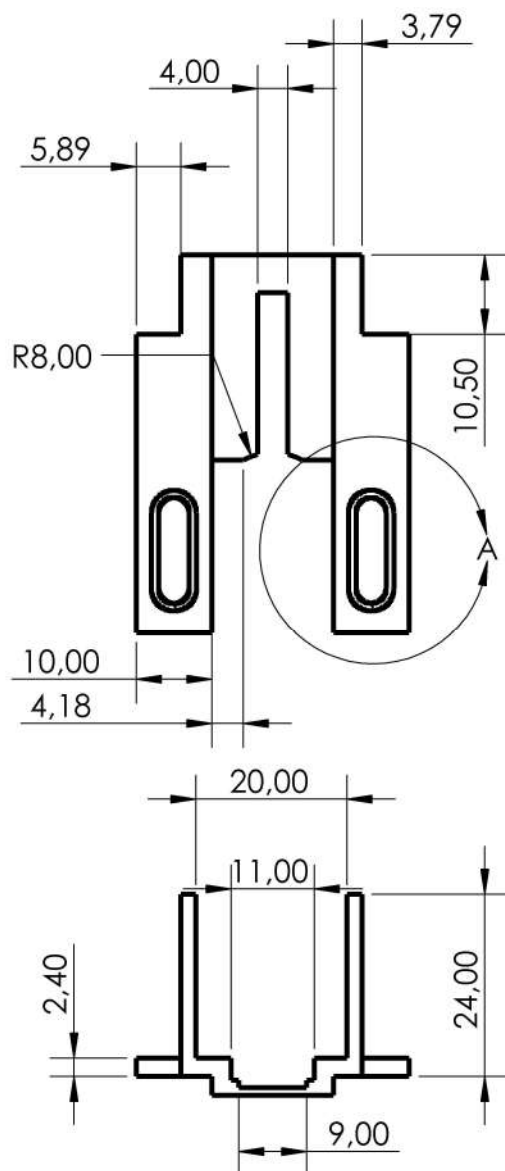
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO		Diseño, implementación y control de un péndulo invertido lineal	
	Aprobó					
Esc. 1:2		PAREDES			Trabajo Final de Grado	
						
Formato A4						
					01-03-11	



	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
Esc. 1:1	SK8 MOTOR				Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
Formato A4					01-03-12	
					15/25	

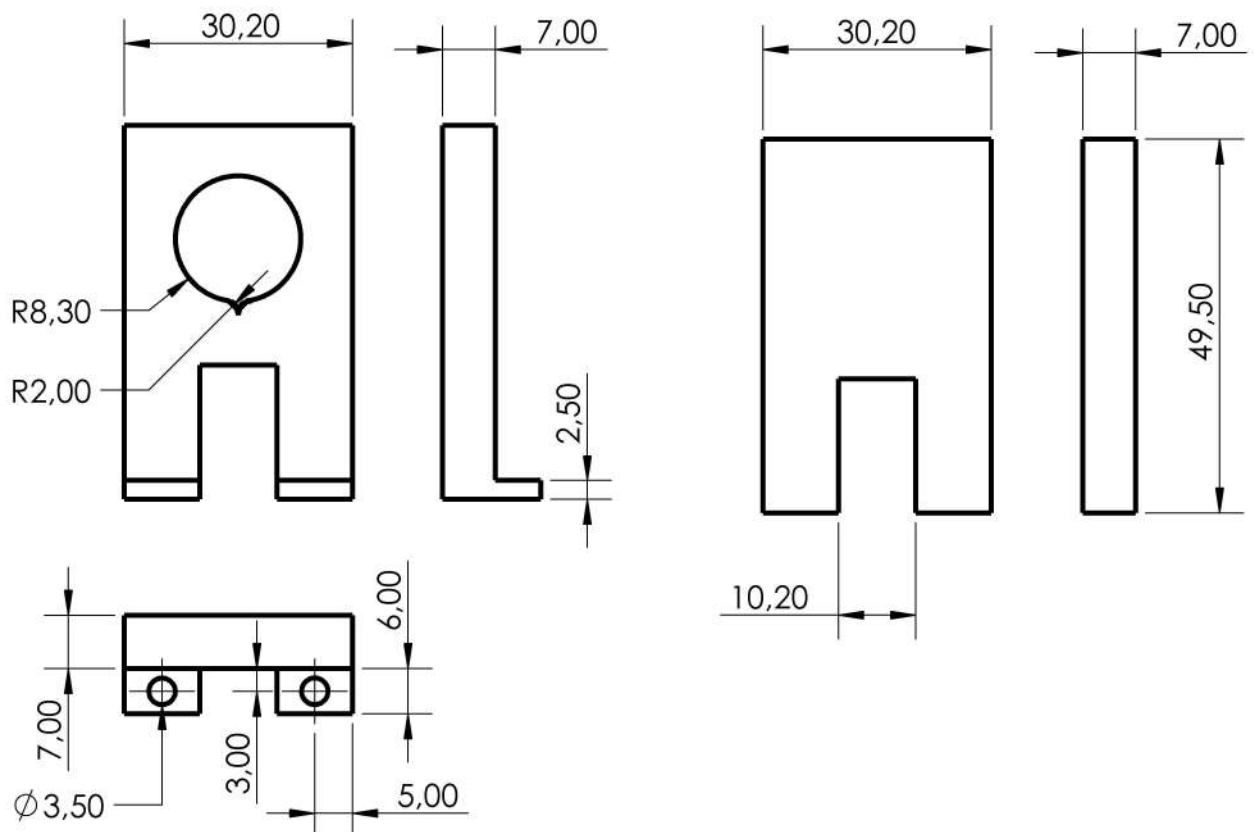
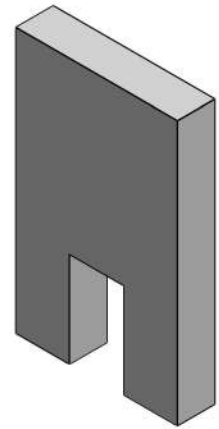
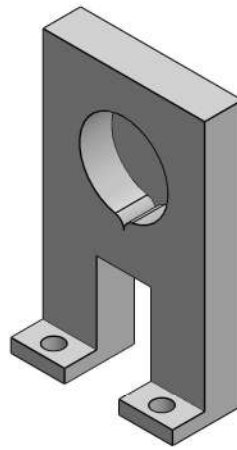


Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias		
Dibujó	26/08/2024	J.M.ROMERO		Diseño, implementación y control de un péndulo invertido lineal		
Revisó	27/08/2024	M.F.BERGOGLIO				
Aprobó						
Esc. 1:1	SK8 1			Trabajo Final de Grado		
				01-03-13		16/25
Formato A4						

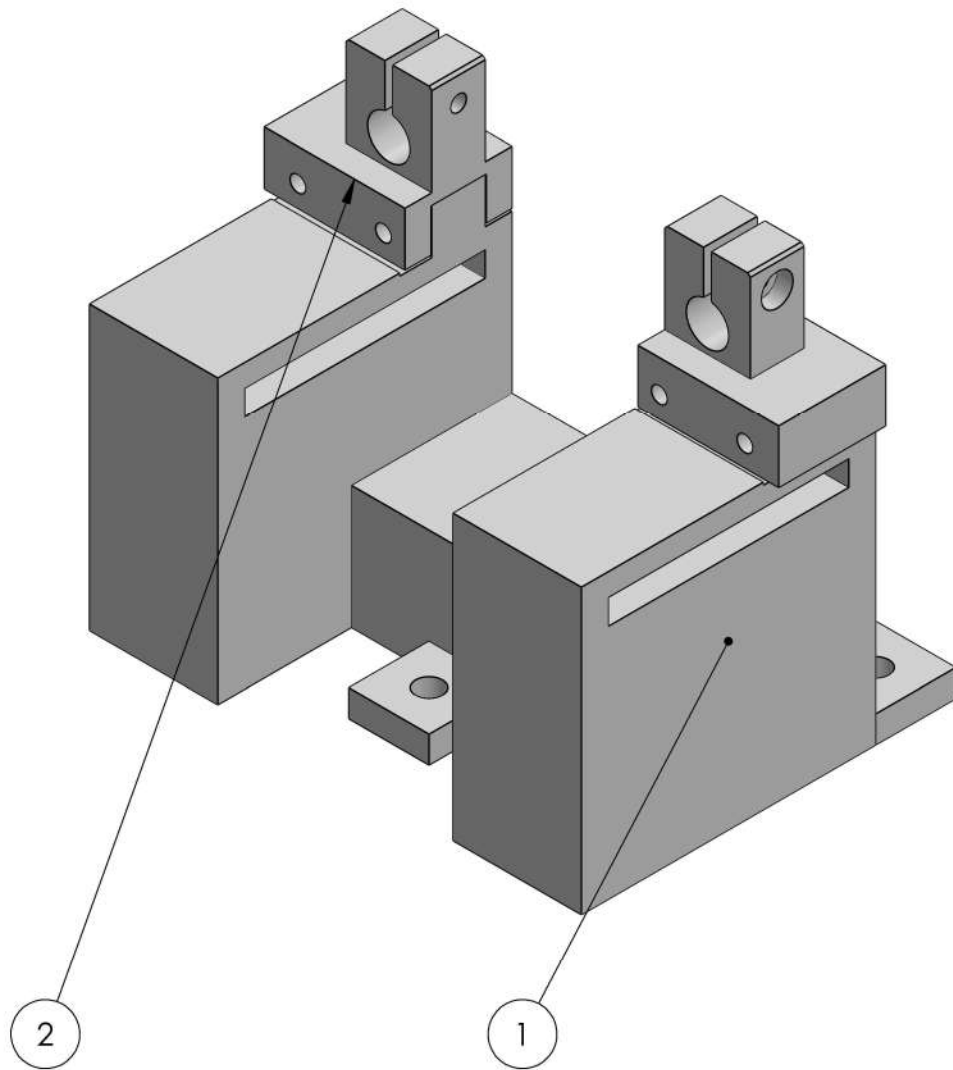


DETALLE A
ESCALA 2 : 1

	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingenieria y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:1	SOPORTE RODAMIENTO MOTOR			Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
	Formato A4				01-03-14	
					17/25	

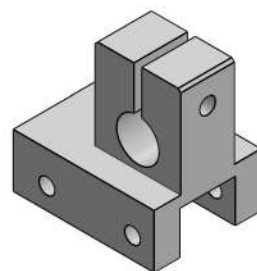
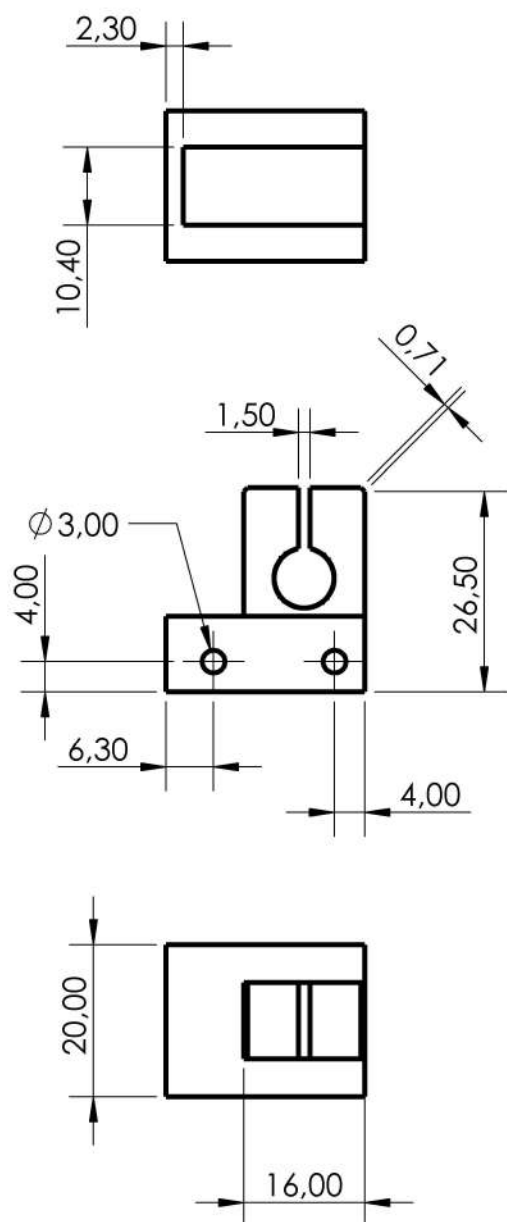


	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGGIO			
	Aprobó					
	Esc. 1:1	SOORTE EJE RODAMIENTO MOTOR Y SUPLEMENTO			Diseño, implementación y control de un péndulo invertido lineal	
						
	Formato A4					
	Trabajo Final de Grado					
	01-03-15		18/25			

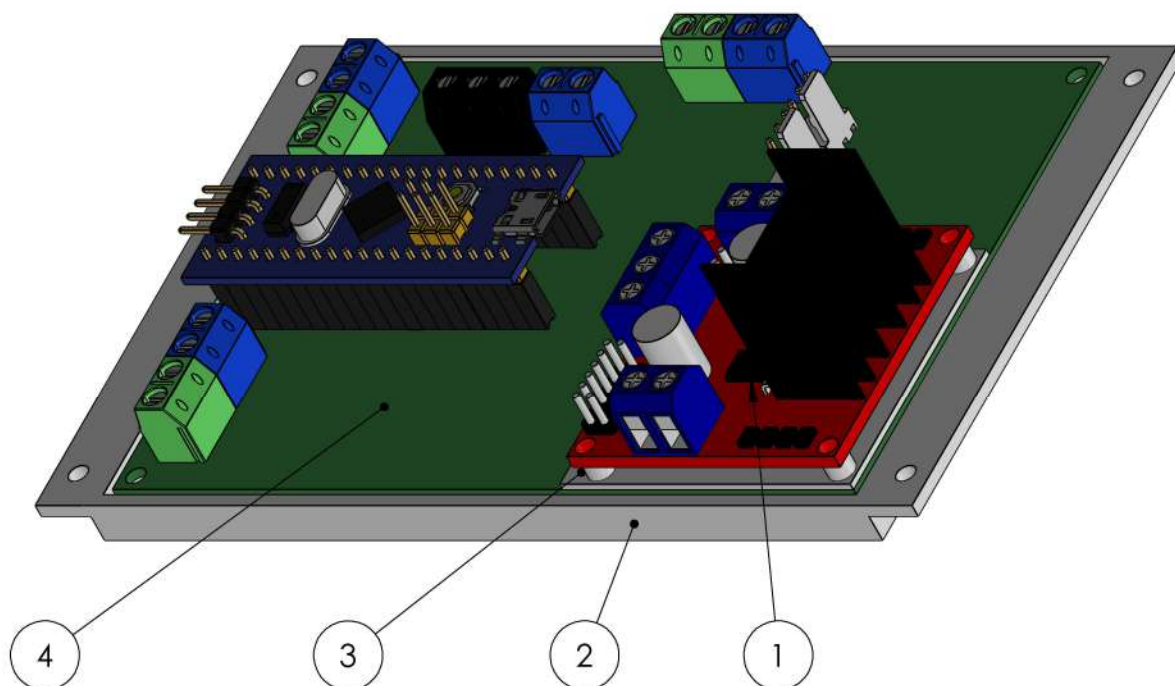


2	2	sk8 2	01-03-22	22	PLA	6.57	
1	1	Pared polea	01-03-21	21	PLA	216.45	
Pos.	Cant.	Denominacion	N° plano	N° de pieza	Material	Masa (gr)	Observaciones

	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias		
	Dibujó	26/08/2024	J.M.ROMERO				
	Revisó	27/08/2024	M.F.BERGOGLIO				
	Aprobó						
Esc. 1:1		SOPORTE POLEA			Diseño, implementación y control de un péndulo invertido lineal		
					Trabajo Final de Grado		
Formato A4					01-03-20		19/25

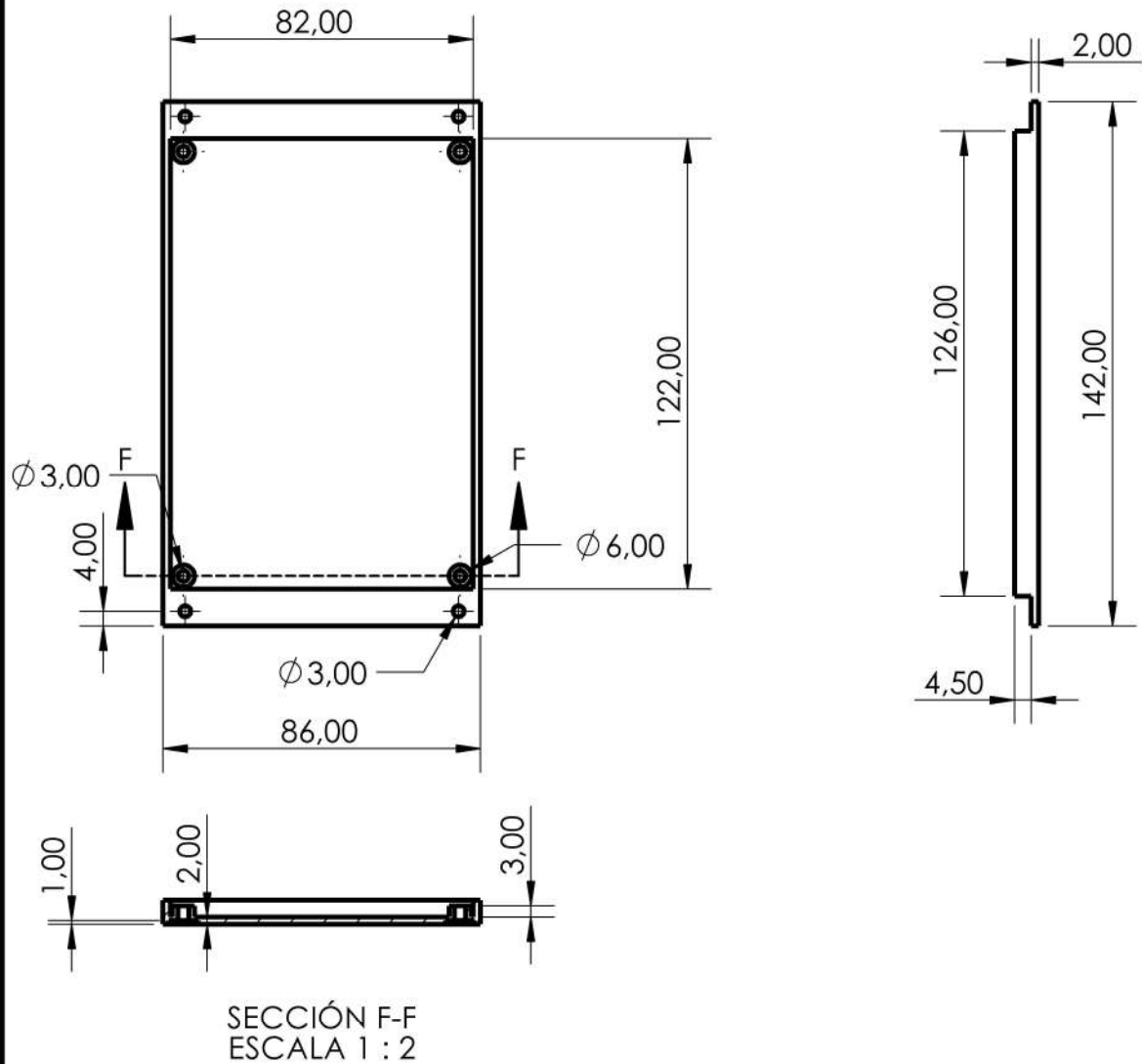
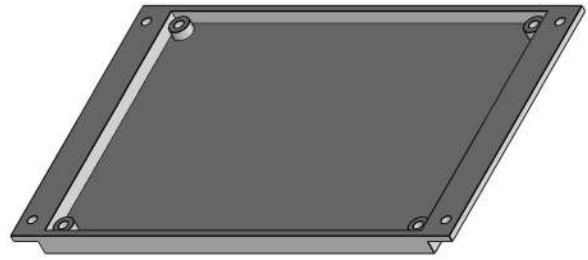


	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:1	SK8 2			Diseño, implementación y control de un péndulo invertido lineal	
	Trabajo Final de Grado					
Formato A4	01-03-22					
	21/25					

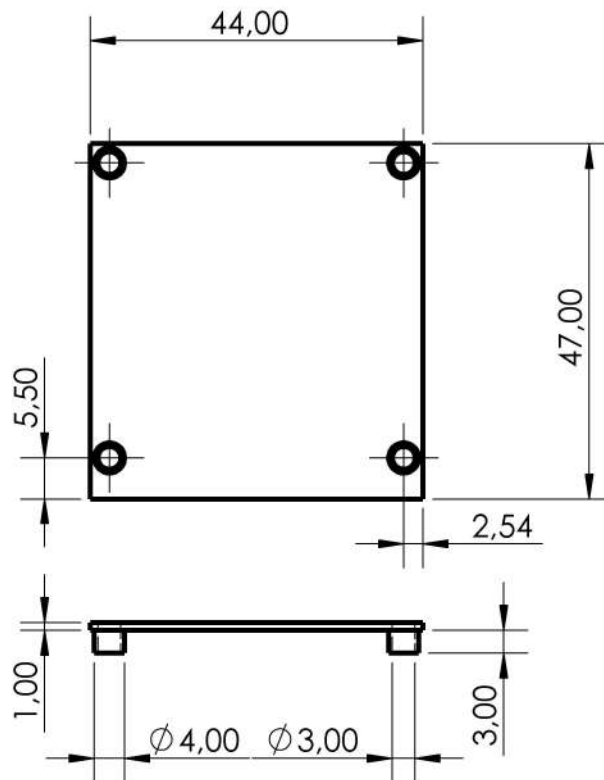
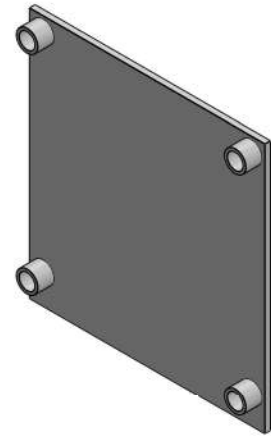


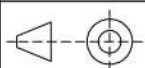
4	1	Ensamblaje placa electronica	01-04-03				Ensamblaje
3	1	Base puente H	01-04-02	25	PLA	2.74	
2	1	Base caja	01-03-01	24	PLA	36.58	
1	1	Puente H		23			L298N
Pos.	Cant.	Denominacion	N° plano	N° de pieza	Material	Masa (gr)	Observaciones

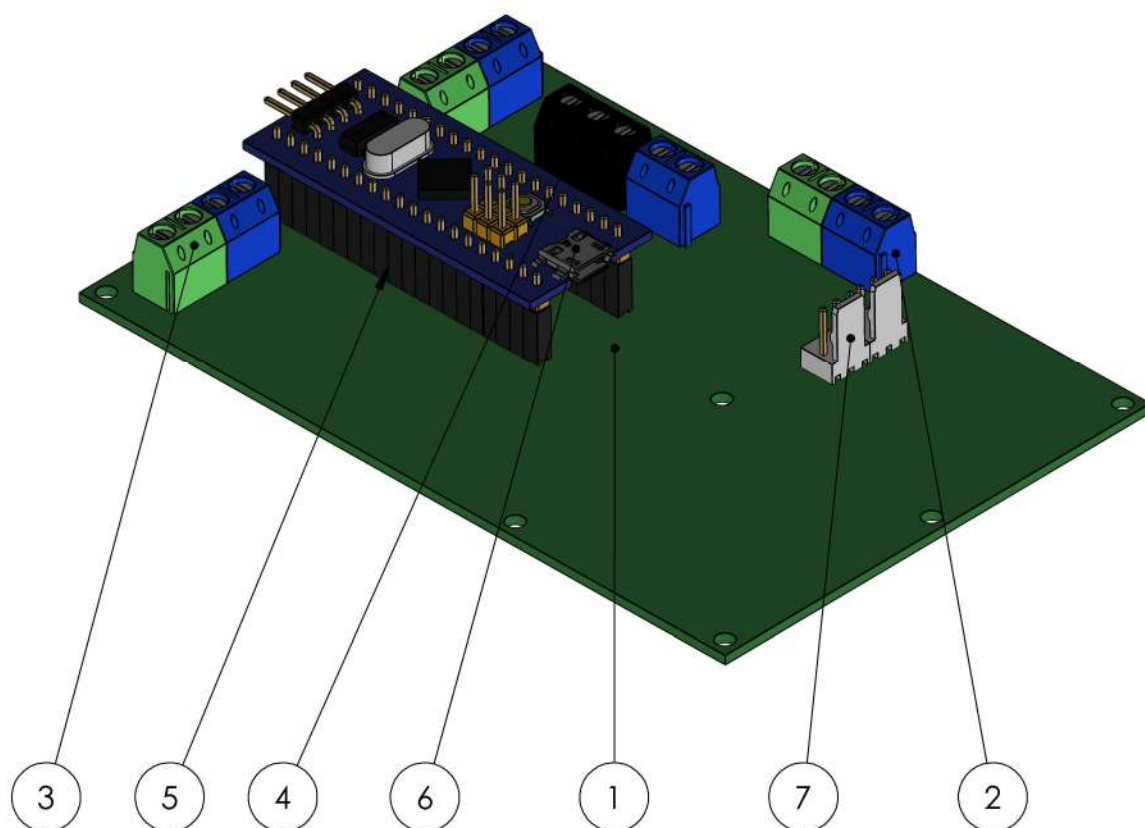
	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:1	PLACA ELECTRÓNICA			Diseño, implementación y control de un péndulo invertido lineal	
						
Formato A4						
					01-04-00	22/25



	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:2	BASE CAJA			Diseño, implementación y control de un péndulo invertido lineal	
						
	Formato A4				Trabajo Final de Grado	
					01-04-01	
		23/25				



	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
<div>Esc. 1:1</div> <div></div> <div>Formato A4</div>	BASE PUENTE H			Diseño, implementación y control de un péndulo invertido lineal		
				Trabajo Final de Grado		
				01-04-02	24/25	



7	2	fan header 2.54 3pin		33			Genérico
6	1	STM32F103C8T6		32			Genérico
5	2	Single row		31			Genérico
4	1	Bornera doble negra		30			Genérico
3	3	Bornera doble verde		29			Genérico
2	4	Bornera doble azul		28			Genérico
1	1	Placa electronica		27			Perforada 80 x 120 [mm]
Pos.	Cant.	Denominacion	Nº plano	Nº de pieza	Material	Masa (gr)	Observaciones

	Proyectó	26/08/2024	J.M.ROMERO	UNIVERSIDAD NACIONAL DE SAN LUIS	Facultad de Ingeniería y Ciencias Agropecuarias	
	Dibujó	26/08/2024	J.M.ROMERO			
	Revisó	27/08/2024	M.F.BERGOGLIO			
	Aprobó					
	Esc. 1:1	ENSAMBLAJE PLACA ELECTRÓNICA			Diseño, implementación y control de un péndulo invertido lineal	
					Trabajo Final de Grado	
	Formato A4				01-04-03	
					25/25	