

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Modelado, simulación y control de dos propuestas de extremidades altamente dinámicas bio-inspiradas en lepóridos.

Trabajo de graduación presentado por Christian Alejandro Sandoval
Porras para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,

2021

Modelado, simulación y control de dos propuestas de extremidades altamente dinámicas bio-inspiradas en lepidóridos.

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Modelado, simulación y control de dos propuestas de extremidades altamente dinámicas bio-inspiradas en lepóridos.

Trabajo de graduación presentado por Christian Alejandro Sandoval
Porras para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,

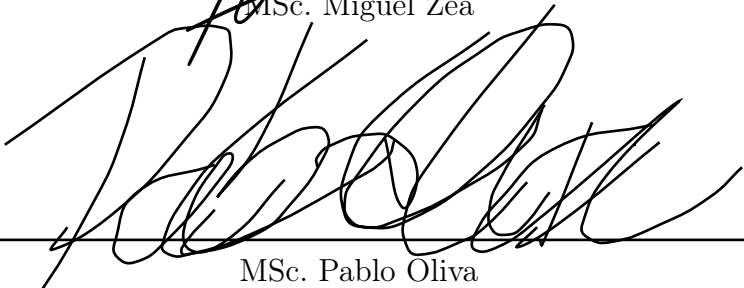
2021


Vo.Bo.:

(f) 
MSc. Miguel Zea

Tribunal Examinador:

(f) 
MSc. Miguel Zea

(f) 
MSc. Pablo Oliva

(f) 
PhD. Luis Rivera

Fecha de aprobación: Guatemala, 19 de Enero de 2021.

Expreso un sincero agradecimiento a MSc. Miguel Zea por el asesoramiento y arduo trabajo para la realización de este trabajo de graduación, así como el conocimiento brindado. Se le agradece al Ing. Pablo Mazariegos por la ayuda brindada en la manufactura y asesoramiento de maquinado aditivo durante la pandemia.

Se le agradece al departamento de ingeniería mecatrónica por brindar los materiales para realizar el proyecto.

Finalmente agradezco a mi familia y seres queridos por el apoyo incondicional que me han brindado para mi desarrollo profesional.

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| Prefacio | v |
| Lista de figuras | xii |
| Lista de cuadros | xiii |
| Resumen | xv |
| Abstract | xvii |
| 1. Introducción | 1 |
| 2. Antecedentes | 3 |
| 2.1. FastRunner | 3 |
| 2.2. Spot y ANYmal | 4 |
| 2.3. Desarrollo de una extremidad bio-inspirada en las extremidades posteriores de lepóridos con manejo eficiente de energía | 4 |
| 2.4. Diseño e implementación de una pata robótica de un grado de libertad bio-inspirada en las extremidades posteriores de lepóridos | 5 |
| 3. Justificación | 7 |
| 4. Objetivos | 9 |
| 4.1. Objetivo general | 9 |
| 4.2. Objetivos específicos | 9 |
| 5. Alcance | 11 |
| 6. Marco teórico | 13 |
| 6.1. Lepóridos | 13 |
| 6.2. Robótica bio-inspirada | 13 |
| 6.2.1. Morfologías bio-inspiradas | 14 |
| 6.2.2. Sensores bio-inspirados | 15 |
| 6.3. Sistemas de control | 16 |

| | | |
|-----------|---------------------------------------------------------------|-----------|
| 6.3.1. | Sistemas de control clásico: PID | 16 |
| 6.3.2. | Sistemas de control clásico: Control digital | 17 |
| 6.3.3. | La transformada Z | 17 |
| 6.4. | Control moderno | 18 |
| 6.4.1. | Control LQR | 18 |
| 6.4.2. | Control LQI | 18 |
| 6.5. | Optimización de trayectorias: programas cuadráticos | 19 |
| 6.5.1. | Método de single-shooting | 19 |
| 6.5.2. | Método de multiple-shooting | 20 |
| 6.5.3. | Método de colocación directa | 20 |
| 6.6. | Análisis utilizando visión de computadora | 20 |
| 6.6.1. | El marco coordinado de la cámara | 21 |
| 6.6.2. | Calibración de la cámara | 21 |
| 6.6.3. | Determinando los parámetros de la cámara | 22 |
| 6.6.4. | Posición y orientación | 22 |
| 6.7. | Análisis y simulación en software RecurDyn | 22 |
| 6.8. | Modelo SLIP (Spring-Loaded Inverted Pendulum) | 24 |
| 6.8.1. | Análisis de Poincaré | 25 |
| 6.8.2. | Mapa del modelo SLIP | 26 |
| 6.8.3. | Control del modelo | 26 |
| 6.9. | Mecánica de Lagrange | 26 |
| 7. | Análisis y simulación de las extremidades | 29 |
| 7.1. | Recurdyn | 29 |
| 7.1.1. | Simulink | 32 |
| 7.2. | Resultados | 33 |
| 7.2.1. | Desplazamiento angular | 33 |
| 7.2.2. | Desplazamiento horizontal | 34 |
| 7.2.3. | Desplazamiento vertical | 34 |
| 7.2.4. | Esfuerzos de control(Torque) | 34 |
| 7.2.5. | Discusión | 35 |
| 8. | Plataforma de programación. | 37 |
| 8.1. | UART | 38 |
| 8.2. | CMSIS DSP | 38 |
| 8.3. | Timer 0 | 39 |
| 8.4. | Timer 1 | 39 |
| 8.5. | Interfaz de encoder de cuadratura | 39 |
| 8.6. | PWM | 40 |
| 8.7. | Validación de la programación | 41 |
| 8.8. | Esquema de conexiones | 42 |
| 8.9. | Control de versiones y estándar de programación | 42 |
| 8.10. | Unidad de medición inercial | 42 |
| 9. | Rediseño de las extremidades | 45 |
| 9.1. | Trayectoria de correcciones a la fase 1 | 45 |
| 9.1.1. | Actualización en software. | 48 |
| 9.1.2. | Diseños futuros. | 48 |

| | |
|----------------------------------------------------------------|-----------|
| 9.2. Trayectoria de rediseño en la segunda fase | 49 |
| 9.2.1. Resultados del resorte | 50 |
| 9.3. Parametrización para impresión en 3D | 52 |
| 9.4. Modificaciones a la fase 2 | 53 |
| 10. Pruebas y simulaciones | 55 |
| 10.1. Fase 1 | 55 |
| 10.1.1. Modelo matemático fase 1 | 55 |
| 10.1.2. Parámetros y matriz de Denavit-Hartenberg | 58 |
| 10.1.3. Simulación con single-shooting | 59 |
| 10.1.4. Control dinámico para sistemas de base fija | 59 |
| 10.1.5. Pruebas realizadas | 60 |
| 10.2. Fase 2 | 61 |
| 10.2.1. Modelo slip | 61 |
| 10.2.2. Comparación entre modelo y pruebas realizadas. | 61 |
| 10.3. Limitaciones encontradas | 63 |
| 11. Conclusiones | 65 |
| 12. Recomendaciones | 67 |
| 13. Bibliografía | 69 |
| 14. Anexos | 71 |
| 14.1. Programación del microcontrolador | 71 |
| 14.2. Listado de piezas fase 1 | 72 |
| 14.3. Listado de piezas fase 2 | 73 |
| 15. Glosario | 75 |

Lista de figuras

| | | |
|-----|-----------------------------------------------------------------------------------------|----|
| 1. | Comparación entre FastRunner y una avestruz. | 3 |
| 2. | Robot Spot de Boston Dynamics. | 4 |
| 3. | Sistemas de control utilizado en robot ANYmal. | 4 |
| 4. | Comportamiento de la extremidad real(superior) y extremidad simulada(inferior). | 5 |
| 5. | Prototipo final de la extremidad realizada durante la segunda fase. | 6 |
| 6. | Lepórido y la anatomía muscular de la extremidad posterior. | 7 |
| 7. | Descripción muscular de la extremidad de una liebre. | 14 |
| 8. | Descripción muscular de la extremidad de una liebre. | 14 |
| 9. | Robot CONRO desarllado por USC-ISI. | 15 |
| 10. | Controlador PID.[9] | 16 |
| 11. | Convertor digital a analógico.[9] | 17 |
| 12. | Distintos métodos de transcripción directa. | 20 |
| 13. | Marco coordenado de la cámara | 21 |
| 14. | Interfaz general del programa | 23 |
| 15. | Juntas, fuerzas y restricciones. | 24 |
| 16. | Modelo SLIP. | 24 |
| 17. | Mapa de cúspide a cúspide del modelo SLIP. | 26 |
| 18. | Creación de tierra dentro de RecurDyn. | 30 |
| 19. | Juntas utilizadas dentro de la extremidad. | 30 |
| 20. | Configuración de una junta revoluta. | 31 |
| 21. | Fuerza rotacional simulando el motor. | 31 |
| 22. | Salida del sistema. | 31 |
| 23. | Configuración de Simulink en RecurDyn. | 32 |
| 24. | PID creado en Simulink. | 32 |
| 25. | Desplazamiento angular resultante. | 33 |
| 26. | Desplazamiento horizontal. | 34 |
| 27. | Desplazamiento vertical. | 35 |
| 28. | Esfuerzo de control. | 36 |
| 29. | Diagrama de flujo indicando la lógica detrás de la programación. | 38 |
| 30. | Salto en extremidad de la segunda fase sin restricciones de contacto. | 41 |

| | | |
|-----|-----------------------------------------------------------------|----|
| 31. | Circuito implementado. | 42 |
| 32. | Modificaciones realizadas a la transmisión de potencia. | 46 |
| 33. | Modificación al sistema de montaje de la extremidad. | 46 |
| 34. | Fractura de dos dientes debido a estrés. | 47 |
| 35. | Deformación debido a temperatura. | 47 |
| 36. | Ensamble realizado en software. | 48 |
| 37. | Pieza sugerida para la transmisión de potencia. | 49 |
| 38. | Engranajes cónicos. | 49 |
| 39. | Pin para transmisión de potencia. | 50 |
| 40. | Montaje para obtener constante del resorte. | 51 |
| 41. | Comportamiento del resorte a diferentes pesos. | 52 |
| 42. | Regla de parametrización. | 52 |
| 43. | Cuadro de diálogo para modificar tolerancias. | 53 |
| 44. | Modificaciones a la fase CAD. | 53 |
| 45. | Implementación de actuador en la cadera. | 54 |
| 46. | Modelo gráfico de la derivación analítica. | 56 |
| 47. | Esfuerzos de control realizados por el modelo. | 60 |
| 48. | Salto en un único lugar. | 60 |
| 49. | Salto del modelo en un único lugar. | 61 |
| 50. | Salto del modelo con ángulo de $\pi/8$ | 62 |
| 51. | Salto de la extremidad en un mismo lugar. | 62 |
| 52. | Salto de la extremidad con un desfase angular. | 63 |

Lista de cuadros

| | | |
|----|---------------------------------------------------------------------------|----|
| 1. | Longitudes promedio encontradas en liebres.[3] | 15 |
| 2. | Constantes PID utilizadas para el salto en extremidad con dos actuadores. | 35 |
| 3. | Constantes PID utilizadas para la caída en extremidad con dos actuadores. | 36 |
| 4. | Resultados del experimento. | 51 |
| 5. | Masas y centros de masa de los eslabones. | 58 |
| 6. | Inercias de cada eslabón. | 58 |
| 7. | Matriz D-H. | 59 |
| 8. | Constantes PID para realizar un salto. | 63 |

En el siguiente trabajo se propuso caracterizar extremidades, bio-inspiradas en lepóridos, altamente dinámicas de fases previas realizadas por estudiantes y evaluar distintos esquemas de control para manipular y tener un mejor aprovechamiento de energía. Para evaluar y tener métricas cuantificables, así como validaciones, se caracterizaron ambas extremidades utilizando el software RecurDyn junto a la plataforma Simulink en Matlab. Este permitió validar los distintos esquemas de control propuestos revisando el comportamiento con la plataforma de pruebas real. Se unificaron las plataformas de prueba utilizando el microcontrolador TM4C123G para controlar las plantas de los sistemas e implementar nuevos métodos de control junto con el software uVision debido a las amplias librerías especialmente en el área de manejo de matrices. Por último, se implementó un software capaz de resolver programas cuadráticos utilizados en optimización de trayectorias. Los resultados de estos se implementaron empleando el microcontrolador en el cual se validaron los resultados utilizando programas externos. Durante el trabajo se comprobó por medio de modelos matemáticos que el control mínimo necesario para que la extremidad funcione adecuadamente es un controlador PID.

In the following work we proposed to characterize highly dynamic limbs of previous phases, bio-inspired in leporidae, carried out by previous students. In this project we evaluate different control schemes to better manipulate the extremities and have a better energy management. To evaluate and have quantifiable metrics as well as validations, both extremities were characterized using RecurDyn software coupled with the Simulink platform in Matlab. This allowed to validate the different proposed control schemes by reviewing the behavior with the real test platform. The test platforms were unified using the TM4C123G micro controller to control the plants of the systems and implement new control methods together with the help of uVision software along the Texas Instruments library that provides useful resources and functions. Finally, a software capable of solving quadratic programs used in trajectory optimization was implemented. The results of these were implemented using a microcontroller in which the results were validated using external programs. During the work it was verified by means of mathematical models that the minimum control necessary for the limb to function properly is a PID controller.

Durante esta fase de investigación se buscó mejorar fases existentes de robots inspirados en lepóridos. Para que, futuramente este pueda salir exitosamente de una plataforma de pruebas e implementarse en el mundo real para realizar tareas que sean beneficiosas para el ser humano.

En este proyecto no se tiene únicamente el objetivo de implementar un robot eficiente si no altamente asequible y que sea accesible a todas las personas para futura investigación. Utilizando equipo de maquinado aditivo, así como resortes y tornillos estandarizados, se es capaz de construir las extremidades siendo estas fáciles de modificar durante el diseño CAD.

Finalmente, utilizando técnicas de control y microcontroladores de bajo poder computacional se controlaron las extremidades para lograr un comportamiento deseado, probando su validez utilizando modelos matemáticos.

En el capítulo 7 se introduce al software de simulación dinámica RecurDyn. En este se explica el modo de funcionamiento y la caracterización de las extremidades. Junto con Simulink se muestran los distintos resultados dentro de un entorno de simulación en los cuales se obtuvieron parámetros importantes para comparar los diseños de ambas extremidades. Durante el capítulo 8 se mostró la lógica seguida para realizar la programación del nuevo microcontrolador, así como las pruebas tempranas realizadas a las extremidades. Para el capítulo 9 se muestra una trayectoria de rediseño para ambas extremidades. En este capítulo se discuten los temas que causaron problemas durante las pruebas físicas y cómo se solucionaron. Por último, en el capítulo 10 se muestran las caracterizaciones matemáticas de ambas extremidades y los resultados de las pruebas físicas junto con una comparación directa entre estas dos situaciones para ambas extremidades.

2.1. FastRunner

El FastRunner es un robot bípedo basado en las extremidades de los avestruces, animales que pueden llegar a velocidades de hasta $70km/h$. Este robot bípedo aprovecha energía almacenada en cada paso para correr de manera más eficiente. Esto se logra mediante resortes modelados no linealmente ya que se muestra que, a través de varios experimentos, esto incrementa la velocidad alcanzada por el robot. A diferencia de otros robots el FastRunner tiene un único actuador por pierna haciendo que este tenga un bajo consumo de energía.[1]

Debido a la tecnología mecánica dentro del FastRunner este no requiere de esquemas de control a menos que se realicen maniobras complicadas. Es por eso que existen eslabones incluidos en el mecanismo que se encargan de emular elementos pasivos así como tendones y ligamentos. Tomando en cuenta todas estas consideraciones mecánicas dicho robot es capaz de alcanzar velocidades de hasta $34.5km/h$. [1]



Figura 1: Comparación entre FastRunner y una avestruz.



Figura 2: Robot Spot de Boston Dynamics.

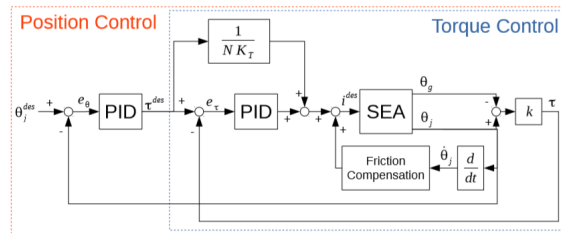


Figura 3: Sistemas de control utilizado en robot ANYmal.

2.2. Spot y ANYmal

Este tipo de robots son inspirados en cuadrúpedos capaces de correr a altas velocidades y escalar fácilmente. En el caso del robot ANYmal, elaborado por ETH Zürich, utiliza 12 actuadores entre sus cuatro extremidades para hacer el movimiento. Este utiliza sensores de contacto para realizar esquemas de control complicados. Estos sensores sensibles a la fuerza, combinados con sensores rotativos LIDAR “Hokuyo”, proveen una percepción tridimensional del ambiente para que el robot sea apto para distintos escenarios.[2]

Para el control de los actuadores, se utilizan estructuras en cascada que consideran el motor como una fuente de torque y dos controladores PID para posición y torque con retroalimentación junto con una compensación de fricción. Este esquema se puede observar en la Figura 3.

2.3. Desarrollo de una extremidad bio-inspirada en las extremidades posteriores de lepóridos con manejo eficiente de energía

Esto conforma la primera fase del proyecto. Con dicha fase del proyecto se inicia la construcción de la extremidad. Este prototipo se basa en una serie de proyectos, así como “FastRunner” y “Extremidades felinas robóticas biomiméticas”. Dichos estudios hablan acerca de mecanismos comparados con la estructura ósea de animales y la comparación entre los resultados obtenidos en los proyectos y el desempeño real que se da en dichos seres vivos.[3]

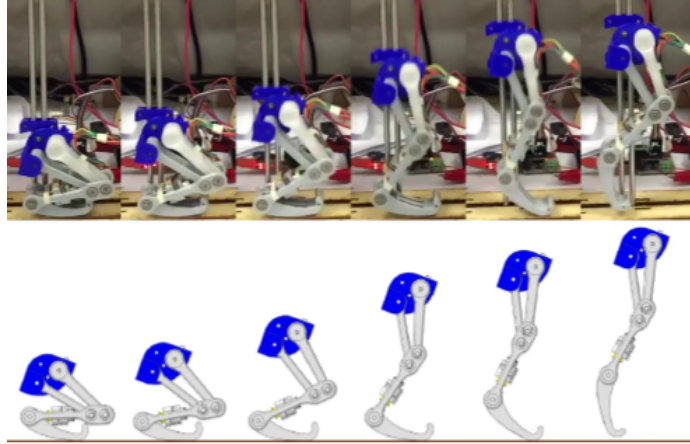


Figura 4: Comportamiento de la extremidad real(superior) y extremidad simulada(inferior).

En la realización del proyecto se hablan de distintos modos de diseño para simular la extremidad del lepórido. En esta se menciona un análisis por medio de resortes en el cual se reemplazarían los músculos de los lepóridos. Sin embargo, durante la realización de los prototipos este método se eliminó. En lugar de utilizar resortes, se utilizó este mismo concepto de reemplazo de tendones utilizando controladores PID en los motores. Como último prototipo se fabricaron diversas piezas con material de policarbonato y se utilizaron dos motores DC para el accionamiento de la extremidad. Luego se realizaron diversas plataformas de prueba en la cual se mencionó el uso de distintas juntas para dar como resultado una junta prismática para implementar la extremidad en la plataforma de prueba. Para la realización de la plataforma se buscó que el contacto y desplazamiento de la guía tuviera la menor fricción posible y se diera un movimiento natural sin limitaciones. Para la implementación de un control sobre los motores se utilizó la teoría de control clásico utilizando controladores PID para cada motor en la extremidad. Debido a la baja cantidad de dispositivos de retroalimentación se utilizó un tiempo estimado para cambiar valores de control. [3]

Como recomendación de este trabajo se tomó el incremento de sensores para sincronizar los movimientos de la extremidad, así como hacer un rediseño del mecanismo y de la placa de control.[3]

2.4. Diseño e implementación de una pata robótica de un grado de libertad bio-inspirada en las extremidades posteriores de lepóridos

Este trabajo de investigación conforma la segunda fase del proyecto. Durante el trabajo de graduación, se discute el mejoramiento de la plataforma de pruebas así como la reducción de un grado de libertad (al remover un motor) para un manejo más eficiente de la extremidad.[4]

Durante esta fase se implementó el uso de hules como simulación de un resorte para dar un mejor aprovechamiento de la energía contenida dentro del sistema. Dicha fase logró

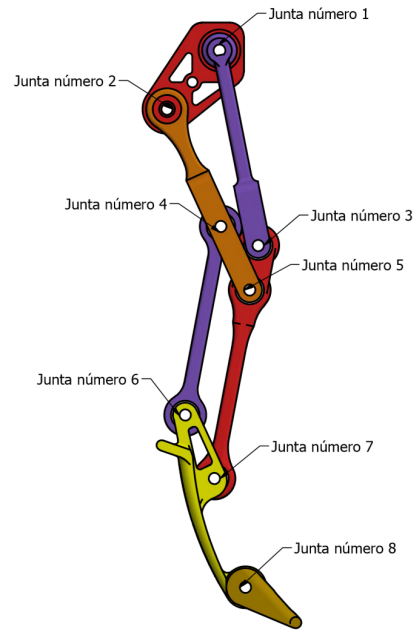


Figura 5: Prototipo final de la extremidad realizada durante la segunda fase.

disminuir el peso de la extremidad y lograr diversos avances en el aspecto mecánico. Las recomendaciones tomadas de este proyecto fueron discretizar los parámetros medibles que varían el comportamiento de la extremidad para que el proyecto tenga una mejor reproducibilidad. [4]

Los lepóridos conocidos como liebres o conejos poseen dos extremidades posteriores que les permite alcanzar altas velocidades (de hasta 70km/h) y alcanzar alturas que para un animal de ese tamaño parecieran imposibles. Estas velocidades y alturas se alcanzan con músculos retractores del glúteo y músculos en la entrepierna que entran en funcionalidad al final de cada golpe contra el piso. Con sus extremidades anteriores los lepóridos son capaces de realizar movimientos y cambios bruscos de dirección. Como se observa en la Figura 6, las extremidades posteriores son altamente complejas y están compuestas por una cantidad elevada de músculos y tendones.[5]

El desarrollar una extremidad bio-inspirada en lepóridos nos permitiría explotar estas velocidades y alturas que se observan en dicha familia. Con este trabajo se replicaron dichos mecanismos con actuadores mecánicos. Además, se investigó el aprovechamiento y la conservación de energía para realizar una extremidad altamente eficiente capaz de llegar a altas velocidades.



Figura 6: Lepórido y la anatomía muscular de la extremidad posterior.

Actualmente existen varios proyectos bio-inspirados en extremidades complejas, este es el caso del FastRunner y los robots Spot y ANYmal. Estos presentan una gran cantidad de actuadores mecánicos exceptuando al FastRunner que implementó un actuador por cada extremidad, para que el consumo de energía de este fuera reducido. Todos los proyectos presentan simulaciones de tendones ya sea reemplazándolos con algún tipo de resortes u otro material que tenga cierta regeneración de energía.

Durante este trabajo de graduación se continuó con las fases previas que consisten en dos extremidades distintas y se evaluó el desempeño de estas implementando software de simulación dinámico y distintos esquemas de control para alcanzar un máximo desempeño en cada una de estas. Se implementaron una mayor cantidad de sensores para tener un movimiento sincronizado y modelado del sistema correcto. El aumentar la cantidad de sensores e implementar análisis utilizando visión de computadora permitió encontrar parámetros de medición para medir dicho desempeño de una forma cuantitativa.

4.1. Objetivo general

Caracterizar y controlar extremidades resultantes de fases previas en la línea de investigación para evaluar distintos esquemas de control.

4.2. Objetivos específicos

- Mejorar la respuesta dinámica de los sistemas durante el despegue, aterrizaje y la secuencia de saltos mediante la caracterización utilizando software de simulación dinámica de cuerpos rígidos.
- Implementar distintos esquemas de control para obtener un mejor desempeño del sistema utilizando programas cuadráticos.
- Cambiar el microcontrolador utilizado para unificar y establecer una plataforma específica.

Durante el desarrollo del proyecto se planeó caracterizar y controlar extremidades altamente dinámicas existentes de fases previas. En el siguiente trabajo se implementó lo siguiente:

- Únicamente se rediseñaron las fases existentes para facilitar el proceso de manufactura y optimizar la respuesta de las extremidades para que estas no presenten problemas a la hora de realizar pruebas.
- Se comprobó el porqué es suficiente implementar un controlador PID mediante optimización de trayectorias.
- Se modelaron matemáticamente las extremidades para validar cuantitativamente las extremidades físicas dentro de una plataforma de pruebas.
- Se realizaron las pruebas dentro de una plataforma especializada para restringir el movimiento eliminando las rotaciones y uno de los ejes coordenados.
- Se trabajó con el mismo hardware existente para optimizar procesos.

Debido a la pandemia del COVID-19 durante el año 2020, se presentó cierta dificultad en varios aspectos del proyecto. Manufacturar las piezas por medio de un proceso de impresión 3D de forma libre no fue posible ya que no se asistió presencialmente a la universidad. El equipo de trabajo (fuente de voltaje) se adquirió para realizar las pruebas necesarias. Por último, se presentó cierta dificultad para conseguir los materiales electrónicos.

6.1. Lepóridos

Los lepóridos son una familia de mamíferos conformados por liebres y conejos, estos provienen del orden taxonómico de los lagomorfos en donde también se encuentran los pikas. Este grupo taxonómico tiene como característica las extremidades posteriores. [6] Debido a dichas extremidades las liebres son capaces de correr rápidamente a una velocidad máxima de $20m/s$. [7]

Las extremidades encargadas de este movimiento son las pélvicas y las torácicas. La pelvis se cree que es el propulsor primario de las liebres mientras el tórax es el encargado de soportar el peso y la desaceleración de la extremidad. [7] En la Figura 7 se aprecia la fisiología de la extremidad de una liebre. Esta extremidad es la que se desea reproducir durante los experimentos.

En la Figura 8 se muestra el uso de las extremidades posteriores a bajas velocidades (a) y el uso de las extremidades a altas velocidades (b). Para ambos movimientos las liebres utilizan una caída con la planta del pie, en donde ambas son idénticas. Por lo tanto, esta especie es capaz de realizar ambos movimientos de galope y correr a altas velocidades de manera similar.

Como se observa en el Cuadro 1 estas son las extremidades promedio de una liebre. Esto es importante porque las extremidades deberían tener dimensiones similares para replicar lo más posible la extremidad de una liebre.

6.2. Robótica bio-inspirada

La ingeniería ha encontrado inspiración en los productos de la naturaleza. En los últimos años, las personas especializadas en robótica se han basado mucho más en matemáticas,

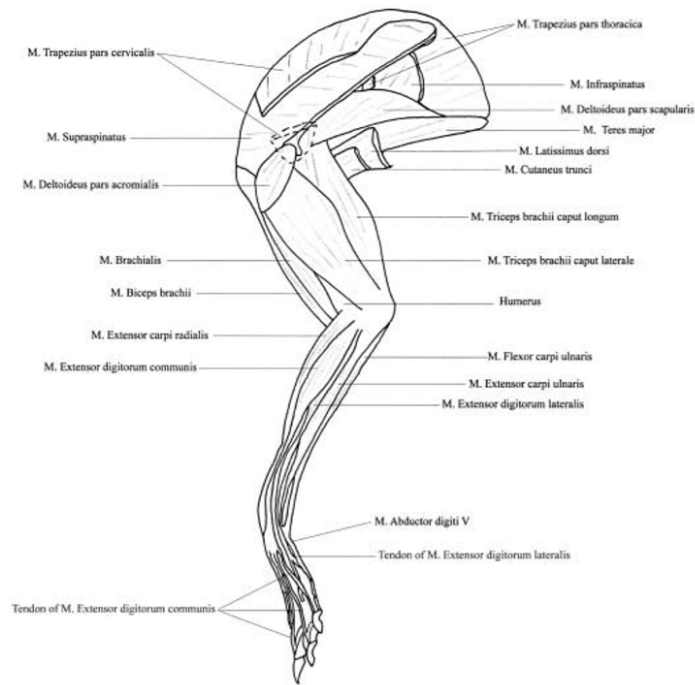


Figura 7: Descripción muscular de la extremidad de una liebre.

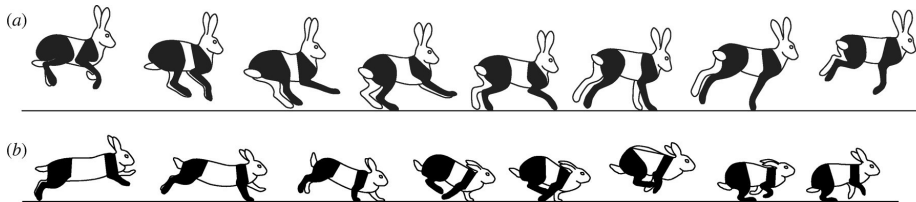


Figura 8: Descripción muscular de la extremidad de una liebre.

mecánica y electrónica que de la biología en sí. Esto ha creado una separación entre los robots y los animales.

Hoy en día los avances en biología producen una gran cantidad de datos y modelos que pueden ser explotados por los ingenieros, no se hable de los avances tecnológicos con los sistemas de cómputo siendo cada vez más baratos y la exploración de nuevos materiales que presenten propiedades totalmente distintas.[8]

6.2.1. Morfologías bio-inspiradas

La morfología de un robot se puede dividir en distintas áreas como por ejemplo, se puede decir que un robot muy apegado a la realidad puede ser aquel hecho para entretenimiento o más conocido como animatronics. Sin embargo, esto también puede significar aquel robot que tenga habilidades para facilitar las interacciones con los humanos.[8]

Un área interesante explorada hoy en día son los robots que pueden cambiar de forma o

| Hueso | Longitud (mm) |
|-------|-----------------|
| Coxis | 101.1 ± 2.5 |
| Fémur | 132.4 ± 2.2 |
| Tibia | 148.2 ± 4.4 |
| Pie | 188.4 ± 4.8 |

Cuadro 1: Longitudes promedio encontradas en liebres.[3]

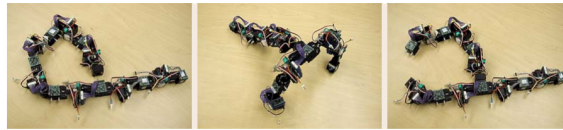


Figura 9: Robot CONRO desarrollado por USC-ISI.

más bien reconfigurar su morfología dependiendo de circunstancias internas o externas como lo es el caso del robot “Conro” hecho por el laboratorio polimórfico de USC-ISI (Figura 9).[8]

6.2.2. Sensores bio-inspirados

Los sensores bio-inspirados pueden variar en un gran rango y existen diversos tipos.

- **Visión:** Esto puede ir desde un dispositivo foto sensitivo hasta dispositivos binoculares usados para el reconocimiento de objetos. En este rango de sensores no solo se encuentran aquellos que tratan de replicar el ojo humano sino existen sensores especializados en reconocer formas y colores para establecer una detección y seguimiento de objetivos.[8]
- **Sensores auditivos:** Estos sensores se utilizan para un amplio rango de aplicaciones. Un ejemplo de esto es el sensor desarrollado por la universidad de Edinburgo, en el cual es capaz de reconocer las vibraciones emitidas por los grillos y reconocer objetos como murciélagos para evitar colisiones.[8]
- **Sensores del Tacto:** Usualmente se cree que de los cinco sentidos, el tacto es el más difícil de replicar de forma mecánica. Un sensor altamente sensible fue desarrollado por la universidad de Stanford con el robot “Sprawlette” desarrollando inspiración de cómo las cucarachas utilizan las antenas para un control por retroalimentación.[8]

Otra alternativa de sensores de tacto puede encontrarse en los robots humanoides en el cual utilizan transistores orgánicos como sensores de presión. Con esto se puede realizar una piel que sea sensible a la presión y a la temperatura. La universidad de Nebraska ha sido capaz de desarrollar un sensor táctil que es capaz de crear un mapa topográfico del objeto sensado.[8]

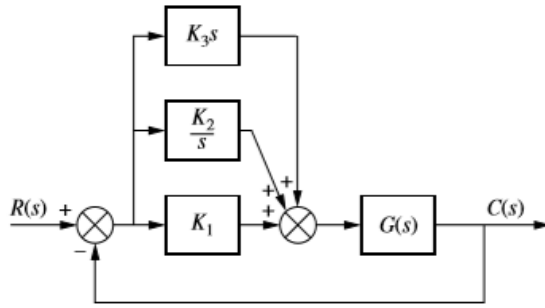


Figura 10: Controlador PID.[9]

6.3. Sistemas de control

Un sistema de control es aquel encargado de manipular un sistema para que se comporte de una manera deseada. Los sistemas de control son una parte integral de la sociedad moderna, estos se encargan de diversas aplicaciones. Los sistemas de control no solo se encuentran en la industria, sino que también existen en la naturaleza, un ejemplo de esto es el páncreas que regula el azúcar en la sangre.[9]

Un sistema de control consiste en una serie de subsistemas y procesos o comúnmente llamados plantas para obtener una salida deseada con el desempeño especificado. Estos sistemas de control se construyen para una amplificación de potencia, control remoto, conveniencia en la forma de entrada y compensación de perturbaciones.[9]

6.3.1. Sistemas de control clásico: PID

Este tipo de controlador es el más utilizado en la industria. Existen diversos tipos de controladores PID, como por ejemplo: hidráulicos, neumáticos y electrónicos. Durante este trabajo se utilizaron los controladores PID electrónicos.[10]

Estos controladores PID son muy útiles cuando el modelo matemático de la planta no se conoce y por lo tanto no se pueden emplear métodos de diseño analíticos. El diagrama de un controlador PID se observa en la Figura 10.

A partir de este diagrama de bloques se obtiene en el dominio de tiempo:

$$u(t) = k_{pe}(t) + k_I \int_0^t e(\tau) d\tau + k_d \dot{e}(t). \quad (1)$$

Con esto se observa que el primer término es aquel proporcional, que corresponde al error presente. El segundo término es la parte integral que corresponde al error pasado o acumulado. El tercer término es la parte derivativa que corresponde al error futuro en el que se trata de predecir el error.

Usualmente las constantes del controlador PID se seleccionan a prueba y error, aunque de ser necesario existen métodos para obtener parámetros iniciales de los controladores y a

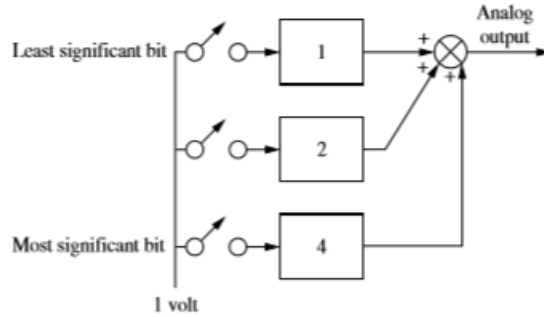


Figura 11: Conversor digital a analógico.[9]

partir de estos valores se van modificando hasta lograr la respuesta deseada del sistema.

6.3.2. Sistemas de control clásico: Control digital

Una gran parte de sistemas hoy en día utiliza microprocesadores o microcontroladores para los sistemas de control, por este motivo fue necesario hacer un replanteo de los sistemas de control. Para esto es necesario conversores ADC y DAC que conviertan las señales según el uso del sistema.

Conversor digital a analógico(DAC): Este tipo de conversión es simple e instantáneo. En este se suman todos los voltajes de cada una de las salidas del microcontrolador, se establece un orden del bit menos significativo hasta el bit más significativo y se multiplican por una constante de cada bit. Luego se suma cada señal y se obtiene la reconstrucción de una señal analógica como se puede observar en la Figura 11.[9]

Conversor analógico a digital(ADC): Este proceso no es tan sencillo como el DAC ya que requiere de un proceso de dos pasos y no es instantáneo. Para este proceso existe un retraso entre la señal de entrada analógica y la salida digital.[9]

Para este se debe establecer una frecuencia de muestreo que sea por lo menos dos veces mayor a la frecuencia de la señal. A esto se le conoce como el criterio de Nyquist:

$$f_s \geq 2f_m. \quad (2)$$

6.3.3. La transformada Z

A partir de esto se obtuvo una señal muestreada $u^*(t)$ que es el resultado de la multiplicación de la señal continua en el tiempo $u(t)$ con un tren de impulsos $p(t)$.

$$u^*(t) = u(t)p(t). \quad (3)$$

De esto se puede establecer una señal discreta definida en un tiempo continuo:

$$u^*(t) = \sum_{k=-\infty}^{\infty} u(kT)\delta(t - kT). \quad (4)$$

Lo cual se traduce al dominio de la frecuencia como:

$$U^*(s) = \sum_{k=-\infty}^{\infty} (kT)e^{-kTs}. \quad (5)$$

El término e^{sT} se puede denominar por medio de una transformación $z(t)$, en donde se obtuvo la transformada Z de la señal discreta en el tiempo $u(kT)$

$$U^*(s) = \sum_{k=-\infty}^{\infty} (kT)z^{-k}. \quad (6)$$

De esto se puede concluir que la transformada z es el equivalente a la transformada de Laplace.[9]

6.4. Control moderno

Debido a que la caracterización de una extremidad se considera como una aplicación compleja, quizá no sea suficiente un control PID. Esto se debió a que se tienen cambios en los objetivos de control durante el transcurso de tiempo. Por lo tanto, se necesita un cambio de control luego que la extremidad se haya levantado y vaya en el proceso de caída. También con los esquemas de control moderno se puede implementar una optimización de trayectorias para que la extremidad haga su mínimo esfuerzo.

6.4.1. Control LQR

La implementación de LQR se puede realizar solo si el sistema es completamente controlable. Este método de control es de tipo lineal, cuyas constantes se determinan resolviendo un problema de optimización con respuesta:[11]

$$u = -k(x - x_{ss}) + u_{ss}. \quad (7)$$

6.4.2. Control LQI

Este tipo de control toma cierta intuición a partir del control clásico tomando en cuenta variables proporcionales, derivativas e integrales. Este se diferencia al control LQR ya que se añade una retroalimentación añadiendo una parte integral al sistema. Este provoca que el sistema tenga un error en estado estable de cero.[12]

Este método tiene como objetivo crear un estado en el controlador que calcula la integral de la señal de error, lo cual se utiliza luego como retroalimentación. Esto se hace creando un estado z de la forma:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix}. \quad (8)$$

6.5. Optimización de trayectorias: programas cuadráticos

Hay muchas maneras de formular un problema de optimización de trayectorias. Para este proyecto se asumió que el sistema dinámico presentado por la extremidad es continuo a través de toda la trayectoria. En general, la función objetivo puede incluir dos términos. Un objetivo de frontera y un camino integral a lo largo de toda la trayectoria. Un problema con los dos términos se dice que está en la forma “Bolza”. [13]

Un problema solo con el término integral se dice que tiene la forma Lagrangiana y un problema con únicamente el término de frontera se dice que se encuentra en la forma de Mayer. Esto se observa en la siguiente ecuación:

$$\min_{t_0, t_F, \mathbf{x}(t), \mathbf{u}(t)} \underbrace{J(t_0, t_f, \mathbf{x}(t_0), \mathbf{x}(t_F))}_{\text{Término de Mayer}} + \underbrace{\int_{t_0}^{t_F} w(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau}_{\text{Término de Lagrange}}. \quad (9)$$

Para optimización se utiliza el término “variables de decisión” para describir las variables que se ajustarán en el solver para minimizar o maximizar la función objetivo. [13]

Los métodos para resolver optimización de trayectorias se clasifican en dos, directos e indirectos. En el método directo se discretiza la trayectoria de optimización, convirtiendo el problema a un programa no lineal. Esto se da por el proceso de transcripción. [13]

La mayoría de los métodos de colocación directa transcriben un problema de optimización de trayectorias en tiempo directo a un programa no lineal, este se da de la siguiente manera: [13]

$$\begin{aligned} \min_z \quad & J(z) \\ \text{s.t} \quad & \mathbf{f}(\mathbf{z}) = \mathbf{0}, \\ & \mathbf{g}(\mathbf{z}) \leq \mathbf{0}, \\ & z_{inf} \leq z \leq z_{sup} \end{aligned} \quad (10)$$

Es importante notar que un programa cuadrático está contenido dentro de los programas no lineales. Por lo tanto, se estará buscando un programa con función de costo de la forma: [14]

$$J(k) = \mathbf{u}^T(k)H\mathbf{u}(k) + 2x^T(k)F^T\mathbf{u}(k) + x^T(k)Gx(k). \quad (11)$$

Para la discretización de las trayectorias existen dos métodos: Los métodos de disparo y los métodos de colocación. Los primeros son basados en simulación y son mejores para problemas que requieren un control simple y que no tenga restricciones de trayectoria. Los métodos de colocación son basados en aproximación de funciones y son mejores para problemas con ya sea un control complicado o restricciones de trayectorias. [15]

6.5.1. Método de single-shooting

El método de single-shooting resuelve un problema de optimización de trayectorias transformándolo en un programa no lineal. Las variables de decisión y el estado inicial en este método son una parametrización en lazo abierto del control a lo largo de la trayectoria. [15]

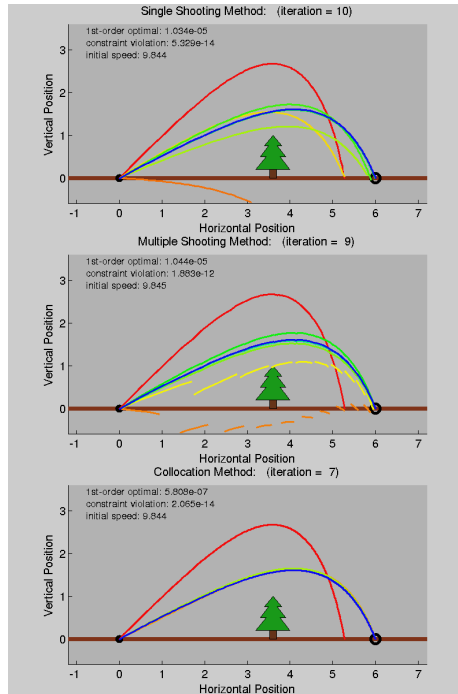


Figura 12: Distintos métodos de transcripción directa. Obtenido de: [16]

6.5.2. Método de multiple-shooting

A comparación del método único de disparo en donde se representa la trayectoria completa en una única simulación, la trayectoria se divide en segmentos. En donde cada trayectoria es representada por una simulación. Al utilizar este método de discretización se obtiene una optimización más robusta.[15]

Una de las desventajas de la discretización por disparo es que se presenta una alta dificultad para implementar restricciones de trayectoria ya que las variables de estado intermedias no son variables de decisión en el programa no lineal.[15]

6.5.3. Método de colocación directa

Los métodos de colocación directa son capaces de discretizar un problema de optimización de trayectoria continua mediante la aproximación de todas las funciones como polinomios. Dichos polinomios son utilizados ya que estos pueden ser representados por una cantidad finita de coeficientes y que es fácil de calcular las integrales y derivadas mediante métodos computacionales.[15]

6.6. Análisis utilizando visión de computadora

Si un robot debe interactuar con el ambiente entonces el robot necesariamente tiene que ser capaz de sensor este.

6.6.1. El marco coordenado de la cámara

Se define el plano de imagen como el plano que contiene una matriz de sensado. Los ejes z_x y z_y forman una base para el plano de la imagen mientras que el eje z_c es perpendicular a la imagen y alineado con el eje óptico del lente de la cámara. El origen de la cámara es ubicado a una distancia λ del plano imagen, este punto es conocido como el centro de proyección. El punto en el que el eje óptico interseca a la imagen se refiere como punto principal. Esta descripción se puede observar en la Figura 13. [17]

Proyección en perspectiva Las ecuaciones para una proyección en perspectiva son las siguientes:

$$u = \lambda \frac{x}{z}, \quad v = \lambda \frac{y}{z}. \quad (12)$$

El plano imagen y el sensor. Se supone que las filas y columnas de la matriz de pixeles se denotan (r, c) , estas se definen en una esquina de la imagen en vez del centro de ellas (su punto de origen está dado por (o_r, o_c)). Para relacionarlas con el mundo en 3D se necesita hacer la relación entre esta matriz y la proyección en perspectiva. Se supone como s_x y s_y las dimensiones horizontal y vertical de un pixel respectivamente.[17] Se puede hacer la siguiente relación:

$$u = -s_x(r - o_r), \quad v = -s_y(c - o_c). \quad (13)$$

6.6.2. Calibración de la cámara

El objetivo de realizar una calibración de cámara es para determinar todos los parámetros que son necesarios para predecir las coordenadas de los pixeles con un punto de vista de la cámara. Si conocemos la posición y orientación del marco de la cámara relativo al mundo se obtiene:

$$x^w = R_c^w x^c + O_w^c.$$

Con esto se puede establecer que las cámaras usualmente están montadas en un trípode y se puede realizar una técnica de grabación conocida como paneo. En este caso la matriz de rotación está dada por:

$$R = R_{z,\theta} R_{x,\alpha}. \quad (14)$$

En donde θ es el ángulo de paneo y α es el ángulo de inclinación.

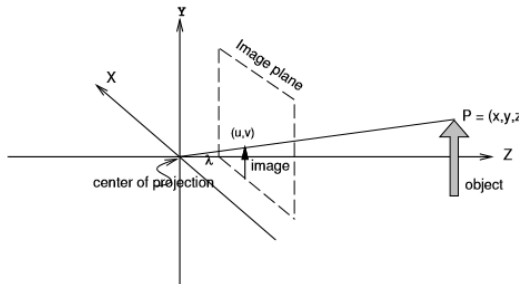


Figura 13: Marco coordenado de la cámara

Parámetros de cámara intrínsecos Estos están dados por:

$$f_x = -\frac{\lambda}{s_x}, \quad f_y = -\frac{\lambda}{s_y}. \quad (15)$$

Estos parámetros no cambian cuando la cámara se mueve.

6.6.3. Determinando los parámetros de la cámara

Para realizar una calibración se debe colocar un cubo en el espacio de trabajo y encontrar las orillas del cubo en la imagen. Esto producirá tres sets de líneas paralelas ortogonales. Luego de haber realizado este procedimiento se debe construir un sistema lineal de ecuaciones en términos de puntos coordinados conocidos en el mundo y las coordenadas de los pixeles en las proyecciones. Con esto se denota el siguiente sistema:

$$x^c = r_{11}x + r_{12}y + r_{13}z + T_x, \quad (16)$$

$$y^c = r_{21}x + r_{22}y + r_{23}z + T_y, \quad (17)$$

$$z^c = r_{31}x + r_{32}y + r_{33}z + T_z. \quad (18)$$

6.6.4. Posición y orientación

Momentos. Los momentos son funciones definidas en la imagen que pueden ser usados para resumir varios aspectos de la forma y el tamaño de los objetos en una imagen. Estos momentos están denotados por:[17]

$$m_{ij}(k) = \sum_{r,c} r^i c^j I_k(r, c). \quad (19)$$

El centroide de un objeto Es conveniente definir la posición de un objeto como la posición del centro de masa o el centroide. Este está dado por:

$$\sum_{r,c} \vec{r}_i I_i(r, c) = \sum_{r,c} r I_i(r, c) \rightarrow \vec{r}_i = \frac{\sum_{r,c} r I_i(r, c)}{\sum_{r,c} I_i(r, c)} = \frac{m_{10}(i)}{m_{00}(i)}. \quad (20)$$

6.7. Análisis y simulación en software RecurDyn

RecurDyn es un software de simulación dinámico que permite resolver modelos conformados por varios cuerpos, para simular contactos múltiples y cuerpos flexibles. Este utiliza un solver recursivo optimizado para obtener un mejor desempeño y reducción de tiempo a la hora de realizar una simulación. Este software permite realizar pruebas e implementar distintos esquemas de control sin necesidad de implementarlos de manera física.[18]

Dicho software presenta una interfaz amigable al usuario, como se puede observar en la Figura 14.

Este software resulta ser práctico ya que con los conocimientos brindados por la universidad es capaz de exportar archivos con extensión STEP desde el software de Inventor. Este presenta una mayor facilidad para crear los objetos, pero carece de una parte de simulación real de contacto mientras se realiza un análisis dinámico. A pesar de que se pueden importar los archivos de otro software este también presenta sus métodos para realizar las piezas aunque no es recomendable realizarlas dentro de RecurDyn.

Luego de importar las piezas utilizadas es necesario ya sea crear dentro del software un sistema de tierra o establecer a partir de alguna pieza importada que esta sea la tierra. Esto sirve para establecer un marco de referencia inercial. Luego se pueden establecer diversos tipos de juntas para los sistemas. El primer tipo de juntas es el denominado Fixed en el cual establece una restricción de movimiento de un cuerpo con respecto a otro. Se tiene el grupo de juntas traslacional el cual permite establecer juntas traslacionales, cilíndricas, de tornillo o planares. Se tiene un grupo de juntas revolutas en el cual se puede definir también las juntas esféricas y universales. Existe otro grupo que es capaz de restringir el movimiento de sistemas coordinados ya que cada pieza cuenta con diversos ejes generales y un eje central, que es donde se encuentra su centro de inercia. Por último, en el sistema de juntas se tienen los engranajes en los cuales se pueden establecer relaciones de engranajes helicoidales, de piñón y corona e incluso de seguimiento de levas.

Para el grupo de fuerzas se puede establecer un sistema de resortes, una fuerza axial, un resorte rotacional, una fuerza rotacional axial y una fuerza puramente rotacional. En el grupo de fuerzas también se puede establecer un análisis para vigas y simular la deflexión ante esfuerzos que tienen cada una de estas.

Por último, se tiene el grupo para establecer restricciones de contacto. En el cual se tiene el contacto entre sólidos que es el instrumento más importante y en el que se especializa dicho software. Pero, también se pueden establecer restricciones entre dos esferas, dos cilindros o una superficie ya sea contra un cilindro o una esfera.

Todo lo descrito anteriormente se puede observar en la Figura 15.

Luego de establecer las juntas, fuerzas y restricciones se tiene un sistema listo para simular. RecurDyn presenta diversos recursos para realizar estas simulaciones. El recurso más

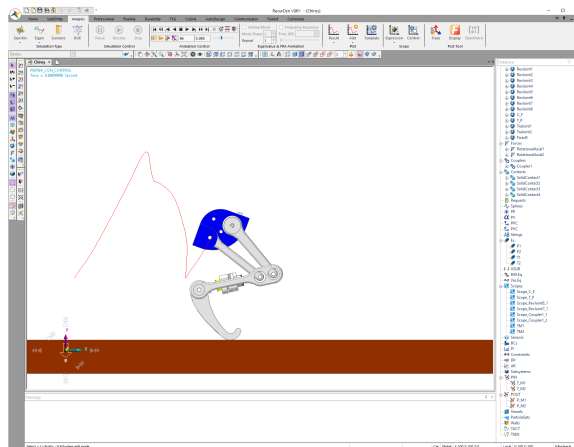


Figura 14: Interfaz general del programa

4. Sirve como modelo para robots con un mayor número de grados de libertad.

Para que el modelo funcione se asume lo siguiente:

- Se tiene una pierna sin masa en donde se puede manejar θ de forma instantánea.
- Se cuenta con una colisión perfectamente elástica.
- La energía siempre se conserva.
- Cuando el pie entre en contacto con el piso este se modelará como una junta revoluta.

6.8.1. Analisis de Poincaré

El modelo contiene una masa m , encima de una pierna con un resorte sin masa con longitud de descanso l_0 y una constante k para el resorte. Para analizar este modelo se toma un ciclo desde la altura máxima hasta la altura máxima presentada por el péndulo invertido.[20]

$$x = \begin{pmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} = 0 \\ \dot{\theta} \end{pmatrix}. \quad (21)$$

Se puede ignorar θ y $\dot{\theta}$ debido al supuesto que este se puede controlar de manera voluntaria. Donde no importa la ubicación en x , por lo tanto, se puede despreciar. Por último se conoce que $\dot{y} = 0$ cuando este llegue a sus puntos de altitud máxima.[20] Para el análisis se consideran las dos fases siguientes.

Primera fase: vuelo

La dinámica del sistema en variables de estado $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$ es la siguiente:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ 0 \\ -g \end{pmatrix}. \quad (22)$$

Segunda fase: contacto (“Stance”)

Para dicha fase se necesita expresar al sistema en coordenadas polares con variables de estado $\mathbf{x} = [r, \theta, \dot{r}, \dot{\theta}]$. Utilizando la física Lagrangiana se describe al sistema de la siguiente manera:[20]

$$m\ddot{r} - mr\dot{\theta}^2 + mg\cos\theta - k(r_0 - r) = 0, \quad (23)$$

$$mr^2\ddot{\theta} + 2mr\dot{r}\dot{\theta} - mgr\sin\theta = 0. \quad (24)$$

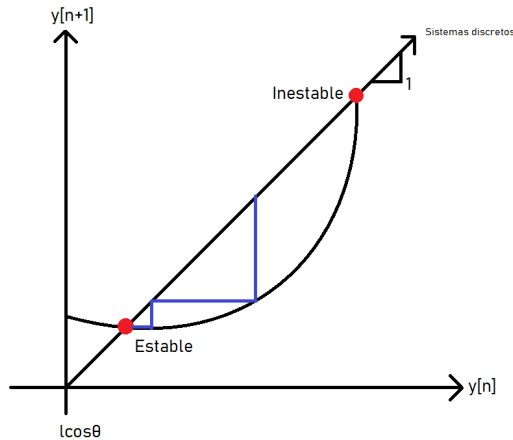


Figura 17: Mapa de cuspide a cuspide del modelo SLIP.

6.8.2. Mapa del modelo SLIP

El sistema cuenta con dos puntos fijos en los cuales se puede analizar la velocidad en y del sistema por medio del diagrama de Poincaré de la Figura 17.[20]

Este diagrama deja de funcionar cuando y se aproxima al suelo. El punto inestable en el sistema indica que existe una velocidad muy baja donde el péndulo invertido, al momento de tener contacto con la superficie, este se caiga hacia atrás en lugar de lograr realizar el movimiento deseado.[20]

6.8.3. Control del modelo

Para realizar un control en el modelo SLIP existen dos soluciones conocidas.

1. Encontrar una entrada u^* sujeta a una posición deseada $y^d = (y^d, u^*)$. Esto se logra linealizando el análisis de Poincaré "P" alrededor de un punto (y^*, u^*) utilizando un LQR discreto en el tiempo.
2. Utilizando un control Deadbeat asumiendo que P es invertible. Esto se logra debido a que el sistema es estable incluso en lazo abierto.

[20]

6.9. Mecánica de Lagrange

Este tipo de mecánica es una reformulación de la mecánica clásica. En esta la trayectoria de un sistema de partículas se deriva resolviendo las ecuaciones de Lagrange de cualquiera de dos maneras existentes. En la primera se resuelve el problema tratando las restricciones como un set de ecuaciones extra mientras que en la segunda el problema se resuelve incorporando las restricciones directamente mediante coordenadas generalizadas.[21]

En general el lagrangiano de un sistema se define mediante:

$$L = T - U, \quad (25)$$

donde T es la energía cinética y U es la energía potencial del sistema. Este en general es una función de un número de coordenadas generalizadas y el tiempo.[21]

El principio de Hamilton indica que el movimiento de un punto del sistema dentro de un rango de tiempo finito es aquel tal que la integral:

$$I = \int_{t_1}^{t_2} L(q_1, \dot{q}_1, t) dt \quad (i = 1, 2, \dots, n), \quad (26)$$

es un mínimo o un máximo de la trayectoria de movimiento. A partir de este principio se obtienen las ecuaciones de Euler-Lagrange:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i, \quad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right)^T - \left(\frac{\partial L}{\partial q_i} \right)^T = Bu. \quad (27)$$

Dichas ecuaciones son prácticas ya que no es necesario descomponer un sistema y analizar las fuerzas internas a comparación de la mecánica Newtoniana.[21]

7.1. Recurdyn

Para caracterizar las extremidades físicas en software se inició importando archivos de Inventor. Dentro de Inventor se exportaron todas las piezas en un ensamble ya que manipular las piezas en Recurdyn no es tan sencillo como lo es en Inventor. Estos archivos se exportaron con la extensión tipo STEP. Ya con los archivos se procede dentro de Recurdyn a importarlos seleccionando el logo principal ubicado en la esquina superior derecha mostrada en la Figura 14. Al haber importado el ensamble es necesario construir una tierra. Para esto se tienen dos opciones, la primera es importar junto con el ensamble de la extremidad una tierra y establecer esta como una o realizar esta dentro de Recurdyn.

Para realizar una tierra dentro de Recurdyn se procede a la pestaña profesional y en el grupo de marcador y cuerpo se selecciona la pestaña general, esto expande un menú en donde está la opción tierra. Al seleccionar esta opción se selecciona la geometría deseada. En este caso se selecciona, dentro del grupo marcador y geometría en la opción de cilindro, la opción box como se observa en la Figura 18. Con esto se seleccionan dos puntos y a partir de ese modelo se modifican las dimensiones de la caja y la ubicación en el espacio para dejarlo debajo de la extremidad. Es importante que este se encuentre lo más cercano a la extremidad ya que el sistema se verá afectado por la gravedad si se deja una distancia con la tierra.

Luego de haber creado una tierra para el sistema se establecen todas las juntas que presenta el sistema. Para ambas extremidades se utilizó principalmente la junta revoluta como se observa en la Figura 19. A parte de la junta revoluta se utiliza también dos juntas traslacionales, estas se colocan de tal manera que se simule la guía que proporciona la plataforma de pruebas. Estas juntas sirven para que la extremidad siga una línea recta y no se desvíe. Luego se pueden colocar juntas tipo Fixed para establecer un mismo movimiento

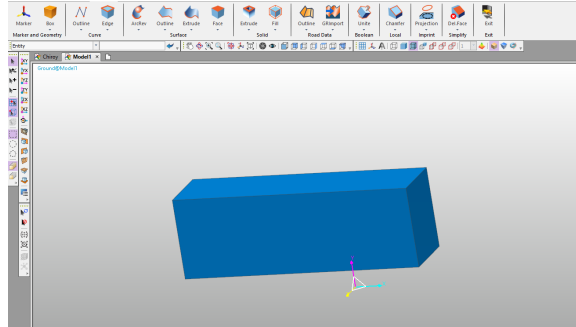


Figura 18: Creación de tierra dentro de RecurDyn.

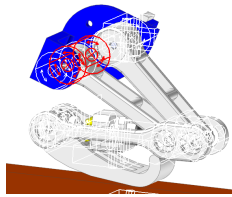


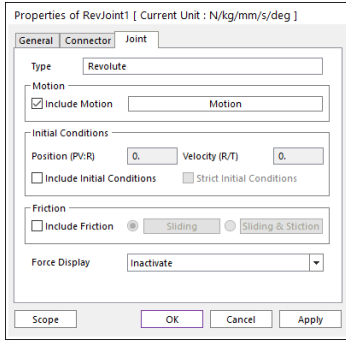
Figura 19: Juntas utilizadas dentro de la extremidad.

entre el motor y la base dentro de la extremidad.

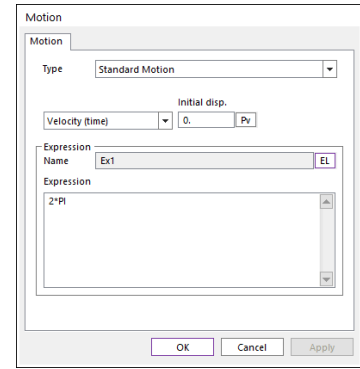
Luego de establecer las juntas para utilizarse dentro del sistema se establecen las restricciones de contacto. Estas restricciones se establecen donde se quiere simular el contacto de las piezas como, por ejemplo: se quiere que exista un contacto entre la tierra y la base de la extremidad para que esta se pueda impulsar. Para probar que las juntas y las restricciones de contacto están funcionando adecuadamente se puede realizar un análisis cinemático. Para esto, dentro del panel base de datos ubicado a la derecha se selecciona la junta en donde se encuentra ubicado el motor. Una vez ubicada la junta se selecciona y se eligen las propiedades dentro de la junta, en la pestaña “junta” se selecciona la caja “incluir movimiento” y se selecciona el botón de movimiento. Dentro de esta se selecciona la lista que tiene desplazamiento y se elige la opción de velocidad(tiempo). Luego se selecciona el botón “EL” dentro de la caja expresión y se crea una función nueva con un valor de $2 \times \text{PI}$. Este procedimiento se puede ver en las Figuras 20a y 20b. Luego dentro de la pestaña de análisis dentro del grupo “tipo de simulación” se selecciona la opción “Dyn/Kin” y se realiza la simulación, animando dentro del grupo control de animaciones seleccionando la opción de reproducir.

Después de verificar que las restricciones de contacto y las juntas estén correctas se procede a quitar el movimiento de la junta y se selecciona una fuerza rotacional axial dentro de la pestaña profesional dentro del grupo fuerza en el subgrupo resorte rotacional. Para esta se seleccionan los dos cuerpos con los que se desea interactuar, la dirección de la fuerza (usualmente es Z) y un punto. Esta fuerza se puede observar en la Figura 21.

Luego se necesita establecer una entrada y una salida de planta como mínimo, pero se pueden establecer diversas entradas y diversas salidas. Para esto se ubica en la pestaña CoLink y se selecciona el botón “Planta de entrada”. Dentro de esta se añade una entrada y se selecciona el nombre deseado. Luego se selecciona el botón “Planta de salida”. Para este se inserta una nueva expresión y dentro de esta pestaña se busca el desplazamiento en Z



(a) Propiedades de la junta revolvente.



(b) Establecer un movimiento dentro de la junta revolvente.

Figura 20: Configuración de una junta revolvente.

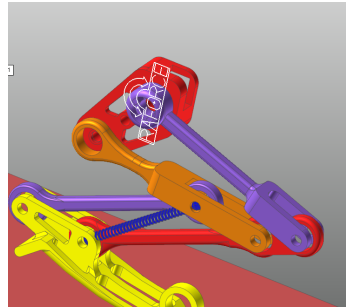


Figura 21: Fuerza rotacional simulando el motor.

que tenga la extremidad o alguna otra medición que se busque obtener. Luego se selecciona el eje coordenado de la fuerza rotacional y se multiplica esto por $180/\pi$ para obtener una medición en grados. Esta expresión resultante se puede observar en la Figura 22.

Luego de haber asignado las salidas y entradas de planta, se asignan las entradas a las fuerzas rotacionales colocadas anteriormente. Para esto se selecciona la fuerza deseada y dentro de la ventana propiedades en la pestaña axial/rotacional se crea una expresión con el módulo Anclar (Entrada Planta). Teniendo ya todas las fuerzas asignadas se puede experimentar con el módulo de Simulink.

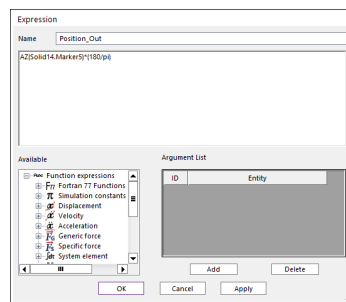


Figura 22: Salida del sistema.

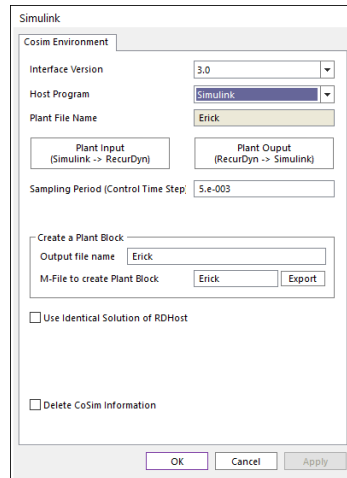


Figura 23: Configuración de Simulink en RecurDyn.

7.1.1. Simulink

Para comunicar RecurDyn con Matlab se tienen que exportar ciertos archivos de RecurDyn. Para esto en la pestaña de comunicación se selecciona la opción Simulink. Dentro de esta es importante establecer dos cosas. La primera es que el programa anfitrión sea Simulink, esto permite que la planta se simule desde glsMatlab y no desde RecurDyn. La segunda opción es el período de muestreo, este número debe ser lo suficientemente bajo como para que exista una cantidad prudencial de mediciones entre ambos programas y lo suficientemente alto para que una simulación no tome demasiado tiempo. Estas simulaciones se deben encontrar dentro del rango de los 60 segundos a 180 segundos. Luego de haber exportado estos archivos se crea un documento con la extensión tipo “.m” y el nombre del proyecto. Esta configuración se observa en la Figura 22

Con RecurDyn abierto, se abre Matlab y se sitúa en la carpeta donde se exportó este archivo. Este se corre y se obtienen diversos parámetros. Para obtener el bloque en Simulink se debe correr la línea de comando **rdblock**. Este permitirá que ambos programas estén funcionando al mismo tiempo. Ya con este bloque se pueden implementar los sistemas de control deseados en Simulink. Es importante mencionar que este bloque debe tener la misma cantidad de entradas y salidas que se establecieron en la planta en RecurDyn.

Luego de establecer el sistema se puede correr la simulación y ver tanto el comporta-

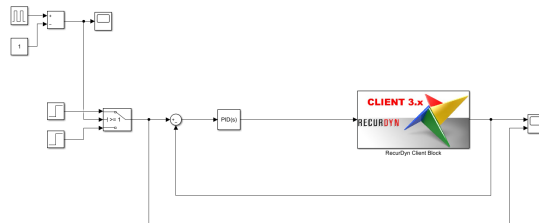
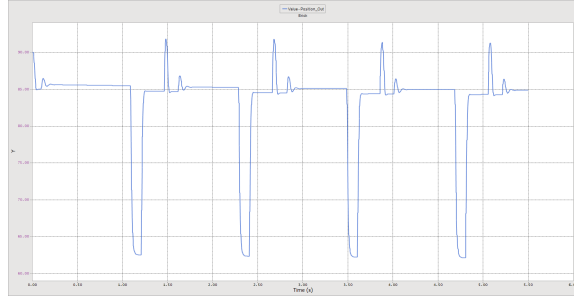
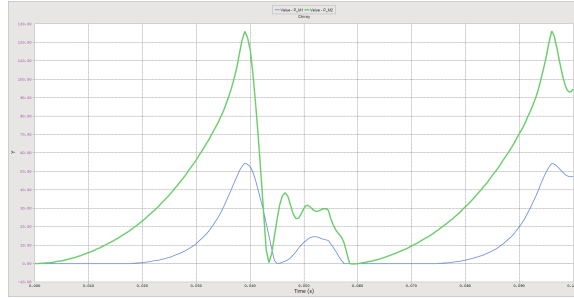


Figura 24: PID creado en Simulink.



(a) Extremidad con un actuador.



(b) Extremidad con con dos actuadores.

Figura 25: Desplazamiento angular resultante.

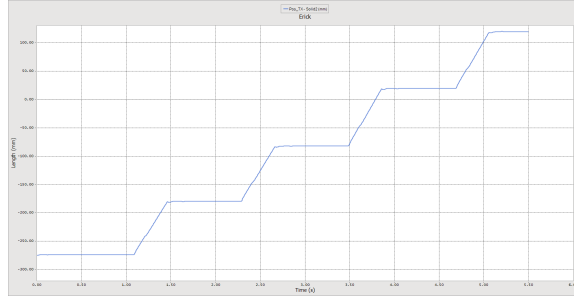
miento en gráficas en Simulink como el comportamiento del sistema dinámico en RecurDyn.

7.2. Resultados

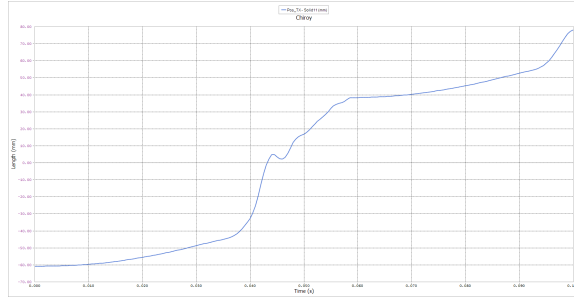
Para establecer una cuantificación se tomaron ciertos resultados del desempeño de la pierna para obtener un punto de comparación. Debido a que se desea comparar el desempeño de ambas extremidades se tomó el desplazamiento vertical y horizontal de las extremidades, la posición angular de los motores y el esfuerzo requerido en los motores para realizar el control. La posición se tomó desde el centro de masa de la parte inferior de la extremidad que tiene contacto con el piso.

7.2.1. Desplazamiento angular

Como se observa en la Figura 25a, se tiene que el ángulo de desplazamiento del motor para la extremidad con un actuador varía de 90 a 63 grados. Al tener esta configuración se observa que el impulso se realiza cuando el actuador pasa de 85 a 63 grados y los picos luego del salto es la energía acumulada por el mecanismo de la pierna. Para la extremidad con dos actuadores en la Figura 25b se tiene que el primer motor ubicado en la cadera tiene un desplazamiento angular leve de 0 a 54 grados, el segundo motor tiene un desplazamiento de 0 a 126 grados en el cual se observa que es donde se realiza el mayor esfuerzo a la hora del salto.



(a) Extremidad con un actuador.



(b) Extremidad con dos actuadores.

Figura 26: Desplazamiento horizontal.

7.2.2. Desplazamiento horizontal

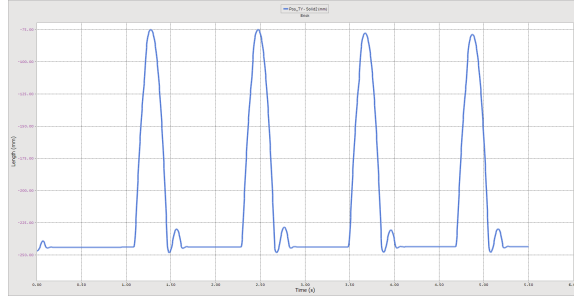
Midiendo el desplazamiento horizontal se observó que para la extremidad con un solo actuador en la Figura 26a se realizan cuatro saltos teniendo un desplazamiento promedio de $94.07mm$ por salto. La extremidad con dos actuadores en la Figura 26b realiza un único salto con un desplazamiento horizontal de aproximadamente $99.16mm$

7.2.3. Desplazamiento vertical

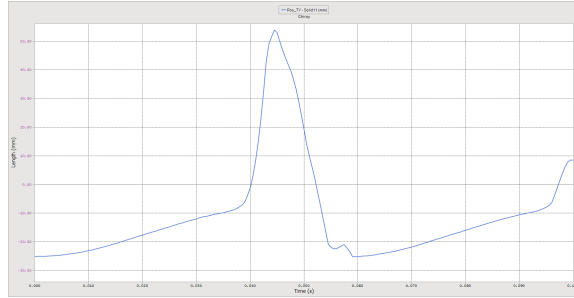
A partir de la comparación mostrada en la Figura 27 se observó la diferencia de distancias verticales que ambas extremidades son capaces de alcanzar. En la Figura 27a se observó que el desplazamiento máximo alcanzado por la extremidad es de $168.59mm$ mientras que el desplazamiento alcanzado por la extremidad con dos actuadores es de $79.09mm$.

7.2.4. Esfuerzos de control(Torque)

En la figura 28 se tiene la comparación de los torques necesarios de entrada para realizar el salto. Comparando ambas extremidades se observó que la extremidad con un solo actuador requiere menos esfuerzo para realizar el salto. Se mostró en la Figura 28b que ambos actuadores necesitan un esfuerzo constante. El segundo motor necesita un esfuerzo de aproximadamente 0.04 segundos y el primer motor necesita un esfuerzo de aproximadamente 0.028 segundos. Mientras en la Figura 28a se muestra que el esfuerzo máximo requerido para movilizar la extremidad es menor y requiere un intervalo de tiempo de aproximadamente



(a) Extremidad con un actuador.



(b) Extremidad con dos actuadores.

Figura 27: Desplazamiento vertical.

0.03 segundos.

7.2.5. Discusión

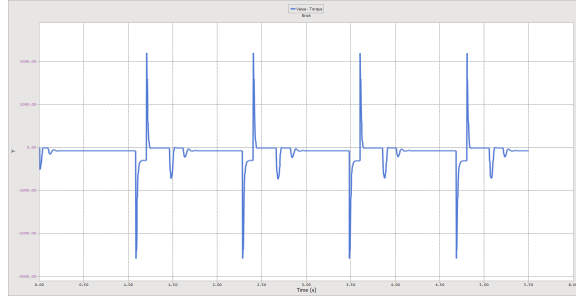
Para ambas extremidades se utilizó un control clásico por PID. Para la extremidad con un único actuador se utilizó un PID tanto para realizar el salto como para la caída. Para este se utilizó una señal de referencia cuadrada que varía entre 60 y 85 unidades. Mediante simulación se encontró que el período de esta señal cuadrada debía ser de 1.2 segundos para lograr realizar una serie de pasos continuos. Esta sincronización al igual que la extremidad con dos actuadores se realizó únicamente con tiempos.

Los valores PID encontrados para el funcionamiento de la extremidad con un único actuador fueron: para la variable proporcional un valor de 100 unidades, la variable integral un valor de 30 unidades y la variable derivativa un valor de 1 unidad.

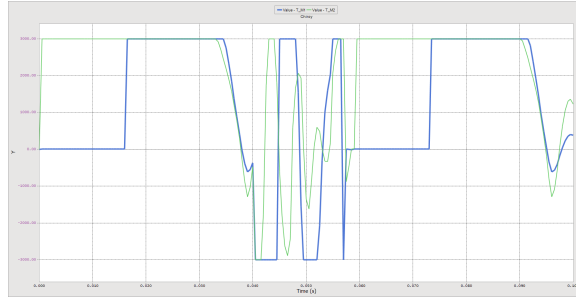
Para la extremidad con dos actuadores se utilizaron dos controladores PID para controlar el salto y la caída independientemente cambiando la referencia de manera distinta para cada controlador. Para este en el momento de la caída se utilizó únicamente un control proporcional.

| | K_p | K_I | K_D |
|-------------------|-------|-------|-------|
| Actuador en cóxis | 140 | 2 | 300 |
| Actuador en tibia | 80 | 0.45 | 52 |

Cuadro 2: Constantes PID utilizadas para el salto en extremidad con dos actuadores.



(a) Extremidad con un actuador.



(b) Extremidad con dos actuadores.

Figura 28: Esfuerzo de control.

| | K_p | K_I | K_D |
|-------------------|-------|-------|-------|
| Actuador en cóxis | 710 | 0 | 0 |
| Actuador en tibia | 160 | 0 | 0 |

Cuadro 3: Constantes PID utilizadas para la caída en extremidad con dos actuadores.

Con estos resultados se pudieron comparar directamente ambas extremidades y de manera general se pudo observar que se tiene un mejor desempeño con la extremidad que cuenta únicamente con un actuador. Esta es capaz de recorrer una distancia horizontal similar a la extremidad con dos actuadores, sin embargo es capaz de alcanzar el doble de la altura por salto y utiliza un menor torque de control debido al mecanismo del resorte que es capaz de almacenar energía.

Sin embargo, debido a que el mecanismo del resorte es un elemento pasivo, este tiene un desgaste que involucra reparar de manera frecuente la extremidad. Esto no es el caso de la extremidad con dos motores, en la cual si se realiza a partir de los materiales requeridos esta provee una gran versatilidad para controlar el mecanismo y presenta un menor desgaste.

El mecanismo con un único motor presenta una gran ventaja ya que este no requiere de la sincronización de ambos actuadores para que funcione de manera correcta.

Plataforma de programación.

Durante este trabajo de graduación se decidió implementar una nueva plataforma para implementar el control de las extremidades. Esto se realizó debido a que la primera fase implementaba un controlador que funcionaba únicamente con dos motores mientras la segunda fase únicamente implementaba un tren de pulsos para realizar el salto habilitando uno de los pines digitales con un único motor.

Se eligió la plataforma integrada de Texas Instruments, Tiva C Series TM4C123G debido a la alta frecuencia de reloj (80MHz) y la librería implementable para el controlador PID de CMSIS-DSP.

La plataforma se programó de tal manera que se pudieran manejar las dos extremidades apagando o encendiendo las salidas deseadas y capaz de tener dos controladores PID totalmente independientes capaces de cambiar la referencia y las constantes para el salto y la caída. Este programa también fue capaz de elegir el modo de salto de la extremidad, ya sea que este salte de manera consecutiva sin pausar o por medio de comandos por medio del módulo UART integrado en la Tiva C para realizar un salto a la vez.

Un factor importante para tomar en cuenta es la relación de engranajes que se tiene dentro del motor. Factores comunes que se encuentran son relaciones de 50:1, 75:1 y 100:1. Esto influye para que el módulo de cuadratura logre colocarse en una posición adecuada.

A continuación, se describe la serie de librerías que se utilizaron para llevar a cabo el programa y su funcionalidad dentro de este.

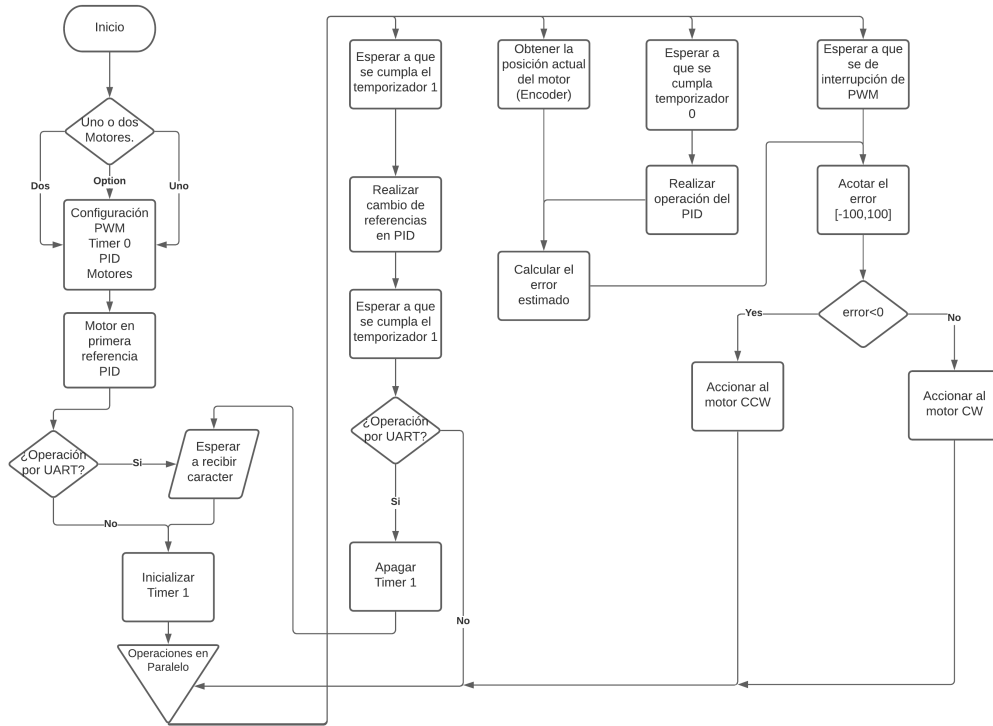


Figura 29: Diagrama de flujo indicando la lógica detrás de la programación.

8.1. UART

El módulo UART se utilizó principalmente para debug en el programa debido a las limitaciones del equipo por la pandemia del COVID-19 del año 2020. Sin embargo, se aplicó la funcionalidad de un único salto enviando un carácter por medio de UART. Esto se logra utilizando un programa para comunicación serial. El programa utilizado para enviar comandos fue “RealTerm”.

Una vez se recibe un carácter por medio de UART, se da una interrupción para habilitar los temporizadores y dependiendo de los motores que se tengan configurados se habilita por motor la configuración del encoder de cuadratura, la generación por medio del tren de pulsos y la configuración del PID.

Este módulo solo inicia la operación, pero el módulo encargado de apagar la dinámica es el timer 1.

8.2. CMSIS DSP

La siguiente librería [22] se utilizó para implementar un controlador PID de manera eficiente y de forma breve.

Esta librería se decidió implementar en modo de punto flotante para obtener una ma-

por precisión, pero existe la posibilidad de implementar en punto fijo para incrementar la eficiencia del programa.

8.3. Timer 0

Dentro del módulo del PID se encontró que los cálculos para el comando “arm_pid_f32” se debían realizar cada cierto intervalo ya que el microcontrolador presentaba problemas para realizarlo de forma consecutiva. Debido a esto se implementaron interrupciones mediante el timer 0. A este se le carga cierto valor dado por:

$$\text{Timer} = \frac{\text{SysCtlClockGet}}{\text{Factor}}, \quad (28)$$

donde timer es el tiempo total que transcurrirá antes que se dé la interrupción, *SysCtlClockGet* es el comando para que se de el valor que tiene el reloj del microcontrolador y Factor es una constante con la que se indica la frecuencia de interrupciones por segundo. Dentro del programa se encontró que este valor de “Factor” estabiliza al controlador PID dentro de un rango entre 30-45 veces por segundo. Incrementar este valor provoca que el sistema se tenga que calcular una mayor cantidad de veces de las que el microcontrolador puede físicamente y disminuir el valor causa que se aumente el error de sobremanera haciendo que el controlador se vuelva inestable.

8.4. Timer 1

El módulo de otro timer se utilizó para hacer el cambio de referencias y de constantes para el PID. Este controlador utiliza la forma de carga de tiempo de manera similar al timer 1. Donde un factor estima la cantidad de interrupciones que se dan en un segundo. Este valor se debe modificar para cada extremidad de manera independiente midiendo los tiempos de subida y aterrizaje respectivamente.

Para este módulo también se implementó un contador de las interrupciones para cuando se activa la funcionalidad de salto mediante UART esta apague todos los módulos una vez haya completado un salto.

8.5. Interfaz de encoder de cuadratura

El módulo de cuadratura se configuró para el funcionamiento directo con los motores. Para este es posible configurar cualquiera de los dos módulos existentes y que estos tengan un funcionamiento separado.

La configuración de los encoder tiene un requerimiento importante directamente relacionado con los motores. Se encontró que al utilizar motores con una relación de engranajes de 50 : 1 y 100 : 1 se debe realizar un cambio en la lectura de las señales. Por lo tanto,

si se desean realizar pruebas con esta relación de motores se debe cambiar la configuración según sea el caso con las palabras de configuración: “QEI_CONFIGURE_SWAP” o “QEI_CONFIGURE_NO_SWAP”.

Dentro de la configuración del encoder de cuadratura se establecen los límites de giro del motor dada por la relación:

$$\text{Límite} = \text{Relación del motor} \times \text{Pulsos por vuelta del Encoder} \times 2, \quad (29)$$

en donde la relación del motor es el número de vueltas que se debe dar en el eje ubicado en el encoder para que el eje en el motor gire una vez y los pulsos por vuelta del encoder es la resolución que tiene dicho encoder dada en la hoja de datos. Para las pruebas de ambas extremidades se utilizaron encoder de 12 pulsos por vuelta y una relación del motor de 75 : 1. Este factor se multiplicó por dos para obtener un rango de 0 a 720 grados.

Una consideración importante es que al momento de realizar la lectura por medio del módulo del encoder de cuadratura es que este devuelve el número en pulsos. Para convertir esto a grados se tomó la siguiente ecuación:

$$\text{Pos. Actual} = \text{QEIPositionGet} \times \left(\frac{360.0 \text{ deg}}{\text{Relación del motor} \times \text{Pulsos por vuelta del encoder}} \right). \quad (30)$$

Por último, debido a que los sistemas pueden involucrar un giro en sentido contrario de las agujas de reloj se establece como posición inicial un ángulo de 360°. Esto provoca que las posiciones deseadas en la extremidad se tengan que dar referenciadas a este ángulo y se pueda girar libremente sin ninguna restricción.

Este módulo no utiliza interrupciones.

8.6. PWM

El microcontrolador cuenta con 14 módulos (pares) para generación de ondas cuadradas. Este se divide en dos bases (0 y 1), teniendo 7 módulos por base. El módulo PWM se implementó desde la misma base para controlar ambos motores.

Este módulo se utilizó para que por cada interrupción que se diera del PWM (utilizando un método de conteo hacia un cero) se actualiza el valor del resultado del cálculo de error PID acotado en un rango entre $[-100, 100]$. Si el resultado resulta ser negativo se invierte el sentido de giro en contra de las agujas del reloj.

Para introducirlo se estableció en la configuración un período en el que opera el PWM a partir de la siguiente ecuación:

$$\text{PWM_Period} = \frac{\text{SysCtlClockGet}}{\text{Desired Period}}, \quad (31)$$

en donde “SysCtlClockGet ” es la frecuencia de reloj utilizada en el microcontrolador y el período deseado es la frecuencia deseada del PWM. Para el período deseado se estableció una frecuencia de 10kHz.

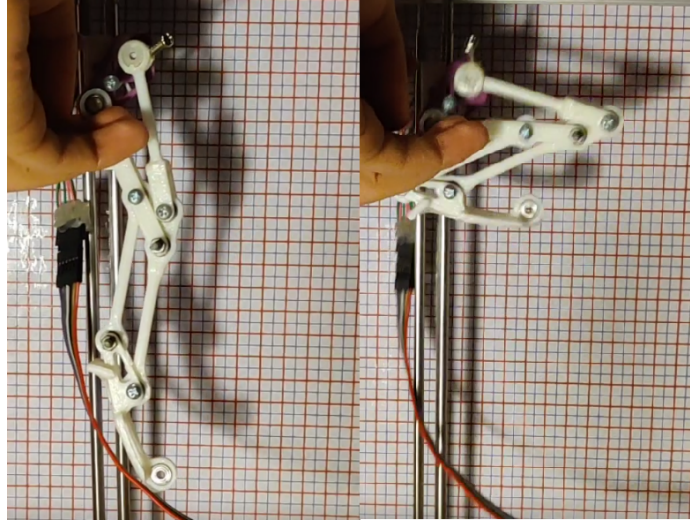


Figura 30: Salto en extremidad de la segunda fase sin restricciones de contacto.

8.7. Validación de la programación

Para validar la programación realizada en el entorno del microcontrolador se ensamblaron las extremidades y se probaron los controladores con constantes del PID previamente utilizadas.

Durante estas pruebas se observó que la programación tenía problemas de estabilidad en ambos controladores PID. Luego de cierto tiempo se incrementaba de sobremanera el error y el motor comenzaba a girar sin control. Debido a que los motores no operan dentro del voltaje nominal de $6V$ si no un voltaje de $12V$, al momento de girar sin control y encontrar limitaciones mecánicas se tuvo un alto esfuerzo y provocaba que los motores se quemaran. Este error se corrigió dentro de la programación, pero se debe tener una alta precaución del consumo de corriente de la extremidad para no forzar a los motores constantemente. Este trae problemas de temperatura por un uso prolongado dañando las piezas y al motor mismo. Para realizar las primeras pruebas en la extremidad se tuvo que limitar la corriente en caso los motores llegaran a un tope por un tiempo prolongado.

Durante la programación se encontró que el controlador no se debía apagar al llegar a su posición final cuando el error se hiciera cero si no este debía permanecer encendido debido a que la extremidad de la primera fase no permanece en una posición de descanso. Finalmente se obtuvieron resultados satisfactorios en donde ambas extremidades presentaban saltos en el aire sin restricciones de contacto.

8.8. Esquema de conexiones

En el siguiente esquema se presenta el diagrama de conexiones utilizado para operar ambas extremidades:

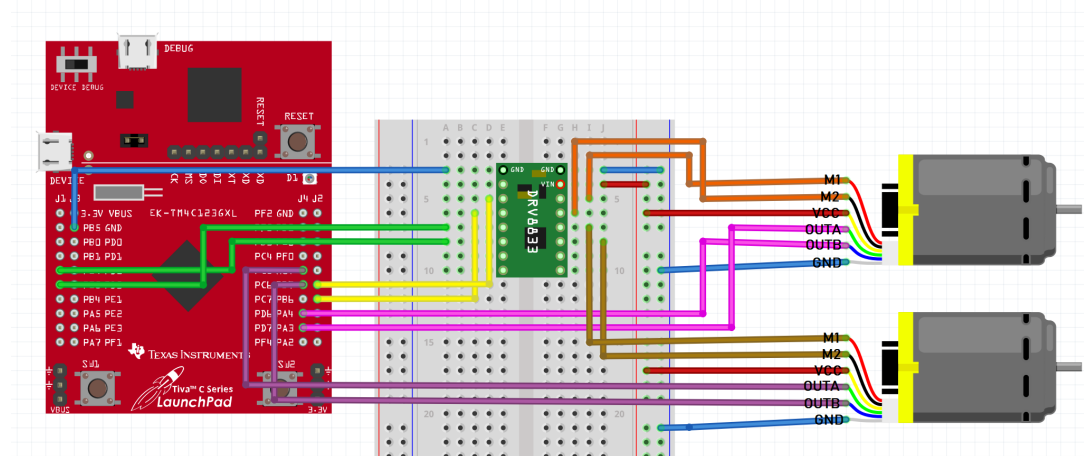


Figura 31: Circuito implementado.

8.9. Control de versiones y estándar de programación

Debido a las múltiples versiones manejadas para manipular dos extremidades distintas se implementó un sistema de control de versiones por medio de GitHub. Esto permitió la creación de nuevas ramas independientes al programa principal y luego unir estos dos módulos para obtener una operación sin errores y organizada.

Junto a este sistema se implementó un estándar de programación para sistemas embebidos. El estándar desarrollado por Michael Barr dio pautas para realizar una estructura de código capaz de prevenir errores en la programación y agregar una directriz importante para permitir que este mismo sea trabajado por distintas personas.

8.10. Unidad de medición inercial

A la programación se le agregó un módulo para realizar operaciones con una unidad de medición inercial capaz de obtener ángulos de giro junto a mediciones de acelerómetros para eliminar el uso de los temporizadores. Esto evita que se dé una sincronización del salto por medio de tiempos. La operación y comunicación de este módulo se dio por medio de I2C, en el que se accedió a los registros de este módulo y luego se realizaron cálculos por cada vez que se obtienen dichas mediciones.

Al agregar la unidad de mediciones inercial se obtuvo que la extremidad contando la operación de los controladores tenía una capacidad máxima de actualización de 15 veces por segundo para la unidad de mediciones inercial y para los motores una actualización de

100 veces por segundo. Esto presentó el problema que el microcontrolador no es capaz de realizar una fusión de sensores para eliminar la sincronización por tiempos.

Rediseño de las extremidades

Para validar la programación realizada en el capítulo previo se ensamblaron las extremidades junto con la plataforma de pruebas. Sin embargo, se encontró con una serie de problemas que extendieron el período de pruebas. Debido a que se planea trabajar en un futuro con dichas extremidades se realizaron modificaciones en el diseño de las extremidades y a lo largo del proyecto se fueron corrigiendo errores en el diseño.

9.1. Trayectoria de correcciones a la fase 1

Debido a que esta extremidad no se retomó en la fase previa al momento de realizar el ensamble de la pieza se observó un desgaste y fracturas que afectarían al desempeño de la extremidad. Para esto se decidió hacer una reimpresión de la extremidad utilizando los archivos ya existentes. Con esto se encontraron una serie de problemas en la fase de diseño.

Dicha extremidad cuenta con una transmisión de potencia por medio de engranajes cónicos los cuales fueron impresos en 3D. Debido a que el material de impresión PLA cuenta con una alta ductilidad el usuario anterior modificó la pieza sin tomar en cuenta la transmisión de potencia. El primer cambio realizado a la extremidad fue diseñar nuevamente esta transmisión de potencia. En esta se implementó un engranaje anclado al pie que contara con una hendidura para transmitir la potencia adecuadamente. Este cambio se tomó en cuenta debido a que los engranajes se diseñaron como un elemento de sacrificio y separar estas dos partes permite una mejora considerable en el tiempo de impresión.

Con esta modificación se decidió imprimir la extremidad completa con dos materiales distintos. El primero es el previamente utilizado PLA y el segundo material fue la resina debido a la alta rigidez del material. Utilizando las impresoras del departamento de ingeniería mecatrónica de la universidad se encontró con el problema que la impresora de resina tenía

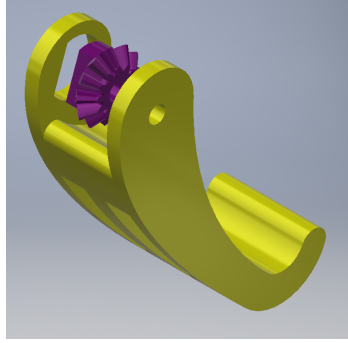


Figura 32: Modificaciones realizadas a la transmisión de potencia.

la dificultad para imprimir ciertas piezas ya que por el tratamiento estas se doblaban y sobre todo la impresora imprimía las extremidades con un factor de escala por lo tanto no se podían combinar las piezas impresas en PLA y piezas impresas en resina.

Utilizando únicamente las piezas impresas en PLA se encontró con el problema que el usuario de la fase anterior utilizó cilindros pequeños hechos de madera como ejes para la extremidad y al tratar de retirar los ejes estos se rompían. Por lo tanto, se decidió implementar un estándar en los agujeros en los cuales se implementaron en su mayoría tornillos M3 con sus respectivas tuercas. Al implementar este estándar de tornillos se observó que estos interferían con los ejes de la plataforma de pruebas por lo tanto se tuvo que rediseñar el montaje dejando una distancia entre la pierna y el eje como se observó en la Figura 33.

Otro problema encontrado dentro de la primera impresión fueron las tolerancias manejadas por la impresora. Debido a que los cojinetes debían entrar a presión se limitaron las dimensiones, pero en las nuevas piezas impresas estos cojinetes no entraban. Para esto se deseaba conocer las dimensiones actuales para revisar discrepancias entre el diseño y los cojinetes reales debido a que estos no se incluyeron dentro del software en la fase previa.

Por falta de herramientas para medición de precisión se obtuvo que los cojinetes eran de marca “NSK” y se buscó dentro del catálogo de esta marca para revisar que las dimensiones aproximadas tomadas con una regla coincidieran con algún tipo de rodamiento. Para dicha fase se encontró que los rodamientos tienen el código siguiente:

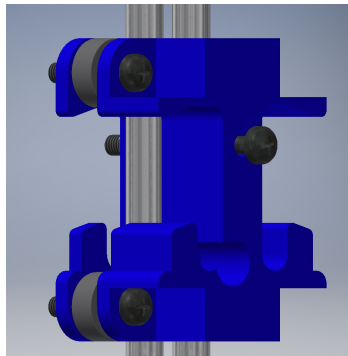


Figura 33: Modificación al sistema de montaje de la extremidad.



Figura 34: Fractura de dos dientes debido a estrés.

Rodamientos extremidad fase 1("623ZZ").

- Diámetro interior de $3mm$.
- Diámetro exterior de $10mm$.
- Ancho de cojinete de $4mm$.

La ventaja de encontrar dichos cojinetes dentro del catálogo fue que estos presentaban modelos CAD implementables en Inventor. Estos proporcionan una alta fidelidad en términos de diseño, peso e inercias.

Debido a la falta de piezas en resina se imprimió una segunda vez modificando los archivos para que tuvieran distintas tolerancias. Sin embargo, se optó por imprimir en nuevas impresoras de resina brindadas por el catedrático encargado del MakerLab Pablo Mazariegos. Con estas nuevas piezas se pudieron realizar las primeras pruebas a la pierna. Durante las pruebas se encontró que los engranajes impresos en resina no eran prácticos ya que debido a la rigidez que presenta este material este tiende a ser altamente frágil. Este provocó que algunos dientes se fracturaran como se observa en la Figura 34. Por lo tanto, se decidió utilizar engranajes fabricados de PLA para realizar las primeras pruebas.

Luego de esto se encontró que el modo de funcionamiento de la extremidad es de permanecer en un lugar intermedio entre la posición de salto y posición de descanso.

Utilizando eslabones de PLA se encontró el problema que el impacto entre el motor y los eslabones deforman la pieza lo cual hace que el algoritmo de programación pierda su orientación debido al juego que lentamente se forma de este.

Para este se recomienda imprimir los eslabones encargados de transmitir la potencia a

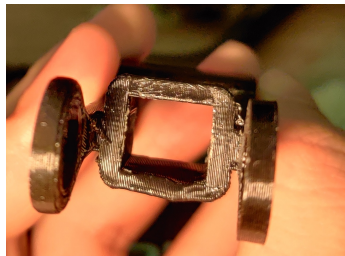


Figura 35: Deformación debido a temperatura.

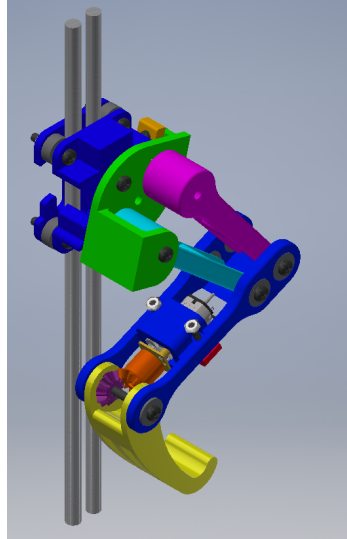


Figura 36: Ensamble realizado en software.

partir de un material llamado PETG. Este material industrial tiene una alta capacidad de absorber los impactos sin presentar deformación, aunque este lentamente presentará cierto desfase en la transmisión de ángulos.

Debido a que los motores no operan dentro del voltaje nominal de $6V$ si no un voltaje de $12V$, al momento de girar sin control y encontrar limitaciones mecánicas estos luego de operarse un par de veces manejan corrientes altas lo cual provocan un aumento en la temperatura del motor. Se encontró que el eslabón encargado de mantener rígido al motor para la transferencia de potencia por los engranajes cónicos presentaba deformaciones por calor como se puede apreciar en la Figura 35. Por lo tanto, se recomienda imprimir este eslabón en ABS debido a su alta resistencia a la temperatura. Este es capaz de soportar hasta $85^{\circ}C$.

9.1.1. Actualización en software.

Finalmente se encontró que dentro del software el ensamblaje completo de la extremidad se encontraba desactualizado a todas las modificaciones realizadas actualmente. Por lo tanto, se decidió empezar un ensamblaje dentro del software implementando los cojinetes, tornillos y cambios a la extremidad limpiando todos los componentes no utilizados. Además de la alta fidelidad que proporciona este ensamblaje nuevo para realizar simulaciones, se pueden obtener listas de los tornillos utilizados. Esta lista se proporciona en el apéndice.

9.1.2. Diseños futuros.

Debido a los problemas encontrados con el diseño actual y las modificaciones realizadas implican la compra de nuevos materiales de impresión para extender el ciclo de vida de las piezas se sugiere emplear piezas maquinadas en aluminio para poder emplear nuevamente solo piezas en PLA pero que no presenten dicho desgaste.



Figura 37: Pieza sugerida para la transmisión de potencia.

Para esto se recomienda utilizar un castigador en la transmisión de potencia del motor ubicado en la cadera similares a los encontrados en la Figura 37.[23] Similar a esto se recomienda utilizar engranajes cónicos maquinados en acero o en bronce que tengan la opción de limitar el movimiento para obtener una mejor transmisión de potencia similar a los que se encuentran en la Figura 38.[24]

9.2. Trayectoria de rediseño en la segunda fase

Con dicha extremidad se trataron de hacer pruebas desde un inicio con las piezas utilizadas en la fase previa. Pero este tenía el inconveniente que presentaba una gran cantidad de fracturas en las piezas por lo tanto se mandaron a manufacturar las piezas fracturadas nuevamente. Debido a que se siguió la misma trayectoria en donde los problemas de impresión fueron los mismos.

Para este caso el código de los cojinetes utilizados fue el siguiente:

Rodamientos extremidad fase 2.(“693JZZMC3E”):

- **69:** Single row deep groove bearings.
- **3:** Bearing bore $3mm$.
- **J:** Pressed steel ribbon cage.
- **ZZ:** Shields on both sides.



Figura 38: Engranajes cónicos.

- **MC3:** Upgrade normal clearance.
- **E:** Quieter than standard.
- Diámetro exterior de $8mm$.
- Ancho de cojinete de $4mm$.

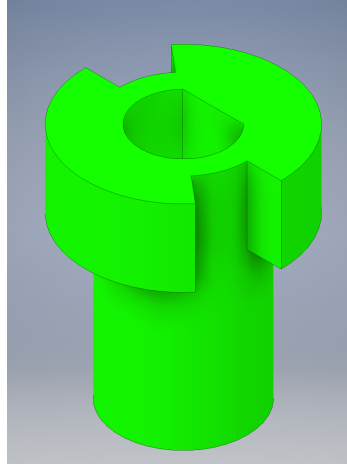


Figura 39: Pin para transmisión de potencia.

Luego de obtener nuevamente las piezas se realizaron las primeras pruebas en donde se encontró el mismo problema donde se perdía la referencia de la posición del motor debido al juego que contenía la pieza encargada de transmitir la potencia del motor. Al momento de revisar las piezas en software no se encontró dicha pieza por lo tanto se tuvo que diseñar completamente la pieza.

Durante la realización de las pruebas el hule utilizado como elemento de regeneración de energía se reventó debido a la antigüedad por lo tanto se trató de conseguir un resorte junto con la estandarización de los tornillos M3. Este resultó tener un coeficiente Hooke muy alto y por lo tanto los motores no eran capaces de estirar el resorte. El resorte contaba con una longitud inicial de $25.97mm$ y el calibre del alambre era de $0.5mm$. Luego se procedió a utilizar hules nuevamente ya que se consultó con distintos proveedores y ninguno era capaz de realizar un resorte con un menor calibre ni de menor rigidez. Para iniciar las pruebas del elemento de regeneración se tomó un hule que resultó ser altamente rígido, sin embargo, se lograba la una ligera compresión de la extremidad. Este no era suficiente para lograr producir un salto. Con un nuevo set de pruebas se compraron 5 bolsas distintas de hules y un set de múltiples resortes de compresión y elongación. Sin embargo, con este nuevo set de resortes se encontró un resorte que lograba cumplir los requerimientos del motor. Finalmente se realizó un nuevo ensamble para la extremidad en donde se incluyeron los cojinetes, tornillos y piezas modificadas para obtener una mayor fidelidad.

9.2.1. Resultados del resorte

Para el resorte con resultados satisfactorios se realizaron ciertas mediciones para encontrar distintos parámetros junto con una prueba de gravedad para encontrar la constante del

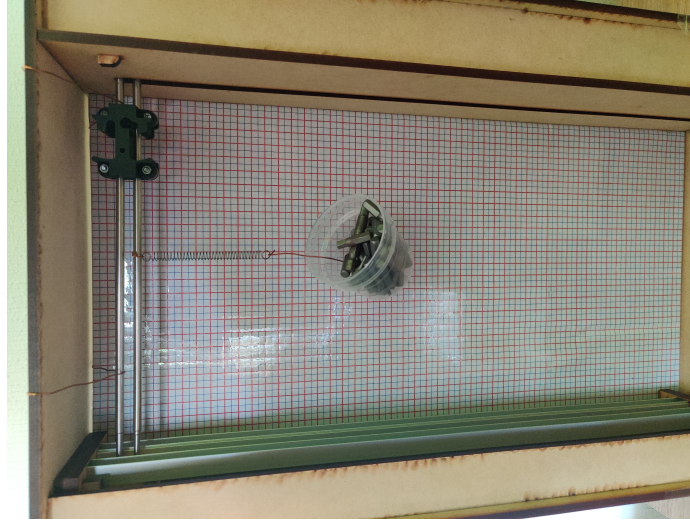


Figura 40: Montaje para obtener constante del resorte.

resorte. Los parámetros se midieron con un vernier para garantizar una medición adecuada. En donde la longitud inicial fue de 13.65mm, el diámetro exterior fue de 5.92mm y el calibre del alambre fue de 0.4mm.

Para calcular la constante del resorte se utilizó la plataforma de pruebas de las extremidades en la cual se limitó el movimiento de los ejes y se colgó al resorte de modo que este se lograra estirar únicamente en un eje. En este se preparó un recipiente con un alambre para colgarlo en el extremo restante del resorte. La masa del recipiente con el alambre era de 11g.

El experimento consistía en introducir un peso conocido dentro del recipiente e ir midiendo la elongación del resorte. Para medir el peso se utilizó una pesa de cocina con una precisión de $\pm 1g$. Como peso se utilizaron puntas de desatornillador y en promedio estas tenían un peso de 5g por lo tanto se iban agregando 4 o 5 puntas por medición. Los resultados obtenidos restando la longitud inicial y tomando en cuenta la gravedad para obtener el peso fueron los encontrados en el Cuadro 4.

| Peso(N) | L(mm) |
|---------|-------|
| 0.00000 | 00.00 |
| 0.37278 | 3.86 |
| 0.61803 | 10.18 |
| 0.84366 | 15.82 |
| 1.10853 | 22.18 |
| 1.36359 | 28.75 |
| 1.62846 | 34.48 |
| 1.86390 | 40.22 |
| 2.08953 | 45.26 |
| 2.52117 | 54.60 |

Cuadro 4: Resultados del experimento.

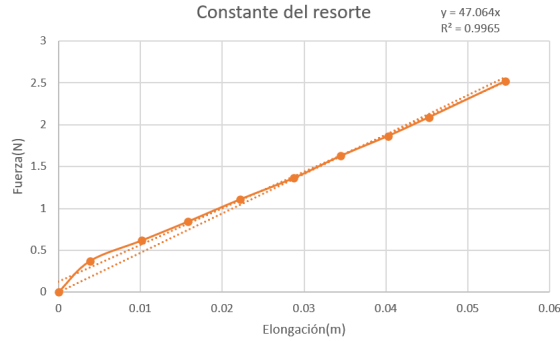


Figura 41: Comportamiento del resorte a diferentes pesos.

Utilizando la ley de Hooke y tomando la longitud en metros se graficó el resultado y se obtuvo la pendiente de la recta debido a que se observó un comportamiento lineal.

Con esto se estableció que la constante del resorte es de 47.064N/m.

9.3. Parametrización para impresión en 3D

Uno de los problemas encontrados durante la impresión y fabricación de piezas fue que cada impresora maneja tolerancias distintas esto provoca que se tenga que modificar pieza por pieza cada vez que se desee manufacturar la pieza. Dicho esto, para ambas extremidades se modificaron todas las dimensiones para incluir una parametrización la cual implementa una tolerancia especificada y en el caso de agujeros o diámetros esta tolerancia se ve multiplicada.

Para realizar esta parametrización se conectaron las piezas directamente con el ensamble y se creó un componente de lógico dentro de inventor para igualar las tolerancias parametrizadas de cada pieza a una sola variable.

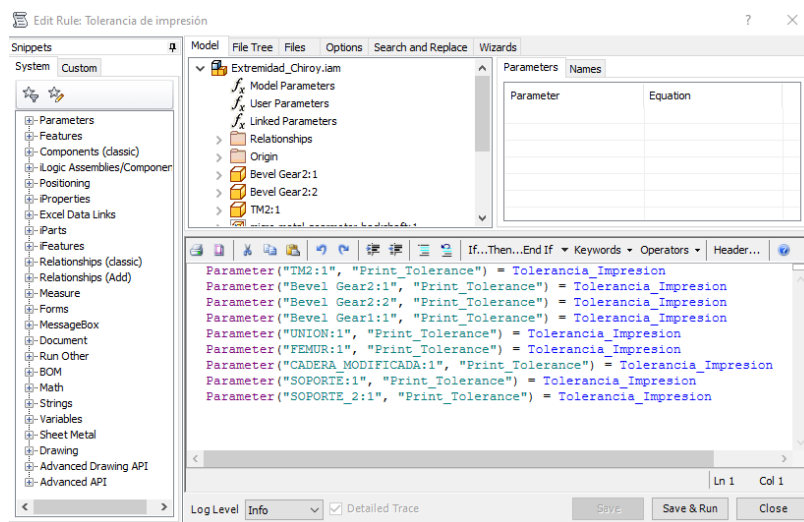


Figura 42: Regla de parametrización.

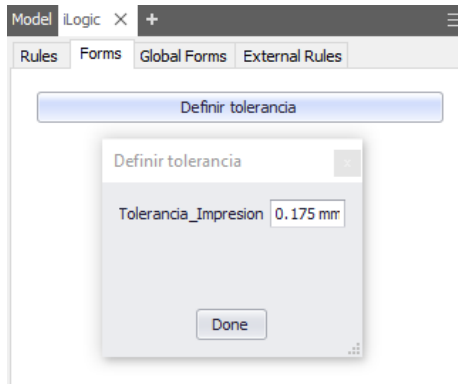


Figura 43: Cuadro de diálogo para modificar tolerancias.

Por último, se agregó una forma en donde se solicita la tolerancia de la impresora para modificar la variable establecida y así cambiarla en cada pieza.

Esta parametrización facilitará futuras impresiones que se realicen de las piezas e implementa una mayor versatilidad para imprimir en diferentes materiales y distintas impresoras.

9.4. Modificaciones a la fase 2

Debido a problemas presentados en el desempeño de la primera fase de la extremidad se decidió, en pruebas posteriores, a continuar únicamente con la fase 2. Se observó que aún con las nuevas piezas impresas este mecanismo presentó un juego capaz de desfasar a un controlador de referencias. Para este se implementó un sistema por medio de castigaderas. En este se introdujo un tornillo M2.5 sin embargo no se obtuvieron resultados satisfactorios. Además como se observa en la Figura 44 se agregó un módulo para sujetar una unidad de mediciones inercial.

Con este método de castigaderas se observó que la extremidad aún generaba impactos

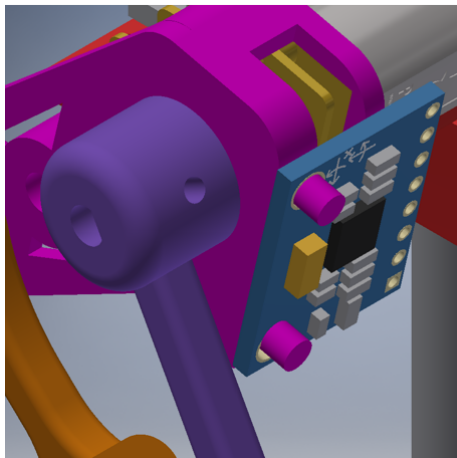


Figura 44: Modificaciones a la fase CAD.

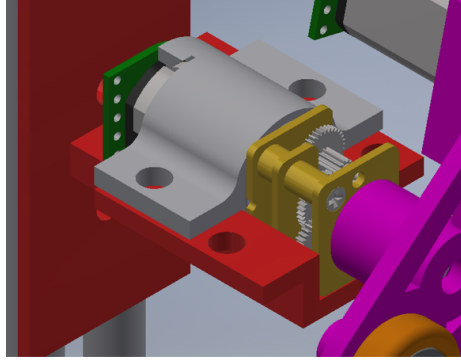


Figura 45: Implementación de actuador en la cadera.

significativos capaces de deformar al sistema, por lo tanto, la última operación realizada para fijar definitivamente a las juntas revolutas implicó el uso de pegamento industrial de secado rápido. Al solidificarse este material se encontró que el sistema no presentó fallas y fue capaz de utilizarse para realizar pruebas y obtener resultados.

Por último, se agregó una modificación propiamente para mejorar el desempeño de la extremidad. Para este se adicionó un motor con ángulo modificable para el sistema de la cadera. Este tuvo la finalidad de validar las simulaciones obtenidas en el cual como se observa en la Figura 45, implementa un mecanismo desde el sistema de montaje de la extremidad encapsulando al motor con protectores.

Para obtener una validación correcta del funcionamiento de las extremidades y entender el porqué del funcionamiento de los sistemas de control se fabricaron modelos matemáticos complejos y modelos existentes de sistemas dinámicos que se caracterizan por realizar saltos en una extremidad. Estableciendo lo siguiente ambas fases de diseño tienen sus diferencias. Por lo tanto, ambas pruebas y simulaciones se realizaron por separado.

10.1. Fase 1

Debido a la complejidad que presenta la extremidad al saltar esta se dedujo como un modelo de tres eslabones con dos torques en las juntas, modelando una junta con torque cero como el contacto de la extremidad con el suelo. El modelo se describe a continuación.

10.1.1. Modelo matemático fase 1

Para obtener el modelo analítico de la primera fase se asumió que este se comportaba como un manipulador serial de tres eslabones. El análisis se realiza en una etapa de aterrizaje en donde se considera la primera junta como el contacto de la superficie del pie con la tierra y en donde no se tiene un torque para control.

Para escribir ecuaciones de movimiento se define el Lagrangiano L como la diferencia entre la energía cinética y potencial.

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q). \quad (32)$$

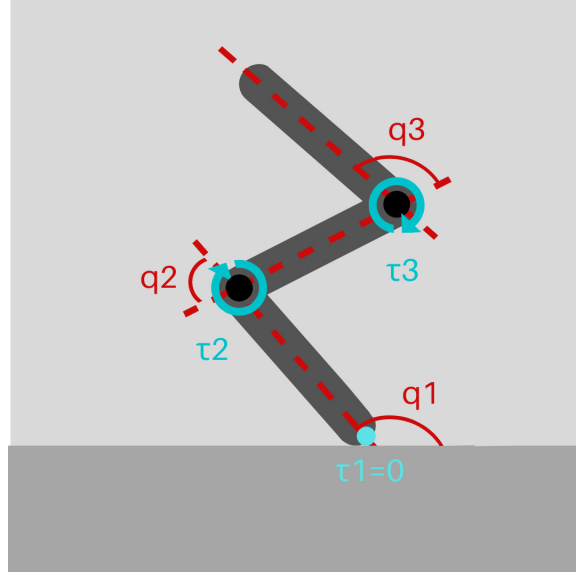


Figura 46: Modelo gráfico de la derivación analítica.

Dónde T corresponde a la energía cinética del sistema y V corresponde a la energía potencial. Esta ecuación de Lagrange usualmente se utiliza de manera vectorial con forma:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \gamma. \quad (33)$$

En este se modela cada eslabón como una barra rectangular homogénea con masa m_i y un momento de inercia:

$$I_i = \begin{bmatrix} I_{xi} & 0 & 0 \\ 0 & I_{yi} & 0 \\ 0 & 0 & I_{zi} \end{bmatrix}. \quad (34)$$

Asumiendo que $v_i \in \mathbb{R}^3$ es la velocidad traslacional del centro de masa de cada eslabón y $w_i \in \mathbb{R}^3$ es la velocidad angular se tiene que la energía cinética del manipulador es:

$$T(\theta, \dot{\theta}) = \frac{1}{2} m_1 \|v_1\|^2 + \frac{1}{2} w_1^T I_1 w_1 + \frac{1}{2} m_2 \|v_2\|^2 + \frac{1}{2} w_2^T I_2 w_2 + \frac{1}{2} m_3 \|v_3\|^2 + \frac{1}{2} w_3^T I_3 w_3. \quad (35)$$

Debido a que se tiene restricción de movimiento que se reduce al plano xy , $\|v_i\|$ es la magnitud de la velocidad del centro de masa de xy y w_i es un vector en la dirección del eje z con $\|w_1\| = \dot{\theta}_1$, $\|w_2\| = \dot{\theta}_1 + \dot{\theta}_2$ y $\|w_3\| = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3$.

Si $p_i = (x_i, y_i, 0)$ denota la posición del i -ésimo centro de masa y r_1, r_2, r_3 la distancia de las juntas al centro de masa de cada eslabón se tiene el siguiente set de ecuaciones:

$$\begin{aligned}
\bar{x}_1 &= r_1 c_1, & \dot{\bar{x}}_1 &= -r_1 s_1 \dot{\theta}_1, \\
\bar{y}_1 &= r_1 s_1, & \dot{\bar{y}}_1 &= r_1 c_1 \dot{\theta}_1, \\
\bar{x}_2 &= l_1 c_1 + r_2 c_{12}, & \dot{\bar{x}}_2 &= -(l_1 s_1 + r_2 s_{12}) \dot{\theta}_1 - r_2 s_{12} \dot{\theta}_2, \\
\bar{y}_2 &= l_1 s_1 + r_2 s_{12}, & \dot{\bar{y}}_2 &= (l_1 c_1 + r_2 c_{12}) \dot{\theta}_1 + r_2 c_{12} \dot{\theta}_2, \\
\bar{x}_3 &= l_1 c_1 + l_2 c_{12} + r_3 c_{123}, & \dot{\bar{x}}_3 &= -(l_1 s_1 + l_2 s_{12} + r_3 s_{123}) \dot{\theta}_1 - (l_2 s_{12} + r_3 s_{123}) \dot{\theta}_2 - r_3 s_{123} \dot{\theta}_3, \\
\bar{y}_3 &= l_1 s_1 + l_2 s_{12} + r_3 s_{123}, & \dot{\bar{y}}_3 &= (l_1 c_1 + l_2 c_{12} + r_3 c_{123}) \dot{\theta}_1 + (l_2 c_{12} + r_3 c_{123}) \dot{\theta}_2 + r_3 c_{123} \dot{\theta}_3,
\end{aligned}$$

en donde $s_i = \sin \theta_i$, $s_{ij} = \sin(\theta_i + \theta_j)$ y $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$ similar al caso c_i , c_{ij} y c_{ijk} . Con esto se tiene que la ecuación de la energía cinética se convierte en:

$$\begin{aligned}
T(\theta, \dot{\theta}) &= \frac{1}{2} m_1 (\dot{\bar{x}}_1^2 + \dot{\bar{y}}_1^2) + \frac{1}{2} I_{z1} \dot{\theta}_1^2 + \frac{1}{2} m_2 (\dot{\bar{x}}_2^2 + \dot{\bar{y}}_2^2) + \frac{1}{2} I_{z2} (\dot{\theta}_1 + \dot{\theta}_2)^2 \\
&\quad + \frac{1}{2} m_3 (\dot{\bar{x}}_3^2 + \dot{\bar{y}}_3^2) + \frac{1}{2} I_{z3} (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2. \quad (36)
\end{aligned}$$

De forma matricial con una serie de simplificaciones se obtuvo:

$$\begin{aligned}
T(\theta, \dot{\theta}) &= \frac{1}{2} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}^T \\
&\quad \begin{bmatrix} \alpha + 2\beta c_2 + 2\eta c_3 + 2\epsilon c_{23} & \delta + \epsilon c_{23} + \beta c_2 + 2\eta c_3 & \gamma + \epsilon c_{23} + \eta c_3 \\ \delta + \epsilon c_{23} + \beta c_2 + 2\eta c_3 & \delta + 2\eta c_3 & \gamma + \eta c_3 \\ \gamma + \epsilon c_{23} + \eta c_3 & \gamma + \eta c_3 & \gamma \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad (37)
\end{aligned}$$

donde

$$\begin{aligned}
\alpha &= I_{z1} + I_{z2} + I_{z3} + m_1 r_1^2 + m_2 (l_1^2 + r_2^2) + m_3 (l_1^2 + l_2^2 + r_3^2), \\
\beta &= m_2 l_1 r_2 + m_3 l_1 l_2, \\
\delta &= I_{z2} + I_{z3} + m_2 r_2^2 + m_3 (l_2^2 + r_3^2), \\
\eta &= m_3 l_2 r_3, \\
\epsilon &= l_1 m_3 r_3, \\
\gamma &= I_{z3} + m_3 r_3^2.
\end{aligned}$$

Se debe tomar en cuenta la energía potencial para considerar los efectos de gravedad para la extremidad. La ecuación está dada por:

$$P(\mathbf{q}) = m_1 g r_1 s_1 + m_2 g (l_1 s_1 + r_2 s_{12}) + m_3 g (l_1 s_1 + l_2 s_{12} + r_3 s_{123}). \quad (38)$$

Finalmente se sustituye el Lagrangiano $L = T + P$ en las ecuaciones de Lagrange para

obtener:

$$\begin{bmatrix} \alpha + 2\beta c_2 + 2\eta c_3 + 2\epsilon c_{23} & \delta + \epsilon c_{23} + \beta c_2 + 2\eta c_3 & \gamma + \epsilon c_{23} + \eta c_3 \\ \delta + \epsilon c_{23} + \beta c_2 + 2\eta c_3 & \delta + 2\eta c_3 & \gamma + \eta c_3 \\ \gamma + \epsilon c_{23} + \eta c_3 & \gamma + \eta c_3 & \gamma \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} -\epsilon s_{23}(\dot{\theta}_2 + \dot{\theta}_3) & -\beta s_2(2\dot{\theta}_1 + \dot{\theta}_2) - \epsilon s_{23}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) & -\eta s_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3) - \epsilon s_{23}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ (\beta s_2 + \epsilon s_{23})\dot{\theta}_1 - \eta s_3\dot{\theta}_3 & -\eta s_3\dot{\theta}_3 & -\eta s_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ \eta s_3(\dot{\theta}_1 + 2\dot{\theta}_2) + \epsilon s_{23}\dot{\theta}_1 & \eta s_3\dot{\theta}_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + g \begin{bmatrix} m_3(l_2 c_{12} + l_1 c_1 + r_3 c_{123}) + m_2(r_2 c_{12} + l_1 c_1) + m_1 r_1 c_1 \\ m_3(l_2 c_{12} + r_3 c_{123}) + m_2 r_2 c_{12} \\ m_3 r_3 c_{123} \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_2 \\ \tau_3 \end{bmatrix}. \quad (39)$$

10.1.2. Parámetros y matriz de Denavit-Hartenberg

Para realizar las simulaciones de la extremidad se utilizaron datos reales presentados en la extremidad. Siguiendo como referencia la Figura 46 para obtener las masas se utilizó una pesa de cocina. Para obtener los centros de masa se utilizó el software de inventor, en donde se aplicó el material de cada pieza y se creó el material PLA para definir una mejor densidad. Las mediciones realizadas se obtuvieron con respecto a la junta más cercana al suelo. Estas se observan en el Cuadro 5.

| | |
|---------|----------|
| m_1 | 8g |
| m_2 | 40g |
| m_3 | 33g |
| CoM_1 | 28.71mm |
| CoM_2 | 33.781mm |
| CoM_3 | 55.840mm |

Cuadro 5: Masas y centros de masa de los eslabones.

Luego se obtuvo las inercias de la misma manera y para validar los parámetros que proporcionaba inventor se asumió que los eslabones se modelaban como cilindros y con esto se obtuvo distintos resultados de inercias.

| | Inercias Inventor ($kg\ mm^2$) | Inercias cilíndricas ($kg\ mm^2$) |
|----------|----------------------------------|-------------------------------------|
| I_{z1} | 3.548 | 2.47 |
| I_{z2} | 26.025 | 23.11 |
| I_{z3} | 10.357 | 13.96 |

Cuadro 6: Inercias de cada eslabón.

Por último, se obtuvo la matriz de Denavit-Hartenberg.

| θ_j | d_j | a_j | α_j |
|------------|-------|-------|------------|
| q_1 | 0 | 49.70 | 0 |
| q_2 | 0 | 66.91 | 0 |
| q_3 | 0 | 54.15 | 0 |

Cuadro 7: Matriz D-H.

10.1.3. Simulación con single-shooting

A partir del modelo matemático del sistema y con los parámetros obtenidos se realizó una simulación por medio de un problema de optimización con single-shooting. Este programa tuvo el planteamiento del problema de optimización adaptando el modelo de referencia del SLIP, mediante optimización se declararon una serie de variables de decisión en el cual se puede variar: el tiempo de muestreo, la longitud inicial de la extremidad, las posiciones iniciales del sistema, las posiciones iniciales de los eslabones y el controlador PID.

Debido a que en pruebas iniciales el sistema se mostraba con reacciones caóticas se decidió no implementar el argumento de la gravedad sobre el sistema ya que, en la plataforma de pruebas real, el movimiento se ve restringido.

Durante las pruebas se realizó una gran variación de los parámetros sin obtener resultados satisfactorios ya que el sistema no se encuentra totalmente restringido, esto es que el tiempo de muestreo y por lo tanto el tiempo pueden presentar comportamientos negativos, la longitud del modelo puede variar presentando un tamaño no real, no se restringen los ángulos de movimiento de cada eslabón, etc. Sin embargo, en trayectorias cortas dentro de las simulaciones realizadas se observó ciertos comportamientos deseados similares al modelo SLIP por lo tanto, se observa que encontrar el mínimo local dentro de la función es sumamente difícil sin embargo, este existe.

10.1.4. Control dinámico para sistemas de base fija

Para la realización de este modelo se implementó un modelo de base fija visto durante el curso de Robótica 2. Para este se implementó el sistema sin restricciones de robots de base fija que tomó la siguiente forma:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u} \quad (40)$$

Implementando un sistema de control por medio de optimización se adaptó la trayectoria del modelo SLIP mediante una curva por polinomios en el cual se forzó a que se cumpliera lo siguiente:

$$\ddot{\mathbf{y}} = \mathbf{w}_y = -\mathbf{K}_P(\mathbf{y} - \mathbf{y}_d) - \mathbf{K}_D(\dot{\mathbf{y}} - \dot{\mathbf{y}}_d) + \ddot{\mathbf{y}}_d \quad (41)$$

Una vez realizando la simulación se estableció que el modelo representó adecuadamente el movimiento de la trayectoria del modelo SLIP durante la fase de contacto con el piso. Sin embargo, este presentó fuerzas en la primera junta la cual no tiene un actuador por la derivación del péndulo triple.

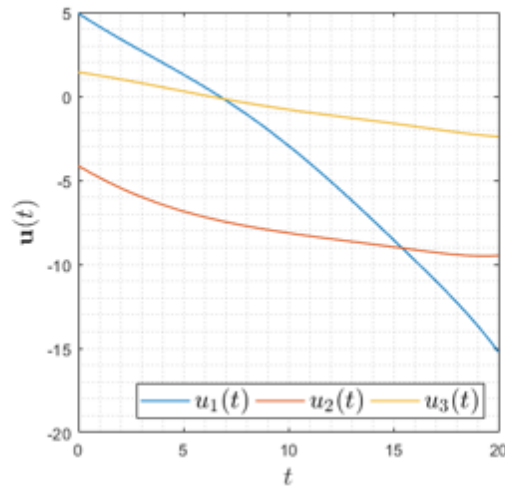


Figura 47: Esfuerzos de control realizados por el modelo.

Con esto se estableció que el modelo de la fase 1 es extremadamente ineficiente para realizar los saltos debido a los altos esfuerzos requeridos en la etapa de control.

10.1.5. Pruebas realizadas

Para obtener el comportamiento deseado en la extremidad se utilizaron dos motores. El primero tiene una relación de 75 : 1 y el segundo una relación de 100 : 1 ambos con 12 pulsos por revolución. El voltaje utilizado para estos motores fue de 12V sin limitaciones de corriente.

Debido a lo discutido en la sección 10.3 únicamente se logró que la extremidad saltara en un mismo lugar utilizando un ángulo inicial de 360° y un ángulo final de 400° para el motor ubicado en la cadera (motor 1). Para el segundo motor se utilizó un ángulo inicial de 360° y un ángulo final de 280° . El resultado se observa en la Figura 48.

Sin embargo, a pesar de no tener las capacidades para entregar la corriente máxima requerida por la extremidad se logró obtener un pequeño desplazamiento horizontal sin despegar la extremidad totalmente del suelo.

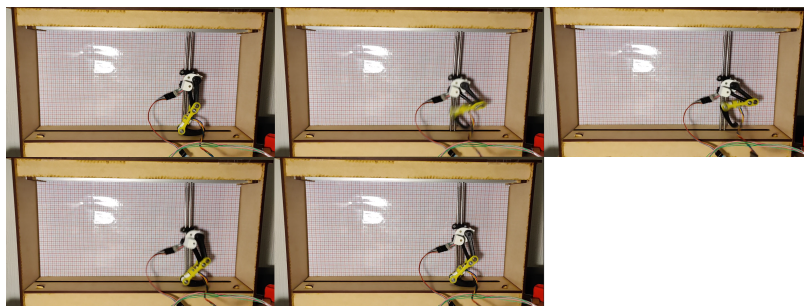


Figura 48: Salto en un único lugar.

10.2. Fase 2

Debido a la simplicidad que se presenta en la extremidad de la fase 2 y la similitud que esta tiene con el modelo SLIP presentado previamente únicamente se utilizaron las simulaciones en Matlab de dicho modelo.

10.2.1. Modelo slip

Para la simulación del modelo se utilizó el empleado en el curso de Robótica 2 en el cual se presenta el modelo del SLIP en donde se pueden realizar pruebas cambiando ciertos parámetros.

Los parámetros del sistema corresponden a la gravedad, masa, longitud, torque de entrada y una constante de compresión del resorte. Además, se presentan distintas condiciones iniciales para modificar el ángulo de la cadera. Este será importante para demostrar el funcionamiento de la extremidad con las pruebas realizadas.

Para el primer set de pruebas se inició con el modelo saltando en un solo lugar. En este se limitó el ángulo deseado a 0° . El resultado de se puede observar en la Figura 49. Dentro de esta se observa que el eje x se mantiene en un solo lugar y se logran varios saltos si el ángulo de inicio y deseado de la extremidad se mantiene en 0. Como segundo método de pruebas se inició la extremidad con un ángulo deseado de $\pi/8$. Debido a que ahora el ángulo es distinto de cero este muestra un desplazamiento horizontal. Como se observa en la Figura 50.

Ambas pruebas se realizaron cumpliendo los requerimientos del modelo en donde no se utiliza ningún torque de entrada para realizar el salto y en donde se da el caso donde la energía se conserva perfectamente.

10.2.2. Comparación entre modelo y pruebas realizadas.

Para las pruebas físicas de la extremidad se inició con un ángulo deseado de 0° . La finalidad de este era comprobar que la pierna era capaz de saltar en un mismo lugar similar al resultado mostrado en la Figura 49. Para realizar las pruebas se inicializó el motor con una caja reductora de 75 : 1 en 8V, presentando 12 pulsos por revolución y toda la corriente

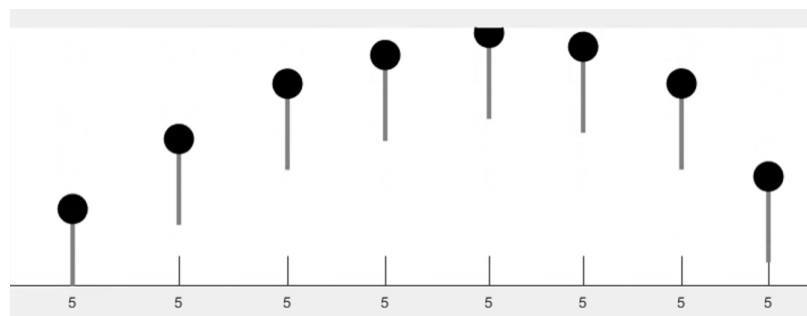


Figura 49: Salto del modelo en un único lugar.

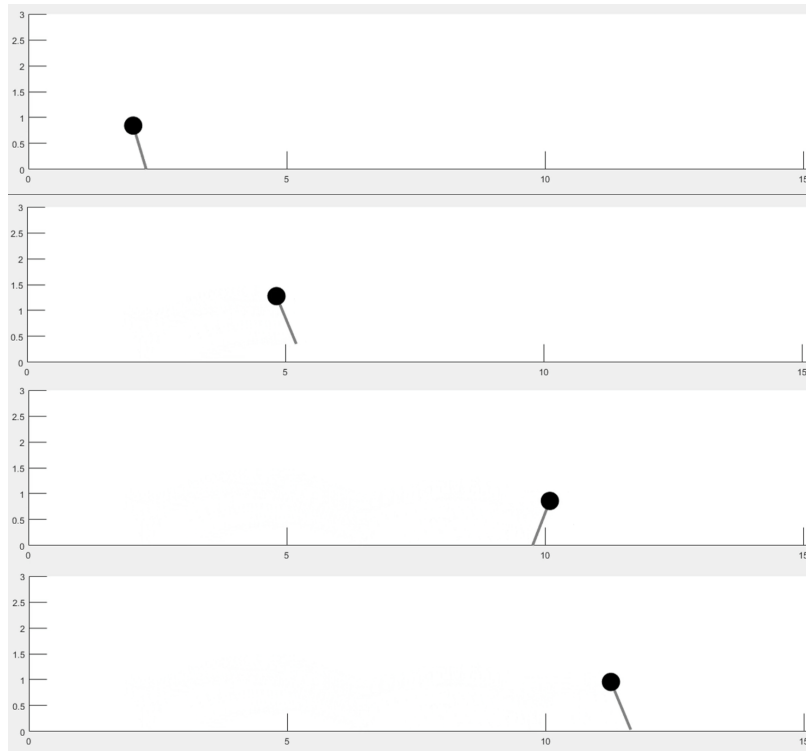


Figura 50: Salto del modelo con ángulo de $\pi/8$.

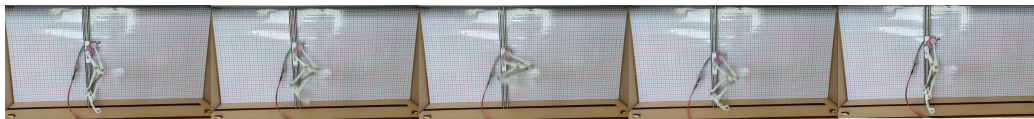


Figura 51: Salto de la extremidad en un mismo lugar.

que la fuente de voltaje pudiera proporcionar. Los resultados obtenidos se muestran en la Figura 51. En esta se puede apreciar que el modelo físico cumple con las simulaciones del modelo slip.

Para demostrar que la extremidad era capaz de realizar varios saltos desplazándose horizontalmente se modificó el ángulo inicial de la extremidad (el equivalente de modificar el ángulo deseado dentro de la simulación). El resultado se muestra en la Figura 52. Con los siguientes resultados se valida el modelo físico. La extremidad se logra comportar de la siguiente manera ya que a diferencia del modelo SLIP, se tiene la combinación de un torque de entrada junto a un resorte como elementos de recuperación de energía.

Para realizar el salto se utilizó el mismo controlador PID en ambos casos en donde el ángulo inicial de la extremidad era de 360° , el ángulo final era de 430° y las variables del controlador fueron las siguientes:

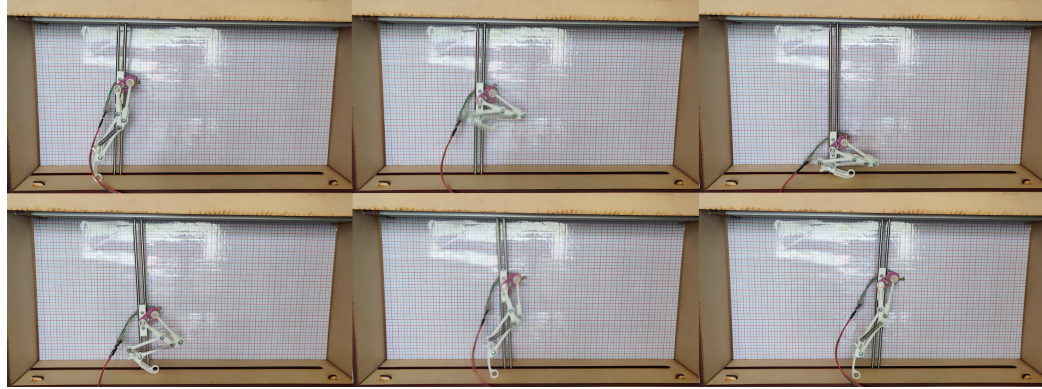


Figura 52: Salto de la extremidad con un desfase angular.

| | Kp | Kd | Ki |
|------------|-----------|-----------|-----------|
| Despegue | 10 | 2.7235 | 0.02599 |
| Aterrizaje | 5.5 | 2.7235 | 0.02599 |

Cuadro 8: Constantes PID para realizar un salto.

10.3. Limitaciones encontradas

Debido a las limitaciones que existieron para la pandemia se adquirió una fuente de voltaje con una única salida. Esta produce una variación de voltaje entre 0 y 30V con un máximo de corriente de 10A. Sin embargo, una desventaja de utilizar una fuente de voltaje distinta a las encontradas dentro del campus es que esta no era capaz de entregar los picos de corriente que necesitaban las extremidades para saltar. Esto resultó ser un inconveniente durante las pruebas ya que para lograr un funcionamiento correcto durante la primera fase del proyecto se utilizaron dos fuentes de voltaje conectadas en paralelo para brindar la mayor capacidad de corriente y lograr producir un salto.

- Implementando una caracterización de la extremidad de la fase 2 por medio de RecurDyn se definió que este es un prototipo más eficiente y con mejor desempeño ya que es capaz de alcanzar aproximadamente dos veces el desplazamiento vertical (170 mm) en comparación a la primera fase (90 mm).
- En términos de la distancia horizontal ambas extremidades tienen la misma capacidad de desplazamiento de 100 mm, con la diferencia que se logró un ángulo de despegue modificable en la segunda fase.
- Para el salto dentro de un entorno de simulación la extremidad de la fase 2 requiere 800 Nm menos de torque utilizando un solo motor y por menos tiempo que la fase 1.
- Se logró mejorar la respuesta dinámica de las extremidades mediante el rediseño de ciertas piezas de ambas fases y un ajuste de tolerancias para las mismas. En la primera fase, se logró un ahorro de material de impresión del 94.71 % disminuyendo el volumen de las piezas de sacrificio para la transmisión de potencia.
- Para la segunda fase, se definió un resorte con una constante de 47.06 N/m ya que previamente no se tenía una métrica para comparar el desempeño de la extremidad.
- Para la extremidad de la fase 2, se implementó un controlador PID debido a que las simulaciones del modelo SLIP demuestran que este controlador satisface el comportamiento deseado. El resorte y el motor se utilizan para cumplir el supuesto de conservación de energía dentro del modelo matemático.
- Similarmente, para la extremidad empleada en la primera fase se implementó un controlador PID mediante optimización de trayectorias demostrando que se logra un comportamiento deseado utilizando dicho esquema de control.
- Se estableció una nueva plataforma de programación unificada para controlar cualquiera de las dos extremidades existentes. Esta tiene la capacidad de cambiar hasta dos referencias de control periódicamente, empleando dos tipos de operación. Existe la operación en modo continuo donde la extremidad sigue saltando de forma indefinida

y el modo por comunicación con el usuario, en donde se espera una respuesta para realizar un ciclo continuo.

CAPÍTULO 12

Recomendaciones

- Implementar nuevos métodos de sensado para definir tanto la posición angular de los motores como información del entorno en la extremidad.
- Implementar castigadores maquinados en aluminio, hierro o bronce para evitar el juego no deseado en ambas extremidades.
- Para la fase 1, implementar engranajes cónicos maquinados en una máquina CNC con material de acero o bronce.
- Emplear un nuevo mecanismo de almacenamiento de energía para no forzar a los motores por encima de su voltaje nominal.
- Como protección, añadir circuitos que restrinjan la corriente hacia el microcontrolador y sobrecorriente en los motores.
- Definir los picos de corriente máximos empleados por los motores en cada extremidad.

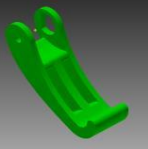
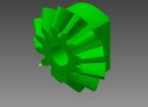
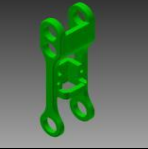

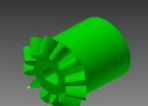
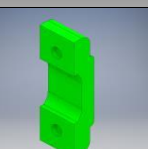
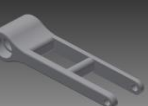

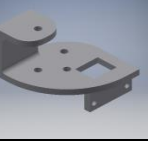
-
- [1] S. Cotton, I. Oлару, M. Bellman, T. van der Ven, J. Godowski y J. Pratt, “FastRunner: A fast, efficient and robust bipedal robot. Concept and planar simulation”, *Proceedings - IEEE International Conference on Robotics and Automation*, págs. 2358-2364, mayo de 2012. DOI: 10.1109/ICRA.2012.6225250.
 - [2] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer y M. Hoepffinger, “ANYmal - a highly mobile and dynamic quadrupedal robot”, en *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, págs. 38-44.
 - [3] C. Chiroy, “Desarrollo de una extremidad bio-inspirada en las extremidades posteriores de lepóridos con manejo eficiente de energía”, Fase 1, Tesis de mtría., Universidad del Valle de Guatemala, 2019.
 - [4] E. Bautista, “Diseño e implementación de una pata robótica de un grado de libertad bio-inspirada en las extremidades posteriores de lepóridos”, Fase 2, Tesis de mtría., Universidad del Valle de Guatemala, 2019.
 - [5] M. R. Dawson, “Later Tertiary Leporidae of North America”, 1958.
 - [6] A. T. Smith, *Lagomorph*, feb. de 2020. dirección: <https://www.britannica.com/animal/lagomorph>.
 - [7] S. Williams, A. Wilson y R. Payne, “Functional specialisation of the thoracic limb of the hare (*Lepus europeus*)”, *Journal of anatomy*, vol. 210, n.º 4, págs. 491-505, 2007.
 - [8] J.-A. Meyer y A. Guillot, “Biologically Inspired Robots”, en. ene. de 2008, págs. 1395-1422. DOI: 10.1007/978-3-540-30301-5_61.
 - [9] N. S. Nise, *CONTROL SYSTEMS ENGINEERING, (With CD)*. John Wiley & Sons, 2007.
 - [10] K. Ogata, *Modern Control Engineering*, 5th. USA: Prentice Hall PTR, 2010, ISBN: 0130609072.
 - [11] M. S. Triantafyllou, “Maneuvering and control of surface and underwater vehicles”, 2004.

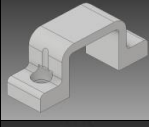







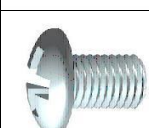

- [12] R. Murray, *Lecture 2 -LQR Control*, ene. de 2006.
- [13] M. Kelly, “An introduction to trajectory optimization: How to do your own direct collocation”, *SIAM Review*, vol. 59, n.º 4, págs. 849-904, 2017.
- [14] M. Cannon, “C21 Model Predictive Control”, *Lecture Notes, Oxford University, Oxford*, 2016.
- [15] M. Kelly, “An introduction to trajectory optimization: How to do your own direct collocation”, *SIAM Review*, vol. 59, n.º 4, págs. 849-904, 2017.
- [16] M. Kelly, *Cannon Example*, <http://www.matthwepeterkelly.com/tutorials/trajectoryOptimization/canon.html>, Accesado: Septiembre 19, 2020.
- [17] M. Spong, S. Hutchinson y M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005, ISBN: 9780471649908. dirección: <https://books.google.com.gt/books?id=wGapQAAACAAJ>.
- [18] *RecurDyn*. dirección: <https://www.enginsoft.com/solutions/recurdyn.html>.
- [19] I. Poulakakis, “Stabilizing Momopedal Robot Running: Reduction-by-Feedback and Compliant Hybrid Zero Dynamics.”, ene. de 2008.
- [20] R. Tedrake, *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying and Manipulation(Course Notes for MIT 6.832)*, <http://underactuated.mit.edu/>,Accesado Agosto 12, 2020.
- [21] K. Ogata, *System dynamics*. Pearson/Prentice Hall Upper Saddle River, NJ, 2004, vol. 13.
- [22] *CMSIS-DSP*, abr. de 2020. dirección: https://www.keil.com/pack/doc/CMSIS/DSP/html/group__PID.html.
- [23] *Pololu Universal Aluminum Mounting Hub for 3mm Shaft, M3 Holes (2-Pack)*, <https://www.pololu.com/product/1996>, Accesado: Septiembre 17, 2020.
- [24] MiSUMi, *Miter Gear (Kyouiku Gear) (M50B20*1103)*, <https://us.misumi-ec.com/>, Accesado: Septiembre 17, 2020.

14.1. Programación del microcontrolador

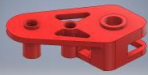


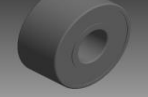


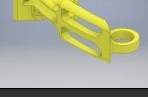
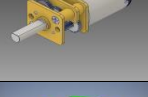
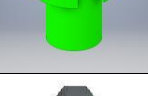
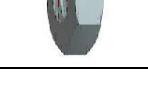
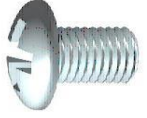



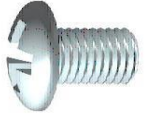

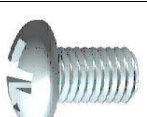
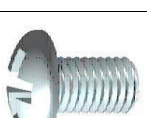
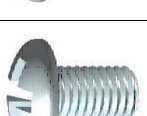
La programación se encuentra en un repositorio en el siguiente enlace:<https://github.com/san16250/Control-PID>.

14.2. Listado de piezas fase 1

| Item | Part Number | Thumbnail | QTY |
|------|---------------------------------|-------------------------------------------------------------------------------------|-----|
| 1 | PIE_MOD |  | 1 |
| 2 | Engrane Pie |  | 1 |
| 3 | TM2 |  | 1 |
| 4 | micro-metal-gearmotor-backshaft |  | 2 |
| 5 | Bevel Gear11_CPY |  | 1 |
| 6 | BRACKET_MODIFICADO |  | 1 |
| 7 | UNION |  | 1 |
| 8 | Fmr |  | 1 |
| 9 | CADERA_MODIFICADA |  | 1 |

| | | | |
|----|--------------------------------|---------------------------------------------------------------------------------------|----|
| 10 | BRACKET_CADERA |  | 1 |
| 11 | 623ZZ |  | 12 |
| 12 | SOPORTE | *Varies* | 2 |
| 13 | EJE |  | 2 |
| 14 | ISO 7045 - M3 x 16 - 4.8 - H |  | 4 |
| 15 | DIN 439 - M3 |  | 9 |
| 16 | ISO 7045 - M2 x 10 - 4.8 - H |  | 2 |
| 17 | DIN 439 - M2 |  | 2 |
| 18 | ISO 7045 - M3 x 30 - 4.8 - H |  | 5 |
| 19 | ISO 7045 - M2.5 x 16 - 4.8 - H |  | 2 |
| 20 | DIN 439 - M2.5 |  | 2 |

14.3. Listado de piezas fase 2

| Item | Part Number | Thumbnail | QTY |
|------|------------------------------|---------------------------------------------------------------------------------------|-----|
| 1 | base de diseño version 2 |  | 1 |
| 2 | femur version 3 |  | 1 |
| 3 | prefemur modificado |  | 1 |
| 4 | 693ZZ |  | 7 |
| 5 | eslabon extra |  | 1 |
| 6 | tibia |  | 1 |
| 7 | pie version 4 |  | 1 |
| 8 | micro-metal-gearmotor-bac |  | 1 |
| 9 | pin_motor |  | 1 |
| 10 | DIN 439 - M3 |  | 11 |
| 11 | ISO 7045 - M3 x 20 - 4.8 - H |  | 2 |
| 12 | frontalV3 |  | 1 |
| 13 | traseraV2 |  | 1 |
| 14 | EJE |  | 2 |
| 15 | ISO 7045 - M3 x 14 - 4.8 - H |  | 1 |
| 16 | ISO 7045 - M3 x 12 - 4.8 - H |  | 2 |
| 17 | ISO 7045 - M3 x 25 - 4.8 - H |  | 1 |
| 18 | ISO 7045 - M3 x 30 - 4.8 - H |  | 1 |
| 19 | ISO 7045 - M3 x 16 - 4.8 - H |  | 2 |

ABS: Referido a un material de impresión 3D, el acrilonitrilo butadieno estireno es un material resistente a la temperatura. 48, 74

ADC: Por sus siglas en inglés, este se refiere a un conversor analógico a digital. 17, 74

animatronics: Es un campo multidisciplinario en donde se integra la anatomía y la mecatrónica para crear imitaciones de animales y humanos para que cumplan una tarea específica. 14, 74

Boston Dynamics: Empresa líder especializada en robots móviles. 74

CAD: Dado por sus siglas en inglés este es un proceso para diseñar piezas asistido mediante computadora. 1, 74

CMSIS: Es una interfaz de software estándar para microcontroladores tipo Cortex. 37, 74

CoLink: Es una extensión de RecurDyn en donde el entorno de programación se da por medio de diagramas de bloques para lograr realizar una tarea. 24, 30, 74

DAC: Por sus siglas en inglés, este se refiere a un conversor digital a analógico. 17, 74

DC: Término referido al modo de corriente directa en un motor, fuente o circuito. 5, 74

Deadbeat: Es un elemento de la teoría de control en términos de tiempo discreto capaz de asegurar una solución en tiempo finito. 26, 74

debug: Indica el proceso de encontrar y resolver problemas que se dan durante la tarea de programación. 38, 74

encoder: Es un sensor capaz de convertir una posición a una señal electrónica. 38–40, 74

Fixed: Es un término utilizado dentro de los programas para referirse al punto inmóvil del marco de referencia inercial. 23, 29, 74

GitHub: Sitio web que provee un servicio para desarrollar software utilizando un sistema de control de versiones. 42, 74

I2C: Es un método de comunicación serial síncrona utilizado en circuitos integrados de bajas velocidades. 42, 74

Inventor: Programa de diseño CAD utilizado para fabricar las extremidades.. 23, 29, 74

LIDAR: Sensor infrarrojo utilizado para tomar mediciones de distancia. 4, 74

LQR: Regulador lineal cuadrático. Es un método de control óptimo que funciona mediante lazo abierto.. 18, 26, 74

MakerLab: Laboratorio del departamento de ingeniería mecatrónica y electrónica de la Universidad del Valle de Guatemala. 47, 74

Matlab: Este es una plataforma de programación diseñada específicamente para ingenieros y científicos. Este permite expresar la matemática de forma natural debido a que es un lenguaje basado en matrices. 24, 32, 61, 74

PETG: Por sus siglas en inglés, se refiere al tereftalato de polietileno. Este es un material rígido y con alta resistencia térmica utilizado para impresión 3D. 48, 74

PID: Es un sistema de control clásico basado en tres variables que cumplen un distinto propósito para definir un error de algún valor o trayectoria deseada con respecto a la actual. 4, 5, 11, 16, 18, 35, 37–41, 59, 62, 74

PLA: Por sus siglas en inglés, se refiere al ácido poliláctico. Este es un polímero utilizado para impresión 3D. 45–48, 58, 74

PWM: La modulación por ancho de pulsos es una señal periódica cuadrada a la que se le puede modificar el ciclo de trabajo para el control de velocidad de los motores. 40, 74

RecurDyn: Programa utilizado para realizar simulaciones dinámicas utilizando restricciones de contacto. 1, 22–24, 29, 32, 33, 74

Simulink: Extensión de Matlab en un ambiente de diagrama de bloques para simulaciones y diseño de modelos. 1, 24, 31–33, 74

single-shooting: Al igual que “Multiple-Shooting” es un método de discretización dentro de la optimización de trayectorias. 19, 59, 74

SLIP: Por sus siglas en inglés de “Spring-loaded inverted pendulum”. Es un modelo usado como base dentro de la robótica bio-inspirada para implementar sistemas más complejos. 26, 59, 61, 62, 74

software: Conjunto de componentes diseñado para la solución de tareas específicas en un entorno de computación. 1, 22–24, 29, 48, 50, 58, 74

solver: Este es una rama del programa capaz de recopilar descripciones de un problema de forma genérica y calcular la solución. 19, 22, 74

STEP: Es un tipo de extensión de computadora utilizado para manufacturar piezas en 3D. 23, 29, 74

timer: se refiere a un temporizador utilizado dentro de la programación para medir tiempos específicos. 38, 39, 74

UART: Es un módulo para comunicación serial asíncrona entre dos dispositivos. 38, 39, 74