

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de un robot bioinspirado con
mecanismo de salto y control de orientación por medio de
palanca de inercia**

Trabajo de graduación presentado por Jerry Daniel Rivera Molina para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2020

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de un robot bioinspirado con
mecanismo de salto y control de orientación por medio de
palanca de inercia**

Trabajo de graduación presentado por Jerry Daniel Rivera Molina para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,


2020

Vo.Bo.:

(f) 
MSc. Miguel Zea

Tribunal Examinador:

(f) 
MSc. Miguel Zea

(f) 
Ing. Pablo Mazariegos

(f) 
Ing. Kurt Kellner

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

Agradezco a la Universidad del Valle de Guatemala el haberme abierto sus puertas y haberme guiado por el sendero del conocimiento que me ha permitido crecer como persona, estudiante y profesional.

Agradezco a todos los docentes que, a través de su enseñanza a lo largo de estos cinco años de estudios, me han transmitido los conocimientos necesarios para encontrarme hoy en donde estoy, en especial a Miguel Zea, quien me ha acompañado en parte fundamental de este trayecto.

Agradezco a la Fundación Juan Bautista Gutiérrez, y en especial a Isabel Gutiérrez de Bosch a quien recordaré por siempre con mucho afecto, el haber visto en mí a un joven a quien valía la pena apoyar para llevar a cabo una de sus metas.

Agradezco a Evelyn Barrascouth, Evelyn Hernández, Karín Carranza y Gloria Escobar el haberme acompañado a lo largo de esta etapa de mi vida y por haberme brindado su apoyo y aliento para ser una persona integral.

Agradezco a mis padres, Iván Rivera y Sara Molina, el apoyo incondicional que me han brindado desde siempre, que son los motores que me impulsan a salir adelante y quienes se han esforzado y sacrificado para que yo me encuentre en donde hoy me encuentro.

Agradezco a mi hermana, Kimberly Rivera, por el apoyo y buen ejemplo que me ha dado. Agradezco a mi tía, Helga Rivera, quien siempre ha estado para mí cuando la necesito. Agradezco a todos mis amigos por el apoyo, aliento y confianza que me han brindado, y por todo lo que he aprendido de ellos.

Agradezco finalmente a Dios, gracias a quien todo esto es posible y quien me ha dado la dicha de conocer a estas personas y de cumplir una más de mis metas.

Prefacio	v
Lista de figuras	x
Lista de cuadros	xi
Resumen	xiii
Abstract	xv
1. Introducción	1
2. Antecedentes	3
3. Justificación	5
4. Objetivos	7
5. Alcance	9
6. Marco teórico	11
6.1. Sistemas dinámicos y simulaciones	11
6.2. Control de sistemas dinámicos	13
6.2.1. Control clásico	13
6.2.2. Control moderno	14
6.3. El método de Euler-Lagrange para modelado	17
6.4. Otros métodos de control	18
6.5. Conservación de la energía y la cantidad de movimiento lineal	19
6.6. Inventor y Webots	20
6.7. Nodos principales en Webots	21
6.7.1. Nodo WorldInfo	21
6.7.2. Nodo Robot	21
6.7.3. Nodo Solid	22
6.7.4. Nodo Shape	22

6.7.5. Nudo Joint	22
7. Metodología	23
7.1. Diseño mecánico	23
7.2. Migración del diseño al entorno de simulación	24
7.3. Obtención de datos y mediciones del sistema	24
7.4. Modelado del sistema	24
7.5. Desarrollo e implementación del control	25
7.6. Realización de pruebas	25
8. Diseño mecánico	27
8.1. Definición de parámetros del motor.	27
8.2. Definición de parámetros del mecanismo.	27
8.3. Análisis del resorte.	29
8.4. Selección de resorte.	30
8.5. Bio-inspiración.	30
8.6. Componentes físicos.	31
8.7. Resultados.	33
8.8. Diseño CAD	38
8.8.1. Diseño final	45
9. Modelado del sistema	47
9.1. Modelo inicial	47
9.2. De MATLAB a Webots	54
9.3. Modelo de Autodesk Inventor en Webots	56
9.3.1. Configuración del entorno	56
9.3.2. Definición del robot	58
9.3.3. Obtención de datos y mediciones con MATLAB	64
10. Control del sistema	71
10.1. Buen control y mal aterrizaje	71
10.1.1. Control proporcional	71
10.1.2. Control proporcional derivativo	72
10.2. Buen control y buen aterrizaje	74
11. Conclusiones	79
12. Recomendaciones	81
13. Bibliografía	83
14. Anexos	85
14.1. Códigos	85
14.2. Planos de construcción	85
15. Glosario	95

Lista de figuras

1.	Proceso de diseño para selección de los motores y resortes.	23
2.	Metodología del proyecto.	25
3.	Mecanismo de salto del sistema.	28
4.	DCL de los eslabones del mecanismo.	30
5.	Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.104.130.0425.	34
6.	Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.110.125.0525.	34
7.	Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.118.125.0225.	35
8.	Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.128.125.0325.	35
9.	Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.128.125.0525.	36
10.	Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte D.121.160.0325.	36
11.	Disposición de los motores.	38
12.	Rango de movimiento de la barra de inercia.	39
13.	Base superior.	40
14.	Elementos para disposición de los motores.	40
15.	Base inferior.	41
16.	Patas del robot.	41
17.	Eslabón superior.	42
18.	Eslabón inferior.	42
19.	Elemento de sujeción para ensamblaje de la barra de inercia.	43
20.	Barra de inercia con elemento sujetador.	43
21.	Diseño del robot.	44
22.	Análisis estático de esfuerzos.	44
23.	Diseño final del robot vista 1.	45
24.	Diseño final del robot vista 2.	45
25.	Modelo inicial.	48

26.	Simulación del sistema dinámico.	51
27.	Simulación del sistema dinámico.	52
28.	Simulación del sistema dinámico aplicando la fuerza que experimentará el sistema real.	54
29.	Comprobación de resultados de MATLAB en Webots.	55
30.	Proceso de migración del diseño de Inventor a Blender y a Webots.	56
31.	Definición del entorno en Webots.	57
32.	Forma física del robot en Webots.	58
33.	Mediciones del ángulo de roll en Webots.	59
34.	Definición de la interacción del robot con el entorno en Webots.	60
35.	Junta deslizante en Webots.	61
36.	Eje de junta deslizante en Webots.	61
37.	Definición de la simulación de compresión del Robot.	62
38.	Análisis para aplicación de la fuerza en Webots.	63
39.	Definición de la barra de inercia.	63
40.	Respuesta de la velocidad al escalón de 0.3 Nm.	64
41.	Respuesta de la velocidad al escalón de 0.4 Nm.	65
42.	Respuesta de la velocidad al escalón de 0.5 Nm.	65
43.	Respuesta del sistema de posición al escalón de 0.3 Nm.	66
45.	Respuesta del sistema de posición al escalón de 0.5 Nm.	66
44.	Respuesta del sistema de posición al escalón de 0.4 Nm.	67
46.	Respuesta al escalón del sistema 1 controlado.	67
47.	Respuesta al escalón del sistema 2 controlado con $C = kP$	68
48.	Respuesta al escalón del sistema 2 controlado con $C = kP + kDs$	68
49.	Respuesta al escalón del sistema 3 controlado con $C = kP$	69
50.	Respuesta al escalón del sistema 3 controlado con $C = kP + kDs$	69
51.	Sistema controlado con $kP = -0.863$	72
52.	Sistema controlado con $kP = 0.863$ y $kd = 0.0055$	73
53.	Trayectoria del robot con mal aterrizaje.	73
54.	Definición de parámetros de las patas con amortiguamiento.	74
55.	Trayectoria del robot con buen aterrizaje.	75
56.	Ángulo roll y entrada de control del robot con buen aterrizaje.	75
57.	Geometría para sujetar las patas al mecanismo.	76
58.	Geometría para simular la pata del robot.	76
59.	Propuesta para implementar las extremidades con amortiguamiento.	77
60.	Altura máxima alcanzada por el robot.	78

Lista de cuadros

1.	Componentes sugeridos.	32
2.	Características de los motores recomendados.	32
3.	Masa total (desglosada) del sistema.	32
4.	Parámetros importantes de los motores.	34
5.	Posibles resortes y cálculos de interés usando el motor RB-Pol-446.	37
6.	Posibles resortes y cálculos de interés usando el motor RB-Pol-447.	37
7.	Posibles resortes y cálculos de interés usando el motor RB-Pol-448.	37
8.	Posibles resortes y cálculos de interés usando el motor RB-Pol-452.	37

Este trabajo tiene como objetivo principal diseñar e implementar un robot bio inspirado con mecanismo de salto y con control de orientación por medio de una palanca de inercia. Este trabajo constituye una continuación a la investigación llevada a cabo en la Universidad del Valle de Guatemala acerca del escarabajo clic, la cual tuvo como objetivo el estudio de este insecto para el desarrollo de un mecanismo bio inspirado de auto volteo para su implementación en vehículos de exploración y rescate. Sin embargo, el enfoque del presente estudio centró su inspiración en animales como las langostas, para la implementación del salto, y animales como las lagartijas, para la implementación del control de orientación por medio de la palanca de inercia.

Para alcanzar los objetivos se utilizaron herramientas de software como Autodesk Inventor, para la parte del diseño mecánico, y MATLAB y Webots, para la parte de simulación e implementación del control. El diseño mecánico constó de dos partes: el mecanismo de almacenamiento de energía y la barra de inercia para controlar la orientación del robot. Para el almacenamiento de la energía se utilizaron resortes torsionales y un motor que permitiera comprimirlos. Para la barra de inercia se utilizó una barra cilíndrica acoplada al mecanismo y que contara con un peso de 10 g en uno de sus extremos. Para asegurar el correcto aterrizaje del robot se empleó un mecanismo de amortiguamiento que permite absorber parte de la energía del impacto al aterrizar. Luego de estudiar al sistema se halló que el mismo puede aproximarse como un sistema LTI, por lo que para controlarlo fue suficiente un controlador proporcional derivativo PD. Finalmente, dado que los resultados obtenidos fueron basados en una simulación, se recomienda la implementación física del robot para comprobar los resultados.

The main objective of this work is to design and implement a bio-inspired robot with a jump mechanism and with orientation control by means of an inertia lever. This project constitutes a continuation of the research carried out at the University of the Valley of Guatemala on the click beetle, which had as its objective the study of this insect for the development of a bio-inspired auto-turning mechanism for its implementation in exploration and rescue vehicles. However, the present study focused its inspiration on animals such as lobsters, for the implementation of the jump, and animals such as lizards, for the implementation of the orientation control by means of the inertia lever.

To achieve the objectives, software tools such as Autodesk Inventor, for the mechanical design part, and MATLAB and Webots, for the simulation and control implementation part, were used. The mechanical design consists of two parts: the energy storage mechanism and the inertia bar to control the orientation of the robot. For energy storage, torsional springs and a motor were used to compress them. For the inertia bar, a cylindrical bar coupled to the mechanism and with a weight of 10 g at one end, was used. To ensure the correct landing of the robot, a damping mechanism was used that allows absorbing part of the impact energy when landing. After studying the system, it was found that it can be approximated as an LTI system, so to control it a proportional-derivative PD controller was sufficient. Finally, since the results obtained were based on a simulation, the physical implementation of the robot is recommended to verify the results.

Esta investigación gira entorno a uno de los temas de mayor auge en la actualidad: la robótica bio inspirada, la cual busca aprender o inspirarse en la naturaleza para desarrollar mecanismos que realicen tareas específicas de forma eficiente. Este trabajo tiene como objetivo desarrollar la simulación de un robot con salto, inspirado en el salto de animales como las langostas y que controle su orientación mediante una barra de inercia, tal y como lo hacen animales como las lagartijas al utilizar sus colas. La metodología utilizada se resume en: primero, desarrollar el diseño mecánico para generar el salto y el control, segundo, estudiar el comportamiento natural del mecanismo al realizar el salto para obtener un modelo aproximado que describa su dinámica y, finalmente, aplicar control a dicho modelo y verificar que de esta forma sea posible estabilizar la orientación del robot.

La investigación se presenta dividida en varias secciones. La primera de ellas muestra los antecedentes, en la que se mencionan brevemente los trabajos similares realizados con anterioridad y que han significado el punto de partida y la inspiración para este trabajo. Posteriormente, se presenta la justificación, los objetivos específicos y el alcance que tiene esta investigación. En la sección siguiente, el marco teórico, se presenta la teoría o los conceptos fundamentales en los que se basa esta investigación. En el capítulo de metodología se detallan los pasos del proceso llevado a cabo para alcanzar los objetivos.

En el capítulo de diseño se detalla tanto proceso seguido para el desarrollo del diseño mecánico del mecanismo de salto, así como la selección de los dispositivos electrónicos que lo conforman y permiten controlar al robot. En el capítulo de modelado se describe el proceso para llevar a cabo la simulación, así como los experimentos llevados a cabo con el diseño mecánico para la obtención de un modelo aproximado que describa la dinámica del robot. Finalmente, en el capítulo de control se muestra la implementación de un controlador PD para estabilizar al sistema, y se muestran los resultados obtenidos de las simulaciones llevadas a cabo con el controlador aplicado al robot.

Por último, se encuentran las conclusiones y recomendaciones de esta investigación, y en los anexos se muestran los planos de construcción del mecanismo, así como un enlace para acceder a la programación desarrollada para las simulaciones y el control del robot.

Insectos como las langostas, pulgas y el escarabajo clic han sido foco de atención para diversas investigaciones cuyos objetivos se centran en replicar los mecanismos de escape o movilización de dichos insectos. Tal es el caso de [1], llevada a cabo en la Universidad del Valle de Guatemala. El objetivo de esta investigación fue replicar el mecanismo de salto del escarabajo clic para implementarlo en vehículos terrestres y darles así la capacidad a estos de realizar un auto volteo al presentarse la situación en que el vehículo vuelque debido a las irregularidades del terreno por el que circula.

Por otro lado, es mayor la cantidad de investigaciones que han enfocado sus objetivos en el mecanismo de salto empleado por las pulgas y langostas, como [2], [3], [4] y [5], que tanto en [4] y [5] se encuentra una tabla de comparación de varios de los robots saltarines ya desarrollados. En dichas tablas se comparan estos robots en función de su peso, tamaño, altura del salto, alcance horizontal, actuadores usados y mecanismo de almacenaje de energía. Estas tablas resultan ser de gran utilidad para así seleccionar el mecanismo que mejor se adapte a las necesidades de esta investigación, entre las cuales destaca que el alcance horizontal sea el mayor posible y el tamaño, al igual que su peso, lo más reducido posible.

Basado en estas tablas y en otras referencias como [6], se encontró que los dos mecanismos que mejor cumplen con los requerimientos anteriormente expuestos son los utilizados en [2] y en [4]. Respecto a [2] se tiene que su alcance horizontal es de los más altos y que es el mecanismo con menor peso y tamaño de todos, sin embargo, su pequeño tamaño es debido a que carece de actuadores eléctricos como motores y usa únicamente resortes (Shape Memory Alloy (SMA:)), por lo que en el caso de incluirlos su tamaño incrementaría significativamente y su estructura y funcionamiento cambiarían. Respecto a [4], se observa que también cuenta con uno de los alcances horizontales más grandes, así como un bajo peso y tamaño, además de contar con un solo motor como actuador del mecanismo (lo cual lo vuelve más económico), por lo que se considera a éste como la mejor opción para alcanzar los objetivos de dicha investigación. Adicionalmente se considera también que el mejor método de almacenaje de energía potencial para este mecanismo es por medio de resortes de torsión como los utilizados en [4] y en [6].

Por último, como se mencionó con anterioridad, se desea aplicar control al robot mientras este se encuentra en el aire para así garantizar un aterrizaje seguro sin necesidad de un mecanismo extra que deba reajustar su postura. En [6] se utiliza una palanca de inercia (a la cual llaman cola) para realizar dicho control, tal y como se desea llevar a cabo en esta investigación. Cabe resaltar que en esta investigación no se implementará el desplazamiento mediante ruedas sino simplemente por medio del salto controlado.

A lo largo de la historia el hombre ha realizado diversa cantidad de mecanismos que logran realizar tareas de forma más eficiente que el mismo ser humano. Con el tiempo y la necesidad de mejorar continuamente dichos mecanismos, se ha concluido que, cuando de eficiencia se trata, la naturaleza es la mejor. Es por ello que en los últimos años, con el desarrollo de la tecnología, varias han sido las investigaciones y los intentos de recrear los mecanismos de la naturaleza para la solución de problemas u optimización de una solución dada.

Un claro ejemplo de lo expuesto anteriormente es la robótica bio-inspirada, que busca replicar la forma en que se comportan algunos animales o insectos en sus ambientes, como por ejemplo, cuando intentan escapar de algún depredador. Los mecanismos bio inspirados representan los mayores avances de la tecnología en el área de la robótica ya que permiten una representación más fiel o natural de los seres vivos a los que se busca imitar.

Muchas universidades de primer mundo llevan a cabo investigaciones y proyectos para el desarrollo de dichos mecanismos con el fin de mostrar sus avances tecnológicos y abrir camino a su implementación en distintas áreas como en ingeniería o incluso medicina.

Con el fin de impulsar estos proyectos, y por ende los avances tecnológicos en Guatemala, se lleva a cabo esta investigación que busca implementar un mecanismo de salto bio inspirado, siendo conscientes que a su vez dicho mecanismo puede implementarse en diversas disciplinas como los robots de exploración, tal y como se menciona en [1] y como se llevó a cabo con el Sand Flea Robot de Boston Dynamics, el cual es capaz de realizar saltos de aproximadamente 7.6 metros con el fin de superar muros y realizar vigilancia sobre lo que sucede en los alrededores de dichos muros.

4.1 Objetivo general

Diseñar e implementar un robot bio-inspirado con mecanismo de salto y con control de orientación por medio de una palanca de inercia.

4.2 Objetivos específicos

- Implementar un mecanismo de salto por medio de un motor, juego de engranajes y resortes torsionales.
- Implementar un control de orientación para el correcto aterrizaje por medio de una palanca de inercia y una unidad de medición inercial (IMU).
- Validar el mecanismo de salto y de control por medio de software de simulación.

Se estableció que el objetivo de esta investigación es llevar a cabo el diseño y la implementación de un robot bio-inspirado con mecanismo de salto y con control de orientación por medio de una palanca de inercia. Sin embargo, es de suma importancia aclarar a lo que el término “implementación” se refiere. Inicialmente, éste hacía referencia al proceso de manufactura de todas las piezas desarrolladas en la parte de diseño y su ensamblaje en conjunto con los dispositivos electrónicos necesarios, como los motores, los drivers, el microcontrolador y la batería, para generar así físicamente al robot. Sin embargo, la situación en el país y el mundo derivada por la pandemia de Covid-19 del año 2020, forzó a realizar una “modificación” al término, ya que entre las limitantes que se generaron por el distanciamiento social se encontró la dificultad para manufacturar los diseños. Si bien se contaba con la posibilidad de generarlos, el tiempo para manufacturarlos y obtenerlos se incrementaba debido a la cuarentena y a las restricciones de horarios establecidos en el país, lo cual, en conjunto con la naturaleza iterativa de la fase de diseño, sumaban una considerable cantidad de tiempo que superaría por mucho el tiempo disponible para llevar a cabo la investigación. Por ello, se estableció que la implementación del robot hace referencia al desarrollo de un modelo CAD del mismo con el cual es posible simular el comportamiento que éste tendría en el mundo real, haciendo uso de un software de simulación, tal y como se estableció en uno de los objetivos de la investigación.

El modelo CAD se realizará lo más realista posible con el fin de obtener resultados fiables y apegados a la realidad, sin embargo, el uso de modelos CAD de los dispositivos que utiliza el robot se verá limitado a la facilidad con que éstos se puedan obtener de fuentes de internet, dada su complejidad; de lo contrario, se utilizarán solamente los aspectos más relevantes de estos, como el espacio que ocuparían y su peso. Esto se establece de esta forma con el fin de emplear el menor tiempo posible en la etapa de diseño y enfocar los esfuerzos en la etapa de control.

Por lo anterior mencionado, esta investigación representa un punto de partida al desarrollo de robots con salto controlado con el fin de, como se mencionó en la justificación de este trabajo, impulsar los proyectos tecnológicos en la universidad, tal y como se realiza en las universidades de primer mundo.

Adicionalmente, debido a que, una vez más, no se contará con el robot físico, la programación para su control será llevada a cabo en MATLAB y no en un lenguaje específico para programar un microcontrolador. Sin embargo, sí se realizará una suferencia de qué microcontrolador utilizar y el respectivo lenguaje de programación, basado en los microcontroladores y lenguajes utilizados en la experiencia del autor. Del mismo modo con los demás dispositivos se realizarán sugerencias en cuanto a qué drivers y qué batería utilizar para que, en la siguiente fase de esta investigación, se tenga un punto de partida para la manufactura del robot.

6.1. Sistemas dinámicos y simulaciones

De acuerdo a [7] un sistema dinámico consiste en un conjunto de magnitudes medibles que evolucionan con el tiempo. Estos sistemas y magnitudes son representaciones del comportamiento de algún fenómeno de la naturaleza o de cualquier fenómeno de interés en distintas disciplinas, como por ejemplo la ingeniería. Estos sistemas, como se muestra en [8], pueden clasificarse, de acuerdo a las características que posean, de la siguiente forma:

- Sistema dinámico en tiempo continuo, en el cual los valores de las variables son función del tiempo continuo.
- Sistema dinámico en tiempo discreto, lo cual se refiere a una cantidad finita de muestras de tiempo, debido a la incapacidad de los computadores (a partir de los cuales surge la noción de discretización) de almacenar un conjunto infinito de datos.

Los fenómenos descritos por los sistemas dinámicos son de interés para realizar predicciones o control sobre los mismos, por lo que es de suma importancia contar con un modelo que replique su comportamiento.

En [8] se expresa que la necesidad del modelado es debido a que por medio de éste, como se mencionó con anterioridad, es posible obtener una representación del sistema y, más importante aún, realizar pruebas sobre el mismo, ya sea de forma aislada o considerando las interacciones con su entorno, sin necesidad de recurrir a la manufactura, lo cual genera gastos y una considerable inversión de tiempo, sino por medio de la simulación, lo cual se detallará más adelante. Los modelos se obtienen a partir de análisis matemático que llevan a la deducción de ecuaciones dinámicas que muestran la forma en que se relacionan las magnitudes o variables de interés. Hay diversas formas en que estas ecuaciones pueden expresarse y manipularse, pero las más utilizadas son las siguientes

- Ecuaciones diferenciales
- Funciones de transferencia
- Variables de estado

Independientemente de la forma de expresarlas, las ecuaciones cuentan ya sea con solución abierta o cerrada, a excepción de los sistemas no lineales los cuales rara vez cuentan con solución cerrada, y dependiendo de esto la forma de hallar esta solución puede ser, según [8]

- Analítica, la cual consiste en expresar la solución por medio de una expresión matemática explícita que permite, al sustituir los valores de la variable independiente, obtener los valores de la función incógnita.
- Numérica, la cual consiste en la resolución numérica de ecuaciones diferenciales, usando para ello un método numérico incorporado en una computadora digital. Dependiendo de los recursos del usuario, puede emplearse desde un lenguaje de programación de propósito general, hasta un software especializado ad hoc, con las interfaces específicamente diseñadas para tal fin.

Por lo general, los sistemas dinámicos son representados por expresiones matemáticas de considerable complejidad por lo que, aunque cuente con solución analítica, se recurre a la solución numérica, delegando esta tarea a un dispositivo de cómputo. A pesar de que estos cálculos son llevados a cabo con mayor facilidad por estos dispositivos, existen formas de facilitar aún más esta tarea para los ordenadores de forma que sean más eficientes y que el costo computacional sea el óptimo. Es por ello que el método de variables de estado, que en su forma más general se expresa como

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (1)$$

presenta cierta ventaja, ya que expresar al sistema en forma matricial resulta conveniente debido a que el álgebra lineal es de las formas más óptimas para los ordenadores al momento de realizar los cálculos y al simular los sistemas.

Como se mencionó en la definición de solución numérica, la forma en que los computadores llegan a dar solución a los sistemas es por medio de algoritmos que han sido desarrollados y probados a lo largo de la historia, entre los cuales se encuentran

- Método de bisección
- Método de Newton
- Método bilineal o método de Tustin
- Método de Euler
- Método de Runge-Kutta de orden cuatro

Siendo estos últimos dos de los más utilizados debido a su sencillez o debido a que brindan una respuesta más fiel o cercana a la solución real, sin mencionar que pueden aplicarse tanto para sistemas lineales como no lineales a diferencia del método bilineal que está limitado a sistemas LTI.

Finalmente, es importante dar una definición más formal a un término que ya se ha mencionado vagamente: la simulación. De acuerdo a [8] la simulación es una actividad que puede considerarse como un experimento efectuado sobre el modelo con ayuda de una computadora digital, en contraste con los experimentos físicos, los cuales se realizan directamente sobre el sistema con ayuda de transductores. La simulación es, en resumen, la herramienta final que permite observar cómo reacciona el sistema de forma natural y realizar predicciones, o cómo reacciona al aplicarle control y determinar si el comportamiento es el deseado.

6.2. Control de sistemas dinámicos

Existen diversos métodos de realizar control sobre los modelos que aproximan a los sistemas dinámicos. Estos métodos pueden ser de control clásico o moderno, los cuales se describen a continuación.

6.2.1. Control clásico

Al hablar de control clásico se hace referencia a los primeros métodos de control que se desarrollaron gracias a mentes reconocidas de finales del siglo XIX y mediados del siglo XX como las de Routh, Hurwitz, H. Nyquist y H.W. Bode, que dejaron plasmado su trabajo en una serie de teoremas que rigen el control clásico. En esta perspectiva de control se utilizan herramientas matemáticas como los teoremas antes mencionados y la ampliamente conocida transformada de Laplace para expresar los sistemas como funciones de transferencia. Sin embargo, estas herramientas clásicas, a pesar que pueden ser aplicadas para resolver una amplia cantidad de problemas, presentan muchas limitaciones en cuanto a la forma o naturaleza de los sistemas ya que su aplicación se limita a la minoría de sistemas llamados Lineales Invariantes en el Tiempo (LTI por sus siglas en inglés).

Como se mencionó con anterioridad, a pesar de sus limitaciones, el control clásico ofrece una buena solución para una amplia gama de aplicaciones, esto gracias al reconocido controlador PID que debe su nombre a los tres bloques que lo conforman. Cada uno de estos bloques actúan de una forma distinta sobre la señal de error, la cual representa qué tan alejado se encuentra el sistema de alcanzar el comportamiento deseado. La expresión para el error está dada por

$$e(t) = r - y(t) \tag{2}$$

donde $e(t)$ es el error, $y(t)$ es la salida del sistema y r es la referencia del sistema. Cabe aclarar que r puede ser variante en el tiempo, en cuyo caso se expresa como $r(t)$.

A continuación, y de acuerdo a [9], se describe la acción que cada bloque ejerce sobre

dicha señal y cómo afecta a la señal de salida $u(t)$ del controlador:

- Bloque proporcional (P): el cual actúa como un amplificador de ganancia ajustable K_p que hace que la salida $u(t)$ sea proporcional a la señal de error.

$$u(t) = K_p e(t) \quad (3)$$

donde K_p es la ganancia ajustable llamada constante proporcional.

- Bloque integral (I): el cual provoca que la salida controlada $u(t)$ sea proporcional a la integral de la señal de error.

$$u(t) = K_I \int_0^t e(\tau) d\tau \quad (4)$$

donde K_I es llamada constante integral.

- Bloque derivativo (D): con el que la salida $u(t)$ es proporcional a la derivada de la señal de error.

$$u(t) = K_D \frac{de(t)}{dt} \quad (5)$$

donde K_D es llamada constante derivativa.

Este controlador constituye el máximo exponente de la teoría de control clásica y resulta muy efectivo para la resolución de muchos problemas de aplicación en la industria y su uso es muy popular.

Sin embargo, en problemas en donde el sistema a controlar es mucho más complejo, tiene múltiples entradas y/o salidas, y no cumple con los requerimientos para ser aproximado por un sistema LTI, el control clásico ya no presenta una solución.

6.2.2. Control moderno

Cuando se encuentran casos en que el control clásico ya no es suficiente, se recurre al control moderno el cual presenta soluciones, si bien muchas veces más complejas, libre de las limitaciones del control clásico. De acuerdo con [9], el control moderno surge debido a la necesidad de alcanzar metas cada vez más restrictivas en el desempeño de los sistemas de control, al aumento en la complejidad de los sistemas y gracias a los avances computacionales. Cabe resaltar que una característica principal del control moderno es que se basa en el concepto de estado, el cual, tal y como se expresa en [9], ha existido durante largo tiempo en el campo de la dinámica clásica.

Según [10] para el diseño de sistemas en el espacio de estados es imprescindible verificar en primera instancia la controlabilidad y la observabilidad del sistema, conceptos que determinan la existencia de una solución completa para un problema de diseño de un sistema de control.

La controlabilidad y la observabilidad son definidas en [11] como sigue:

- Un sistema controlable es aquel que converge a una referencia en un tiempo finito, es decir que sin importar cuál sea su estado actual en algún momento la respuesta se estabilizará en un punto definido.

Para comprobar controlabilidad se puede establecer que el rango de la matriz de controlabilidad γ coincida con el orden del sistema como se muestra a continuación

$$\text{rank}(\gamma) = n$$

con

$$\gamma = [B, AB, A^2B, \dots, A^{n-1}B]$$

donde A y B son las matrices del sistema y de control, respectivamente, y n es el orden del sistema; con esto se garantiza que en el comportamiento dinámico del sistema no exista un punto en el que se torne inestable.

- Un proceso es observable cuando los estados pueden ser determinados a partir de las salidas en cada instante de tiempo, es decir hay un acople entre estados y salidas. En caso de que un proceso sea no observable, con la información de las salidas no es posible conocer de manera precisa y completa el comportamiento del sistema.

Se puede comprobar la observabilidad sabiendo que el rango de la matriz Ω coincide con el orden n del sistema como se muestra a continuación

$$\text{rank}(\Omega) = n$$

con

$$\Omega = [C, CA, CA^2, \dots, CA^{n-1}]$$

Donde C es la matriz de salida del sistema.

Un dato curioso acerca del control moderno es que, a diferencia del control clásico, hace uso de herramientas matemáticas un tanto más antiguas y que en su tiempo no fueron valoradas, como los teoremas de Lyapunov.

Los principales controladores utilizados en control moderno LTI utilizan retroalimentación o realimentación de estado y estos se describen a continuación.

- **Ubicación de polos:** De acuerdo con [10] en este método se supone que todas las variables de estado son medibles y que están disponibles para la realimentación. Adicionalmente si se demuestra que el sistema considerado es de estado completamente controlable, los polos del sistema en lazo cerrado se pueden ubicar en cualquier posición deseada mediante una realimentación del estado a través de una matriz de ganancias K de la realimentación del estado.

La expresión para el control se elige como

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (6)$$

donde \mathbf{K} es la matriz de ganancias de realimentación y se determina en función de la ubicación de los polos y las matrices \mathbf{A} y \mathbf{B} .

En [10] se encuentran una serie de pasos para diseñar un controlador mediante la ubicación de polos, sin embargo, herramientas como el software MATLAB permiten ahorrarse esos cálculos y obtener el control expresado en (7) con el siguiente comando

$$\gg \mathbf{K} = \text{place}(\mathbf{A}, \mathbf{B}, p)$$

donde p es el vector de polos deseados y \mathbf{A} y \mathbf{B} son las matrices del sistema y de control, respectivamente.

- **Control LQR:** De acuerdo con [11] el controlador LQR es una estrategia de control óptima ya que se procura la operación del sistema dinámico al menor costo, es decir que los ajustes del controlador proporcional se encuentran usando un algoritmo matemático que minimiza la función de coste o desviaciones no deseadas.

Este controlador es útil tanto para sistemas lineales como para sistemas no lineales. En esta investigación se estudia un sistema no lineal, por lo que, como se especifica en [11], el cálculo de las constantes de control debe hacerse en función del punto de trabajo (x_{ss}, y_{ss}) , por eso es necesario escogerlo previamente dentro del margen de operación de la planta.

En [11] también se halla que la ley control LQR responde en esencia a una acción de control proporcional

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (7)$$

que para un punto de operación en específico (x_{ss}, u_{ss}) se transforma en

$$\mathbf{u} = -\mathbf{K}(\mathbf{x} - \mathbf{x}_{ss}) + \mathbf{u}_{ss} \quad (8)$$

en la que debe determinarse con cálculos matemáticos los componentes de la matriz \mathbf{k} que representa las constantes del control proporcional.

El control LQR corresponde a una optimización de la función \mathbf{J}

$$\mathbf{J} = \int_0^{\infty} [\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}] dt \quad (9)$$

Donde \mathbf{Q} y \mathbf{R} son matrices de penalización o matrices de ponderación de error y de control, respectivamente, que deben cumplir con las siguientes características de acuerdo a [11]

- \mathbf{Q} y \mathbf{R} deben ser matrices positivas definidas.
- Con frecuencia se escoge \mathbf{Q} y \mathbf{R} diagonales asignando valores grandes a las variables que se quieran minimizar.
- Si \mathbf{Q} contiene valores grandes, la exigencia al control será mayor.

- Si R contiene valores grandes, se incrementa la influencia de las acciones de control sobre la planta.
- Q y R deben ser matrices positivas definidas.

Una vez más este control puede determinarse por medio del software MATLAB con el siguiente comando

$$\gg \mathbf{K} = \text{lqr}(A, B, Q, R)$$

donde Q y R son las matrices de penalización, que muchas veces corresponden a matrices identidad, y A y B son las matrices del sistema y de control, respectivamente.

6.3. El método de Euler-Lagrange para modelado

Este método utiliza la ecuación que lleva el mismo nombre, ecuación de Euler-Lagrange, la cual se muestra a continuación

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = Q \quad (10)$$

Donde L es el Lagrangiano, q son las coordenadas generalizadas y Q la fuerza generalizada. Cabe resaltar que q puede ser un vector, en cuyo caso la ecuación (6) representa un conjunto de n ecuaciones de la siguiente forma

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i, i = 1, 2, 3, \dots, n \quad (11)$$

Además, para un sistema dinámico el vector q representa las posiciones, velocidades y aceleraciones ya sea lineales o angulares de los cuerpos acoplados al sistema.

Una cantidad importante es L el cual está definido como

$$L = T - U \quad (12)$$

donde T es la energía cinética del sistema expresada como

$$T = \frac{1}{2}mv^2 \quad (13)$$

donde v es la magnitud de la velocidad o el vector de magnitudes de las velocidades ya sea lineales o angulares de los cuerpos de interés y m sus masas correspondientes.

Y U es la energía potencial, expresada como

$$U = mgh \quad (14)$$

donde m es nuevamente la masa, g es la aceleración de la gravedad y h es la altura a la cual se encuentra el cuerpo respecto a un marco de referencia.

La derivación de esta ecuación puede consultarse en [12], así como varios ejemplos de su aplicación.

Este método es uno de los muchos existentes para derivar las ecuaciones dinámicas de los sistemas, y resulta muchas veces una mejor alternativa al método de Newton-Euler que hace uso de las ecuaciones de las leyes de Newton sobre las fuerzas y momentos acoplados, y cuyo planteamiento puede tornarse sumamente intrincado al analizar sistemas de complejidad más elevada. En este trabajo, al contar con un sistema con considerable complejidad se decidió hacer uso de la ecuación de Euler-Lagrange, la cual a diferencia de las de Newton-Euler, plantea una solución desde el punto de vista energético, lo cual, como muchas veces se ha observado, reduce la complejidad de las ecuaciones obtenidas para un sistema.

6.4. Otros métodos de control

El robot con salto que se busca desarrollar puede llegar a modelarse como un robot con una pata, lo cual es el tema principal en [13], que desarrolla diversos temas acerca de máquinas o robots con patas, iniciando en el caso más sencillo: un robot que salta con una pata. El modelo inicial de este es tomando intuición acerca del péndulo invertido para simplificar el comportamiento real, y por lo tanto mucho más complejo, de una extremidad, y además, enfocando el análisis en el balance o equilibrio y no en la coordinación de varias patas ya que solamente se cuenta con una.

El método de control en que se basa esta referencia es el control proporcional derivativo PD, que ya se mencionó con anterioridad y que constituye uno de los máximos exponentes del control clásico. Adicionalmente, el control es aplicado separándolo en tres fases o partes y, lo más importante, sin tener un modelo matemático del sistema, ya que todos los datos necesarios para conocer el comportamiento del mismo, son obtenidos a partir de pruebas experimentales. Este es el resultado principal en que se enfoca esta investigación. Según [13] el modelo matemático de los sistemas por lo general no se puede obtener debido a su complejidad, por lo que existen alternativas para aplicar control a los mismos realizando ciertos experimentos para obtener mediciones que reflejen el comportamiento del sistema y con estos realizar aproximaciones numéricas.

Esto es de gran importancia para la presente investigación ya que, de resultar demasiado complejo el desarrollo del modelo matemático del sistema, es posible aplicar control al mismo siguiendo una metodología similar presentada en el experimento de la referencia mencionada. En esta se desarrollan expresiones simples que podrían replicarse y adaptarse al modelo del presente estudio para obtener mediciones y cumplir el objetivo de control.

6.5. Conservación de la energía y la cantidad de movimiento lineal

Para un sistema aislado, es decir, un sistema en el que solamente actúan fuerzas conservativas, la energía mecánica del mismo se conserva [14]. La energía mecánica de un sistema se expresa como la suma de su energía cinética K y su energía potencial (de cualquier clase) U , como sigue

$$E_{mec} = K + U \quad (15)$$

Se dice que la **energía mecánica se conserva** si el cambio en la misma es igual a 0

$$\Delta E_{mec} = 0 \quad (16)$$

$$\Delta E_{mec} = \Delta K + \Delta U = 0 \quad (17)$$

Lo que significa que la energía del sistema cambia dinámicamente entre cinética y potencial. Esto se observa más claramente si se expresa como

$$\Delta K = -\Delta U \quad (18)$$

En la que se observa que una disminución de energía cinética equivale a un aumento en la energía potencial y viceversa.

Según [14] **la cantidad de movimiento lineal** \vec{p} de una partícula u objeto se define como el producto de la masa m de la partícula y la velocidad \vec{v} a la que se mueve.

$$\vec{p} = m \vec{v} \quad (19)$$

Es posible relacionar la cantidad de movimiento lineal que experimenta un objeto con la fuerza resultante que actúa sobre el mismo si se hace uso de la segunda ley de movimiento de Newton, la cual se expresa como

$$\sum \vec{F} = m \vec{a} \quad (20)$$

y dado que la aceleración es la razón de cambio con el tiempo de la velocidad, se tiene que

$$\sum \vec{F} = m \frac{d\vec{v}}{dt} \quad (21)$$

en la que, dado que la masa m es constante la misma puede introducirse dentro de la derivada

$$\sum \vec{F} = \frac{d(m\vec{v})}{dt} \quad (22)$$

de lo que se observa que la fuerza resultante que actúa en una partícula es igual a la razón de cambio con el tiempo de la cantidad de movimiento lineal que ésta experimenta

$$\sum \vec{F} = \frac{d\vec{p}}{dt} \quad (23)$$

6.6. Inventor y Webots

Según sus creadores, Autodesk Inventor es un software CAD que proporciona herramientas de diseño mecánico, documentación y simulación de productos 3D de nivel profesional y que permite trabajar eficientemente con una potente combinación de capacidades de diseño paramétrico, directo, de forma libre y basadas en reglas. En [15] se ofrece una serie de vídeos para familiarizarse con el entorno de este software.

Webots es descrito en su página oficial como una aplicación de escritorio de código abierto y multiplataforma utilizada para simular robots, y que proporciona un entorno de desarrollo completo para modelar, programar y simular robots. Este simulador ha sido diseñado para un uso profesional, y es ampliamente utilizado en la industria, la educación y la investigación. En [16] se encuentra un enlace a los tutoriales creados por los diseñadores de este software, el cual brinda una buena introducción al entorno y una excelente base para comenzar en el modelado, creación, simulación y programación de robots.

En esta investigación se utiliza Autodesk Inventor para llevar a cabo el diseño mecánico del robot y Webots para simular su funcionamiento. Sin embargo, para trasladar el modelo realizado en Inventor es necesario exportarlo en cierto formato para posteriormente, haciendo uso de otro software, convertirlo al formato `wrl` que Webots maneja. El software utilizado para realizar dicha conversión es Blender, el cual en su página oficial es descrito como la suite de creación 3D gratuita y de código abierto que admite la totalidad de la canalización 3D: modelado, aparejo, animación, simulación, renderizado, composición y seguimiento de movimiento, edición de video y canalización de animación 2D.

El procedimiento realizado para trabajar en Webots con un modelo creado en Inventor se resume a los siguientes pasos:

1. Desarrollar modelo en Autodesk Inventor.
2. Exportar el modelo como `.obj`.
3. Importar en Blender el archivo `.obj`.
4. Exportar, en Blender, el archivo como `.wrl`.

5. Importar en Webots el archivo `.wrl`.

Este proceso se explica detalladamente en [17], en el que también se mencionan consideraciones importantes para realizar una correcta conversión.

6.7. Nodos principales en Webots

Para desarrollar las simulaciones Webots hace uso de sus Nodos, los cuales son las “piezas básicas más simples” a partir de las cuales el software trabaja para recrear entornos, objetos, robots, etc. A continuación, se describen algunos de los principales nodos y que son de interés para esta investigación.

6.7.1. Nodo WorldInfo

Este nodo permite la personalización del entorno que se desea simularse. Para ello cuenta con varios campos como los siguientes:

- Gravedad: que permite definir la aceleración de la gravedad que se usará en la simulación y el eje a través del cual será aplicada. Por defecto se aplica en la dirección negativa de y y tiene un valor de $9.81m/s^2$.
- ERP: *ErrorReductionParameter* o parámetro de reducción de errores, es el campo que utiliza el código de simulación de Webots para gestionar las uniones de contactos. El ERP especifica qué proporción del error de la junta de contacto se corregirá durante el siguiente *timestep*. De acuerdo a [18] si $ERP = 0$, entonces no se aplica ninguna fuerza de corrección y los cuerpos eventualmente se separarán a medida que avanza la simulación. Si $ERP = 1$, la simulación intentará corregir todos los errores articulares durante el siguiente paso de tiempo. Sin embargo, no se recomienda establecer $ERP = 1$, ya que el error de la articulación no se solucionará por completo debido a varias aproximaciones internas. Se recomienda un valor de $ERP = 0.1$ a 0.8 .
- Período de simulación o *basicTimeStep*: que define el período de simulación con el que se ejecuta las simulaciones de Webots. Está expresado en milisegundos y el mínimo valor posible es 1.
- Dirección del norte: que permite modificar la dirección en que la dirección norte se encuentra dentro del entorno de simulación. Este campo es de especial importancia al utilizar dispositivos como acelerómetros, giroscopios o unidades de medición inercial, ya que dependiendo de su configuración los dispositivos cambiarán las mediciones que brindan.

6.7.2. Nodo Robot

El cual como su nombre lo indica puede ser utilizado como base para crear un robot, ya sea simple, articulado, humanoide, móvil con ruedas, etc. Los principales campos con que

cuenta este nodo son:

- Controlador: el cual permite añadir el controlador programado que el robot estará utilizando durante la simulación.
- Supervisor: que es un set de funciones extra que permiten acceder a información o determinar comportamientos del robot. La característica de mayor interés de este set de funciones para esta investigación es *wb_supervisor_node_add_force* con la cual es posible agregar fuerzas de magnitud y dirección determinadas al robot.

6.7.3. Nodo Solid

Este permite crear o representar un objeto cuyas propiedades físicas, como dimensiones, material, masa o densidad, pueden ser configuradas. Este nodo es la base para la detección de colisiones que emplea Webots. Los robots y dispositivos son sub-clases del nodo Solid. Los campos principales de este nodo son

- boundingObject: el cual permite especificar la forma geométrica que Webots utilizará para simular las colisiones y detectarlas. Si este campo se deja vacío el objeto podrá atravesar objetos y ser atravesado por otros.
- Física: este campo permite definir las propiedades como la densidad o la masa, el centro de masa y coeficientes de amortiguamiento lineal y angular.

6.7.4. Nodo Shape

Este nodo permite crear objetos renderizados en el entorno de Webots. Dichos objetos están constituidos por una geometría, como un cubo, cono, esfera, y por una apariencia, que permite modificar la textura del objeto, su color, entre otros. Este nodo es usualmente utilizado para definir la forma de un sólido, lo que significa que este nodo se puede, o no, encontrar dentro de un nodo Solid.

6.7.5. Nodo Joint

Este nodo es de la clase de nodo abstracto, lo que significa que es un nodo que agrega atributos a un objeto y no un objeto en sí. De este se derivan los nodos que representan el tipo de junta que puede ser agregada a un robot. Dentro de estas junta se encuentran la revoluta, prismática y esférica. Este nodo también permite agregar el dispositivo que actuará dicha junta, como un motor rotacional, para una junta revoluta, o uno lineal, para una junta prismática.

7.1. Diseño mecánico

Basado en las observaciones realizadas sobre los proyectos similares que se han llevado a cabo, y las cuales están recopiladas en los antecedentes, se procedió a diseñar al robot de forma que cumpliera con los requerimientos que dicta uno de los objetivos propuestos: accionar el mecanismo de salto con un motor y resortes torsionales, además de poseer una barra o palanca de inercia la cual será accionada por medio de otro motor. El diseño se llevó a cabo en el software Autodesk Inventor, al cual de ahora en adelante se referenciará simplemente como Inventor. En la Figura 1 se muestra el proceso de diseño utilizado para la selección de componentes.

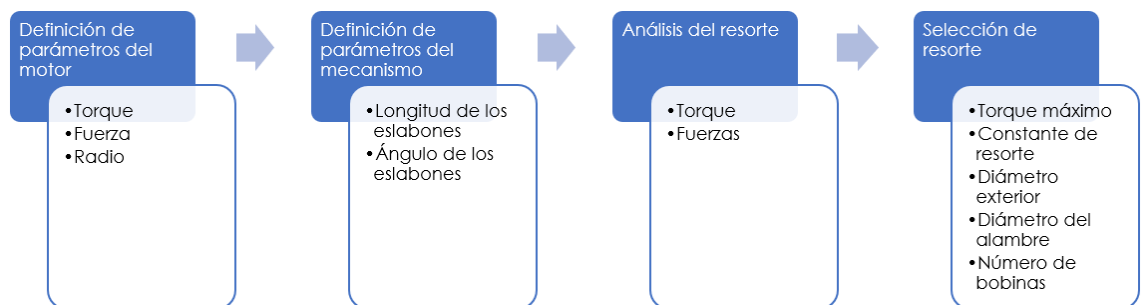


Figura 1: Proceso de diseño para selección de los motores y resortes.

7.2. Migración del diseño al entorno de simulación

El entorno de simulación utilizado, como ya se mencionó, fue Webots, en el cual se importó el diseño mecánico del robot realizado en Inventor. Para ello fue necesario llevar a cabo una conversión de formatos: de `.iam`, que es el formato de un ensamble en Inventor, a `.obj`, que es uno de los formatos en que se puede exportar el ensamble. Se realizó esta conversión para que así, en el software Blender, fuese posible importar el robot como `.obj` y así convertirlo al formato `.wrl` que es el formato con el que se trabaja en el entorno de Webots, formato al cual es imposible convertir directamente desde Inventor.

El procedimiento realizado para trabajar en Webots con un modelo creado en Inventor se resume a los siguientes pasos:

1. Desarrollar modelo en Autodesk Inventor.
2. Exportar el modelo en formato `.obj`.
3. Importar en Blender el modelo `.obj`.
4. Exportar, en Blender, el archivo como `.wrl`.
5. Importar en Webots el archivo `.wrl`.

7.3. Obtención de datos y mediciones del sistema

Los datos de interés en esta investigación son el ángulo roll: del robot y cómo varía éste en función del torque que ejerce la palanca de inercia. Para obtener dicha relación, como primer paso se incorporó una Unidad de Medición Inercial (IMU por sus siglas en inglés), que brinda la medición deseada, roll:, además de los ángulos yaw: y pitch: y con la cual también es posible obtener sus velocidades angulares respectivas. Posteriormente, se realizaron experimentos en los que se varió el torque ejercido por la palanca de inercia y se almacenaron los valores que tomó la velocidad angular del pitch en función de esas variaciones de torque; esto con el fin de hallar la relación entre estas dos cantidades y así ser capaz de generar un sistema o planta o una linealización de la misma, a la cual realizar control.

7.4. Modelado del sistema

Con los datos obtenidos de forma experimental en el entorno de Webots, se realizó el análisis de los mismos en el software Matlab. Se graficaron los datos y se buscó la relación matemática entre ellos para así obtener un sistema, a partir de comparaciones con otros sistemas, simplificaciones o linealizaciones, al cual aplicarle control.

7.5. Desarrollo e implementación del control

Con el modelo del sistema es posible evaluar cuál método de control es el más conveniente aplicar dependiendo de su comportamiento y complejidad, además de tomar en cuenta las ventajas específicas que cada método de control brinda. En esta investigación dado que el sistema en estudio era Single Input Single Output, y por la simplicidad en estructura y uso de un controlador clásico, se planteó implementar, como primera opción, un controlador PD. Dado que este controlador logró estabilizar efectivamente la orientación del robot, la implementación de un LQR se descartó.

7.6. Realización de pruebas

Luego de implementar el control y observar los resultados de forma gráfica, se procedió a implementar el control al robot en el entorno de Webots, para así poner a prueba el control de una forma más realista y observar el comportamiento físico del sistema. Además, para comprobar la robustez del controlador, se realizaron pruebas en distintos escenarios.

La metodología de este proyecto se resume en el diagrama mostrado en la Figura 2

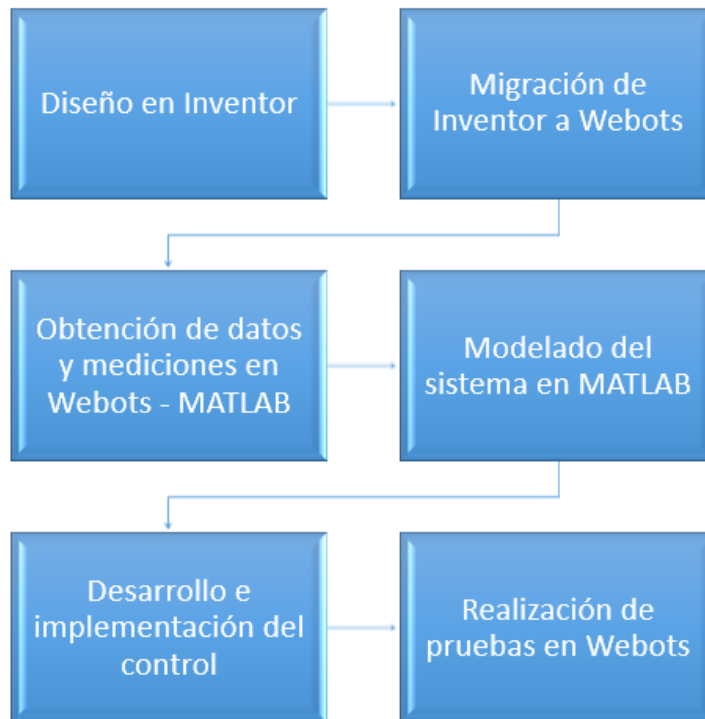


Figura 2: Metodología del proyecto.

8.1. Definición de parámetros del motor.

Tal y como se muestra en el proceso de diseño de la Figura 1, el primer paso a realizar fue la definición de los parámetros del motor tales como el torque, el radio de su eje y la fuerza que puede ejercer. El torque y el radio se obtuvieron de la hoja de datos de los dispositivos, las cuales están disponibles en el sitio web del distribuidor [19], mientras que la fuerza se calculó a partir de estos dos como sigue:

El torque puede expresarse como

$$\tau = Fd \quad (24)$$

Siendo τ el torque del motor y d el radio de su eje, es posible obtener la fuerza que puede ejercer el motor despejando para F

$$F = \frac{\tau}{d} \quad (25)$$

8.2. Definición de parámetros del mecanismo.

Como siguiente paso se realizó la definición de los parámetros del mecanismo, tales como la longitud de los eslabones y el ángulo respecto a la horizontal que tendrán los mismos cuando el mecanismo está comprimido. El mecanismo se definió como se muestra en la Figura 3 de modo que en las uniones A, C, D y G exista un par de resortes torsionales para

almacenar energía para realizar el salto. Un cable conectado a la base inferior del mecanismo se enrollará en el eje de un motor y permitirá que los resortes se compriman entre las bases superior e inferior gracias a la fuerza que ejerce dicho motor. Posteriormente, la fuerza será retirada y la energía potencial de los resortes será liberada provocando el salto.

Este mecanismo está inspirado en el desarrollado en [4], debido al método simple de almacenamiento de energía, las relativamente elevadas alturas que se pueden alcanzar con dicha energía y al reducido tamaño que puede llegar a tener el mecanismo.

Las dimensiones de ancho y profundidad del mismo se determinaron en función de los dispositivos que lo formarán como los motores, drivers, la batería y el microcontrolador en su placa PCB; dicho procedimiento se explica más a detalle en la sección de diseño CAD. La altura H del mecanismo se escogió empíricamente con la única restricción que la altura total del robot no superara los 10 cm para mantenerlo en el rango de altura de los robots investigados en los antecedentes. Se definió entonces a $H = 5\text{cm}$ tomando en cuenta que con los motores montados en la base superior la altura del robot será de aproximadamente 8 cm, cumpliendo con la restricción. Los eslabones tienen longitudes $L1 = L2 = L3 = L4$, y un ángulo θ respecto a la horizontal. Dichas longitudes se escogen de 20mm de modo que cuando los resortes se encuentren relajados el ángulo de los eslabones sea de 75° con respecto a la horizontal y en esa configuración la altura sea menor a la establecida, 5cm, para tomar en consideración la altura que las bases inferior y superior añadirán para garantizar que los eslabones puedan rotar sin interferencia en las juntas A, C, D y G; nuevamente estas consideraciones son más claras en la sección de diseño CAD. El ángulo de 75° se escogió así para lograr que los resortes puedan comprimirse hasta 60° sin que exista interferencia entre los eslabones.

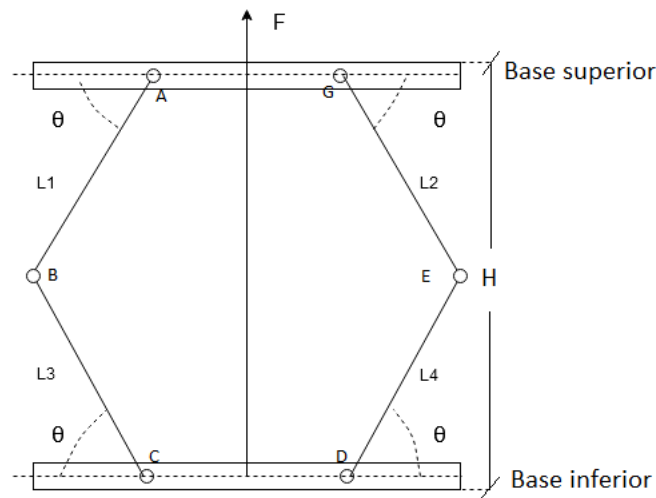


Figura 3: Mecanismo de salto del sistema.

8.3. Análisis del resorte.

Posteriormente, con el mecanismo definido, se procedió a calcular las fuerzas y torques que experimentará el mismo. Para ello se realiza el análisis del cuerpo libre de la Figura 4, en el que se muestran las fuerzas f que deben ejercerse para torsionar los resortes y sus componentes f_x y f_y . En este diagrama se muestran solamente dos eslabones ya que los dos restantes tienen el mismo análisis pero con direcciones contrarias. Se planteó que todos los resortes torsionales utilizados fueran iguales, por lo que se halló que la fuerza F que el motor debe ejercer para torsionarlos es n veces la componente vertical f_y de la fuerza que se requiere para torsionar solamente uno de ellos, donde n es el número de resortes torsionales que el mecanismo posea.

$$F = nf_y \quad (26)$$

Se tiene de (26) la fuerza máxima que puede ejercer el motor. Se puede hallar entonces, por 27, la componente vertical de la fuerza que se aplicará a cada uno de los resortes. Con esta fuerza y conociendo el ángulo θ_e de los eslabones del mecanismo, es posible calcular la fuerza neta f que se ejercerá sobre cada resorte y con esta, el torque τ_e generado en la junta correspondiente (A, C, D o G) conociendo la longitud L_n de los eslabones del mecanismo.

$$f = \frac{f_y}{\cos(\theta_e)} \quad (27)$$

$$\tau_e = fL_n \quad (28)$$

Es importante, además, conocer el torque τ_r que el resorte ejercerá en contra de la fuerza f , esto con el fin de asegurar que dicho torque es menor que el torque τ_e para que así el mecanismo sea capaz de torsionar al resorte a su ángulo máximo de deformación θ_{max} , el cual es calculado para cada resorte con la forma angular de la Ley de Hook

$$\theta_{max} = \frac{\tau_{r,max}}{k} \quad (29)$$

donde $\tau_{r,max}$ es el torque máximo que resiste el resorte y k la constante de resorte del mismo.

Adicionalmente, también debe verificarse que el torque máximo generado en la junta $\tau_{e,max}$ no exceda a $\tau_{r,max}$ ya que de lo contrario el resorte se deformaría plásticamente.

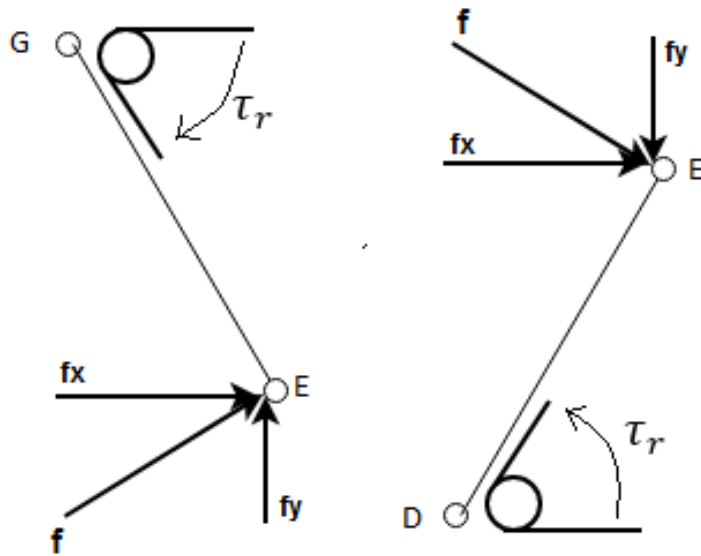


Figura 4: DCL de los eslabones del mecanismo.

8.4. Selección de resorte.

Con las consideraciones anteriores, se procedió a buscar el resorte a emplear, el cual, por facilidad, se busca que sea prefabricado para no emplear más tiempo en la parte del diseño mecánico. Para ello se utilizó el catálogo de muelles del proveedor Vanel, el cual también ofrece diseños CAD de los productos de dicho catálogo [20]. Se encontró que los ángulos de resortes que manejan son de 90° , 180° y 270° . Se buscó entre los muelles con ángulos de 90° cuyo diámetro exterior se encuentre entre los 10 y 12 mm debido al espacio disponible en el mecanismo, y que sean capaces de almacenar suficiente energía para generar un salto de al menos 4 veces la altura del mecanismo.

8.5. Bio-inspiración.

Con los posibles resortes a utilizar, se empleó el concepto de conservación de la energía para estimar la altura máxima que el mecanismo alcanzaría con cada tipo de resorte; esto con el fin de lograr que el robot salte, como ya se mencionó, al menos 4 veces su altura, característica que poseen en particular las langostas de desierto adultas que son capaces de saltar al menos 4 veces su longitud, la cual se encuentra alrededor de los 6 cm, y logran saltos de 25cm según [21]. Para hallar esta altura primero se calculó la energía almacenada en los resortes, la cual está dada por

$$U = n \frac{1}{2} k \theta^2 \quad (30)$$

En la cual el factor n es debido a la presencia de n resortes. Posteriormente, suponiendo que la energía se conserva, toda la energía potencial U en los resortes se convertirá en energía potencial $U_p = mgh$ cuando el robot alcance su altura máxima h , por lo que se tiene la siguiente expresión

$$U = mgh \quad (31)$$

De la que se tiene que h es

$$h = \frac{U}{mg} \quad (32)$$

Donde g es la aceleración de la gravedad, $9.81m/s^2$, y m es la masa de todo el sistema, la cual se estimó con el diseño CAD del mecanismo y con información de los componentes provista por su hoja de datos o por los distribuidores.

Cabe señalar también que el control de la orientación durante el salto es también inspirado por la naturaleza, ya que animales como las lagartijas emplean su cola para controlar la posición en la que aterrizan, como se estudia en [22].

8.6. Componentes físicos.

Los componentes recomendados se listan en el Cuadro 1. La selección de estos, como se mencionó en el alcance, está basada en la experiencia que el autor ha tenido utilizándolos y observando que cumplan con los requerimientos, además de su fácil accesibilidad en el caso de motores, drivers y la unidad de medición inercial.

Los motores propuestos son de los fabricantes de electrónica Pololu que ofrecen una alta gama de modelos de alta y baja potencia y con distintas relaciones de reducción lo cual ofrece una variedad de capacidad de torque. La selección de estos motores radica en dicha variedad y en que, a pesar de su reducido tamaño, es posible incorporar un encoder para realizar mediciones pertinentes, característica que muchos motores de pequeñas dimensiones no poseen. Los motores propuestos son de potencia baja (6V y 0.36A) y se muestran, junto con algunas características, en el Cuadro 2; una descripción más detallada se muestra en la página del distribuidor [19].

El driver sugerido para estos motores es el DRV8833 debido a su reducido tamaño, peso, relativamente bajo voltaje de alimentación (2.7 V) y a su alta capacidad para suministrar de corriente (1.2A) a dos pequeños motores (como los sugeridos) y de permitirle el giro en ambos sentidos; este componente también es distribuido por Pololu y sus características más a detalle se muestran en [23].

El microcontrolador recomendado es el TM4C1233H6PM, el cual es el controlador que incorpora Tiva C Series MCU y que cuenta con un costo relativamente bajo en relación a su gran capacidad; además, es posible programarlo en el lenguaje de programación Energía (lenguaje C), lo cual representa una gran ventaja debido a que cuenta con una amplia

documentación y ejemplos a los cuales se pueden acceder en [24], y con el cual el autor ya ha interactuado.

Para la medición del ángulo pitch del sistema se sugiere el módulo MPU6050 que incorpora un acelerómetro y un giroscopio, con los cuales pueden realizarse otras mediciones de interés como detección de caída libre y la velocidad angular, sin necesidad de incorporar más módulos; además, su bajo voltaje de alimentación y su reducido tamaño hacen ventajoso su uso en esta aplicación. Un sitio para obtenerlo y para observar sus características técnicas se encuentra en [25].

La batería propuesta para esta aplicación es una lipo 3.7V de 150mAh, con la cual, tomando en cuenta que la demanda del sistema es de aproximadamente 360mA (corriente para los motores que es mucho mayor a la corriente que requieren los demás componentes), se tienen 25 minutos de uso continuo del robot ya que los motores se usarán uno a la vez.

Motor RB-Pol-4XX
DRV883 Dual Motor Driver Carrier
Módulo MPU6050
Microcontrolador TM4C1233H6PMI
Batería lipo 3.7V 150mAh

Cuadro 1: Componentes sugeridos.

Modelo	Reducción	Torque (Nm)	Diámetro del eje (mm)	Peso (g)
RB-Pol-446	50:1	0.0431	3.0	10
RB-Pol-447	75:1	0.0628	3.0	10
RB-Pol-448	100:1	0.0726	3.0	10
RB-Pol-452	298:1	0.1961	3.0	10

Cuadro 2: Características de los motores recomendados.

Finalmente, y basados en los datos de los componentes, en el Cuadro 3 se muestra la masa total del sistema aproximando la del mecanismo.

Componente	Peso (g)
Motor 1	10.0
Motor 2	10.0
Drivers	1.00
Microcontrolador	0.343
IMU	2.10
Batería	30.0
Mecanismo	81.0
Masa total	119.0

Cuadro 3: Masa total (desglosada) del sistema.

8.7. Resultados.

Con los datos obtenidos fue posible llevar a cabo el procedimiento explicado anteriormente empleando las ecuaciones (25) - (33). Cabe resaltar que este procedimiento se realizó iterativamente en función de los 4 motores, de los posibles resortes a utilizar, y en función del ángulo máximo de deformación θ_{max} de los mismos. Todo esto con el fin de asegurar que los resortes empleados pueden torsionarse lo suficiente para almacenar la energía necesaria para el salto requerido y que el motor fuera capaz de generar dicha fuerza para torsionarlos. Para emplear las ecuaciones (25) - (33) se requiere conocer la cantidad de resortes que empleará el mecanismo, la cual se escogió de $n = 8$ para que cada una de las juntas A, C, D, G cuente con 2 resortes y estos sean de tamaño reducido. Cabe mencionar que el ángulo máximo de deformación de los resortes se estableció en 60° para que los eslabones no colisionen con las demás partes del mecanismo, por lo que para los cálculos se debe tomar en cuenta lo siguiente

- Si el ángulo máximo de deformación de los resortes es superior a 60° , en los cálculos se utilizará el valor de 60° .
- Si el ángulo máximo de deformación de los resortes está definido por la capacidad del motor, es decir, el motor no es capaz de torsionar a los resortes hasta 60° , se utilizará en los cálculos el ángulo del resorte al cual se saturó el motor.

Además, la longitud elegida para los eslabones L_n se determinó de 0.02m.

Con las aclaraciones anteriores, se procede a mostrar, primeramente, en el Cuadro 4 las cantidades necesarias de los motores y, posteriormente, en los Cuadros 5 - 8 los posibles resortes a emplear así como los cálculos obtenidos a partir de sus datos. La fila correspondiente a F en el Cuadro 4 se obtuvo a partir de (26) y de los datos del Cuadro 2; la fila f_y a partir de (27), de F y la cantidad de resortes n . Las primeras tres columnas de los Cuadros 5 - 8 son datos obtenidos del catálogo de Vanel, la columna θ_{max} fue calculada a partir de (30) y de las dos columnas previas, la columna θ es el ángulo con el cual se realizaron los cálculos de energía y que, como se explicó con anterioridad, es 60° si el ángulo de deformación del resorte es superior a dicho valor y el motor es capaz de aplicar la fuerza necesaria para alcanzar los 60° , de lo contrario se empleará el ángulo máximo al que el motor pueda torsionarlo; su valor es determinado por dichas condiciones las cuales se ilustran en las figuras 5 - 10; dichas figuras fueron generadas con el software MATLAB y con esta misma herramienta se hallaron los valores de θ . La columna de E es la energía y se calcula a partir de (31), k , n y θ , la columna h es la altura que alcanzará el mecanismo y se calculó a partir de (33), y finalmente la columna r es la razón de salto calculada a partir de h como sigue

$$r = h/h_{robot} \quad (33)$$

donde h_{robot} es la altura del robot que se estima ronde los 8 cm, por lo que tiene una valor de 0.08m.

Reducción/ Parámetro	298:1	100:1	75:1	50:1
F (N)	130.76	48.38	41.84	28.77
f_y (N)	16.34	6.05	5.23	3.60

Cuadro 4: Parámetros importantes de los motores.

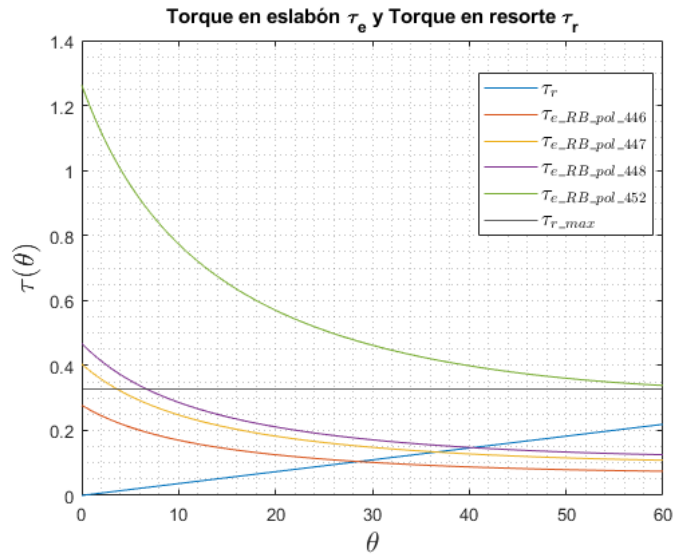


Figura 5: Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.104.130.0425.

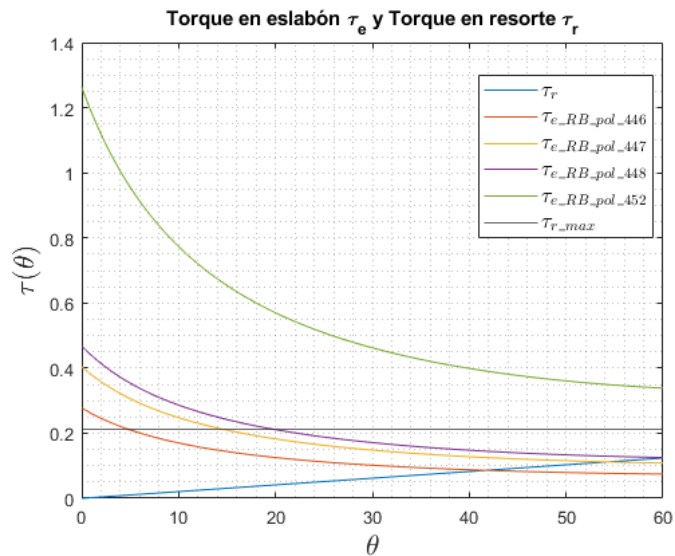


Figura 6: Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.110.125.0525.

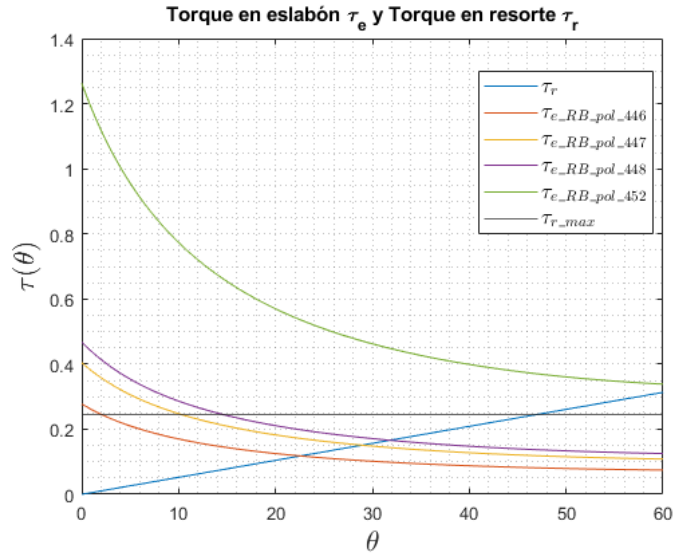


Figura 7: Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.118.125.0225.

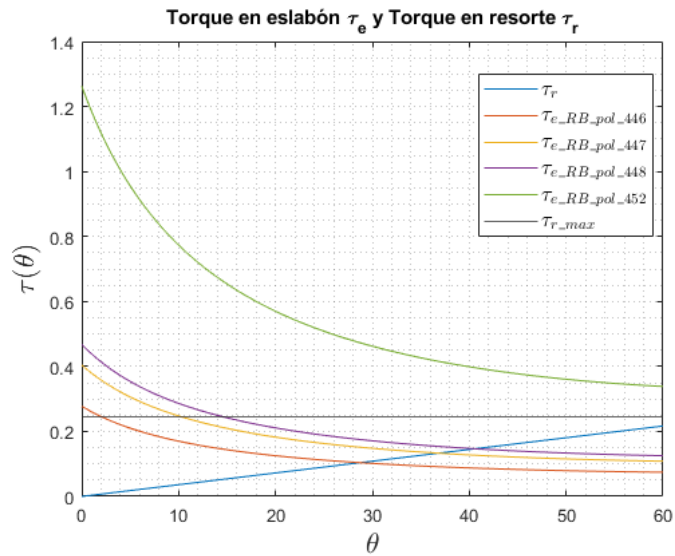


Figura 8: Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.128.125.0325.

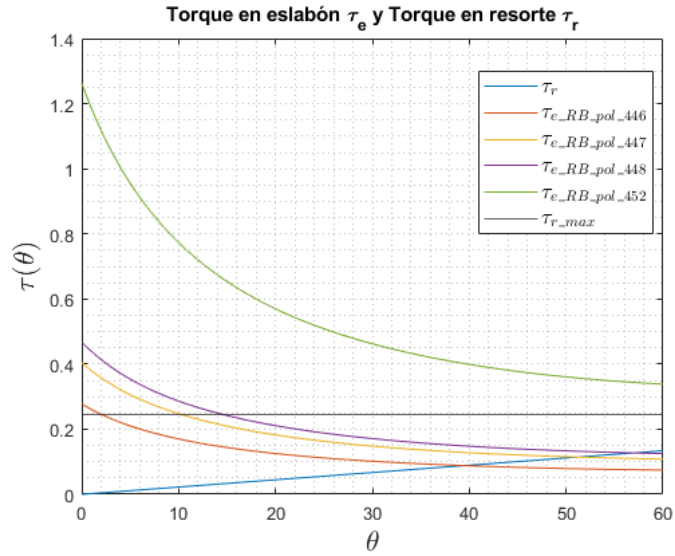


Figura 9: Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte G.128.125.0525.

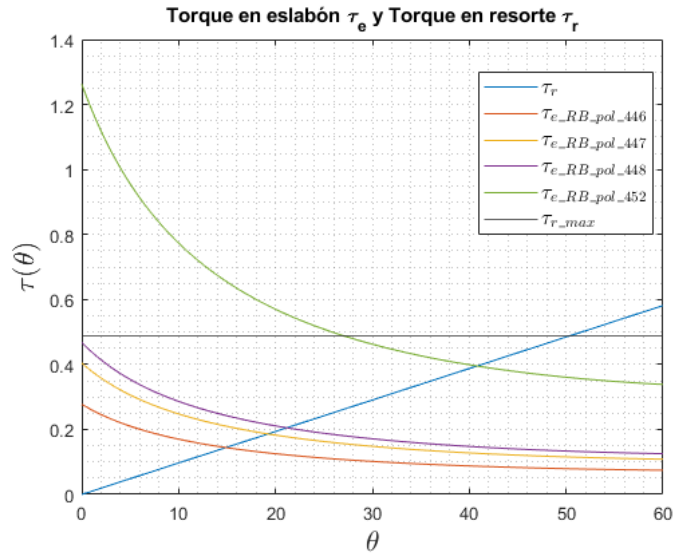


Figura 10: Gráfico de torques ejercidos en el eslabón con distintos motores y torque que ejerce el resorte D.121.160.0325.

Número de referencia del resorte	k (Nm/rad)	τ_{rmax} (Nm)	θ_{max} (deg)	θ (deg)	Energía (J)	h(m)	r
G.104.130.0425	0.209	0.328	90.001	60.000	0.207	0.177	2.21
G.110.125.0525	0.118	0.212	103.124	60.000	0.249	0.212	2.66
G.118.125.0225	0.299	0.243	46.600	46.565	0.185	0.158	1.98
G.128.125.0325	0.207	0.243	67.399	60.000	0.208	0.177	2.22
G.128.125.0525	0.128	0.243	108.800	60.000	0.243	0.207	2.59
D.121.160.0325	0.555	0.490	50.600	40.790	0.150	0.128	1.60

Cuadro 5: Posibles resortes y cálculos de interés usando el motor RB-Pol-446.

Número de referencia del resorte	k (Nm/rad)	τ_{rmax} (Nm)	θ_{max} (deg)	θ (deg)	Energía (J)	h(m)	r
G.104.130.0425	0.209	0.328	90.001	60.000	0.341	0.291	3.63
G.110.125.0525	0.118	0.212	103.124	60.000	0.422	0.360	4.50
G.118.125.0225	0.299	0.243	46.600	46.565	0.304	0.259	3.24
G.128.125.0325	0.207	0.243	67.399	60.000	0.341	0.291	3.64
G.128.125.0525	0.128	0.243	108.800	60.000	0.408	0.348	4.35
D.121.160.0325	0.555	0.490	50.600	40.790	0.249	0.213	2.66

Cuadro 6: Posibles resortes y cálculos de interés usando el motor RB-Pol-447.

Número de referencia del resorte	k (Nm/rad)	τ_{rmax} (Nm)	θ_{max} (deg)	θ (deg)	Energía (J)	h(m)	r
G.104.130.0425	0.209	0.328	90.001	60.000	0.413	0.353	4.41
G.110.125.0525	0.118	0.212	103.124	60.000	0.516	0.440	5.50
G.118.125.0225	0.299	0.243	46.600	46.565	0.368	0.314	3.92
G.128.125.0325	0.207	0.243	67.399	60.000	0.414	0.354	4.42
G.128.125.0525	0.128	0.243	108.800	60.000	0.505	0.431	5.39
D.121.160.0325	0.555	0.490	50.600	40.790	0.303	0.258	3.23

Cuadro 7: Posibles resortes y cálculos de interés usando el motor RB-Pol-448.

Número de referencia del resorte	k (Nm/rad)	τ_{rmax} (Nm)	θ_{max} (deg)	θ (deg)	Energía (J)	h(m)	r
G.104.130.0425	0.209	0.328	90.001	60.000	0.916	0.782	9.77
G.110.125.0525	0.118	0.212	103.124	60.000	0.516	0.440	5.50
G.118.125.0225	0.299	0.243	46.600	46.565	0.789	0.673	8.42
G.128.125.0325	0.207	0.243	67.399	60.000	0.906	0.773	9.66
G.128.125.0525	0.128	0.243	108.800	60.000	0.561	0.479	5.99
D.121.160.0325	0.555	0.490	50.600	40.790	1.124	0.960	11.99

Cuadro 8: Posibles resortes y cálculos de interés usando el motor RB-Pol-452.

De los resultados anteriores se observa que las mejores opciones en motores son el RB-Pol-448 y el RB-Pol-452, ya que con estos es posible alcanzar la razón de salto r mínima requerida. Sin embargo, se observa que con el motor RB-Pol-448 los valores de r son muy cercanos a 4, lo cual, dado que la altura fue calculada a partir de un modelo de energía que considera al sistema como una partícula y no como el cuerpo rígido que es, no es conveniente debido a que al usar los cálculos en el cuerpo rígido la razón puede ser menor a 4, por lo que se escogió el motor RB-Pol-452. Se observa que con este motor, cualquier resorte de los seleccionados es capaz de almacenar la energía necesaria para alcanzar la razón de salto deseada, sin embargo, se descartan los resortes G.110.125.0525 y G.128.125.0525 dado que los valores de r , nuevamente, son muy cercanos a 4. De los resortes restantes se escogió el modelo D.121.160.0325 ya que, observando la gráfica de la Figura 10, se tiene que con estos resortes se alcanza un equilibrio estático, por lo que no será necesario limitar el movimiento del motor para asegurar que el resorte no se contraiga de más y se deforme plásticamente.

8.8. Diseño CAD

El proceso de diseño llevado a cabo para realizar el modelo CAD del mecanismo se realizó en seis etapas:

1. Definición de la disposición de los motores.
2. Definición del ancho y profundidad del mecanismo.
3. Diseño de la base superior.
4. Diseño de la base inferior.
5. Diseño de los eslabones.
6. Diseño de la barra de inercia.

El robot empleará dos motores para su funcionamiento, uno de los cuales estará a cargo del almacenaje de la energía para realizar el salto, y el otro será el responsable de controlar la barra de inercia para el control de orientación. La disposición de estos dos motores debe asegurar que cada función se realizará sin que una interfiera con la otra, por lo que se define la disposición de los motores como se muestra en la Figura 11.

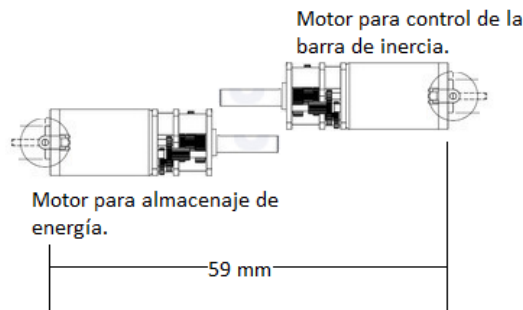


Figura 11: Disposición de los motores.

En la que se observa que ambos motores pueden llevar a cabo sus tareas sin ningún problema: el motor de almacenaje de energía puede girar en ambos sentidos y el motor de control también pero con un rango limitado que se especificará una vez esté definida la medida de profundidad del mecanismo.

Para definir el ancho y la profundidad del mecanismo se consideraron las dimensiones de los componentes más grandes, los motores, las cuales, según su hoja de datos [26], son de 12mm x 10mm x 34mm. En la Figura 11 se observa que los motores ocupan un espacio de 59mm, sin embargo, la medida de ancho se define de 50mm la cual procura espacio suficiente para asegurar los motores. La profundidad del mecanismo se define de 25mm debido a que esa es la dimensión más grande de la batería que se propuso, para así garantizar una buena sujeción de la misma. Con esta dimensión definida se obtuvo que el rango aproximado de movimiento para el motor que actúa la barra de inercia abarca 280° , como se esboza en la Figura 12.

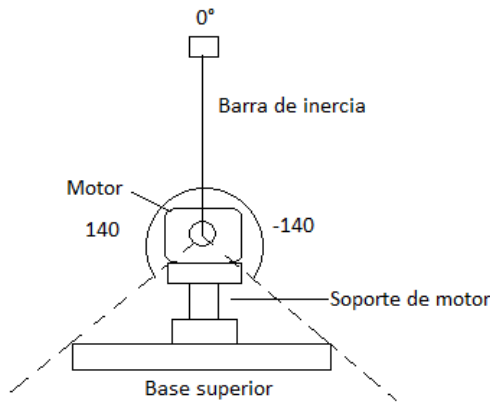


Figura 12: Rango de movimiento de la barra de inercia.

El diseño de la base superior se muestra en la Figura 13, la cual, basado en las definiciones anteriores, tiene dimensiones de 50mm x 25mm y una altura de 14mm definida por un par de pestañas que permiten la conexión con los eslabones. Esta altura se determinó en función del diámetro de los pines que se usarán como juntas para base-eslabón y eslabón-eslabón, el cual se escogió empíricamente de 3mm, lo cual provee un espacio libre entre la base y el pin de 4mm para que el eslabón gire. Además, esta base cuenta con un agujero en el centro de 3mm de diámetro con el fin de permitir el paso del cable que permitirá la compresión del mecanismo.

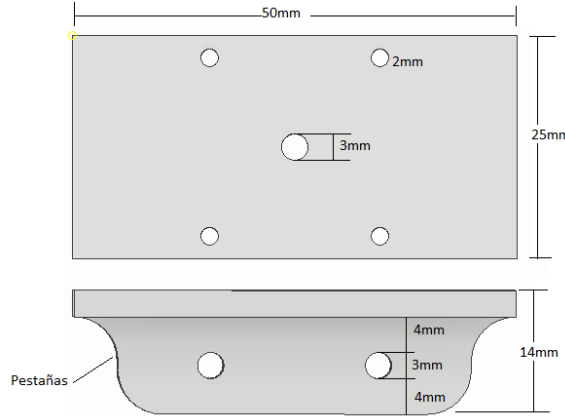


Figura 13: Base superior.

Adicionalmente, esta base requirió otros dos pequeños elementos que permitieran la disposición deseada de los motores. Estos elementos se muestran en la Figura 14. Tienen agujeros para la unión a la base superior por medio de tornillos M2 [27], así como para el ensamble de los motores con dichos elementos. Las medidas de estos componentes se detallan en los anexos. Cabe mencionar que los demás componentes, microcontrolador, módulo MPU6050, driver y la batería, fueron dispuestos en la base superior de forma que no exista cambio en el ángulo de roll; si bien existe un pequeño desbalance, el torque que se genera no provoca cambio considerable en dicho ángulo.

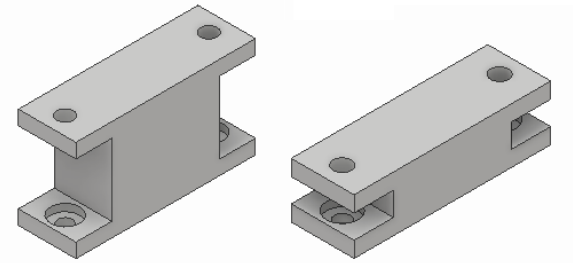


Figura 14: Elementos para disposición de los motores.

El diseño de la base inferior, Figura 15 fue muy similar al de la base superior, cuenta con las pestañas para la conexión con los eslabones y las mismas dimensiones. La diferencia radica en que esta base cuenta con un ángulo respecto a la horizontal. Este ángulo es de 15° para asegurar una orientación de 75° del robot con respecto a la horizontal. La selección de este ángulo nuevamente radica en [4], cuyos resultados mostraron que este es el ángulo que ofrece el mayor desplazamiento vertical en el movimiento parabólico del sistema.

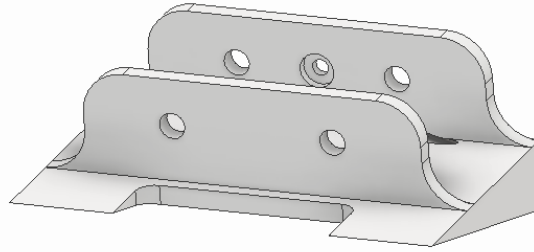


Figura 15: Base inferior.

Además, unidos a la parte trasera de las bases superior e inferior se colocaron geometrías que ayudarán al robot a mantener una buena posición horizontal, análogo a como lo harían una patas. Esta geometría se muestra en la Figura 16 y se unirá por medio de un tornillo, por lo que cuenta con pequeño agujero.

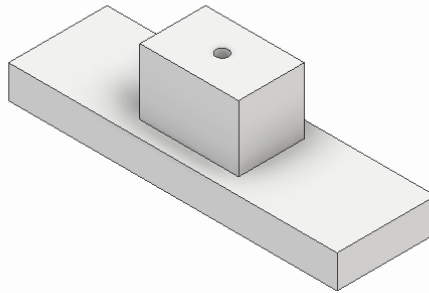


Figura 16: Patas del robot.

Para el diseño de los eslabones se tomaron en cuenta consideraciones como el espacio disponible para los resortes, los cuales tienen un diámetro exterior de 12mm, la conexión eslabón inferior - eslabón superior, la conexión base - eslabón y el espacio libre que existe entre la base y el pin, de forma que el eslabón pueda girar sin interferencia. Por lo anterior, el ancho de los eslabones se escogió de 9mm, la profundidad de 11mm, la longitud efectiva del eslabón ya se estableció de 20mm y su longitud total es de 29mm. En la Figura 17 se muestra el diseño de los eslabones superiores, y en la Figura 18 el de los eslabones inferiores.

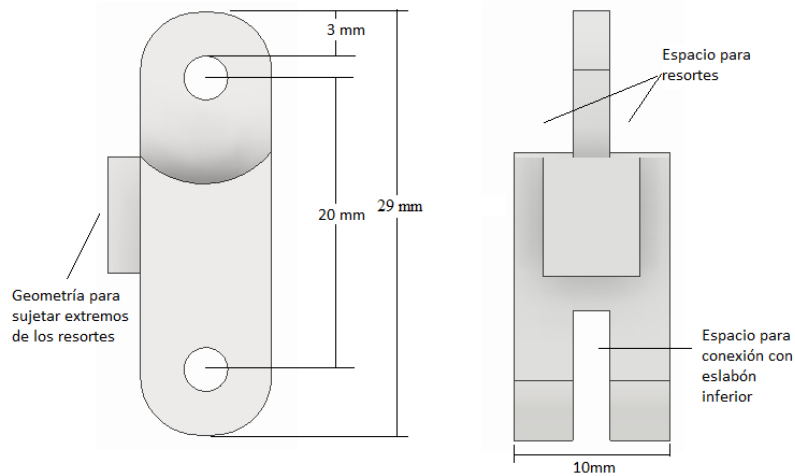


Figura 17: Eslabón superior.

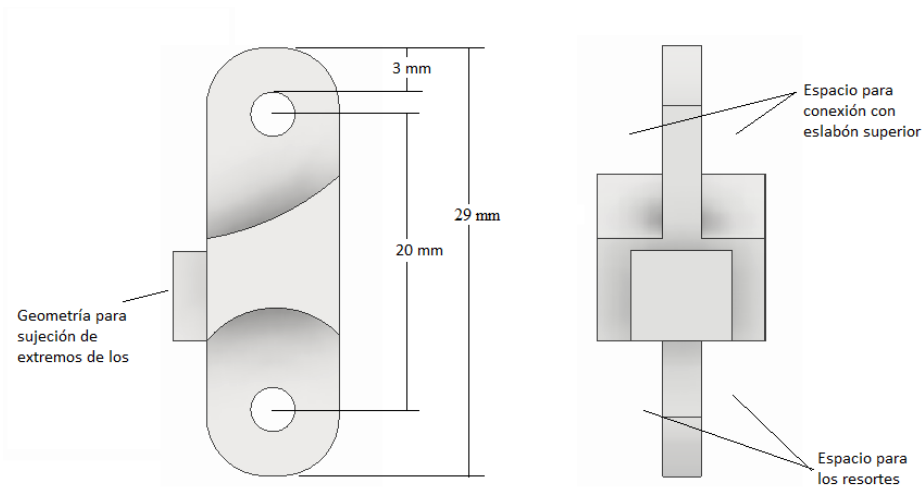


Figura 18: Eslabón inferior.

Finalmente, la barra de inercia se diseñó combinando una geometría cilíndrica con una esférica de cierto peso en uno de sus extremos. Para sujetar la barra al mecanismo se diseñó el elemento mostrado en la Figura 19, el cual se une a la barra introduciendo esta a presión para posteriormente ensamblar al conjunto al eje del motor. La longitud de la barra y el peso de la pieza esférica se iteraron durante varias pruebas para hallar la mejor combinación que ofreciera mejores resultados de control; se halló que estos valores fueron de 85mm y un peso de 10g. El diámetro de la geometría cilíndrica se definió de 3mm. En la Figura 20 se muestra el sujetador y la barra ensamblados. El motor utilizado para actuar la cola también fue el RB-Pol-452 ya que cuenta con la mayor capacidad de torque de los 4 motores.

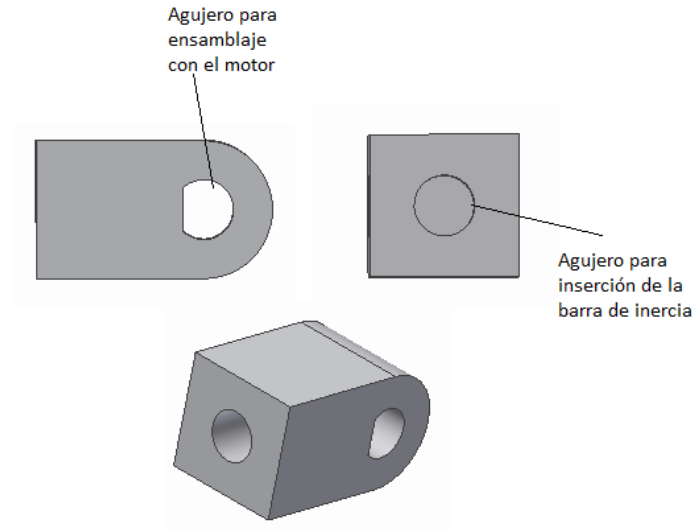


Figura 19: Elemento de sujeción para ensamblaje de la barra de inercia.

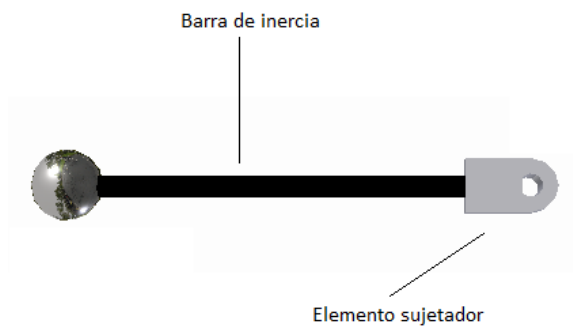


Figura 20: Barra de inercia con elemento sujetador.

El diseño CAD de los motores se obtuvo de 3D Content Central [20] al igual que el diseño del módulo MPU5060, mientras que el diseño del driver DRV8833 se obtuvo de la página del distribuidor Pololu. El microcontrolador y la batería se diseñaron como simples bloques de 10mmx10mmx1.6mm y de 25mmx18mmx6.5mm, respectivamente.

El resultado de todos estos elementos ensamblados se muestra en la Figura 21, mientras que los planos de todos los elementos se muestran en los anexos.

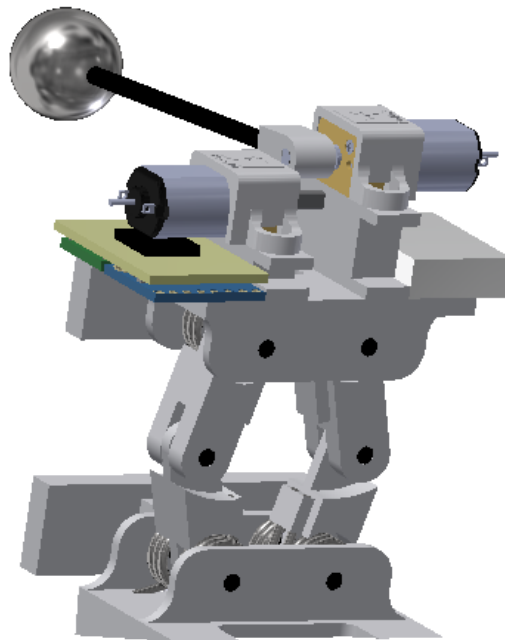


Figura 21: Diseño del robot.

Finalmente, para comprobar que el mecanismo desarrollado soporte la fuerza que experimentará y los esfuerzos provocados por ésta, se realizó un análisis estático de esfuerzos en Autodesk Inventor, de lo que se obtiene la Figura 22. En ésta puede observarse que el máximo esfuerzo es de 70.94 MPa y lo sufren los resortes, los cuales están fabricados de acero al carbono cuya resistencia a la deflexión es de 370 MPa, por lo que se comprueba que los esfuerzos no son lo suficiente grandes para deformar plásticamente o fracturar el mecanismo.

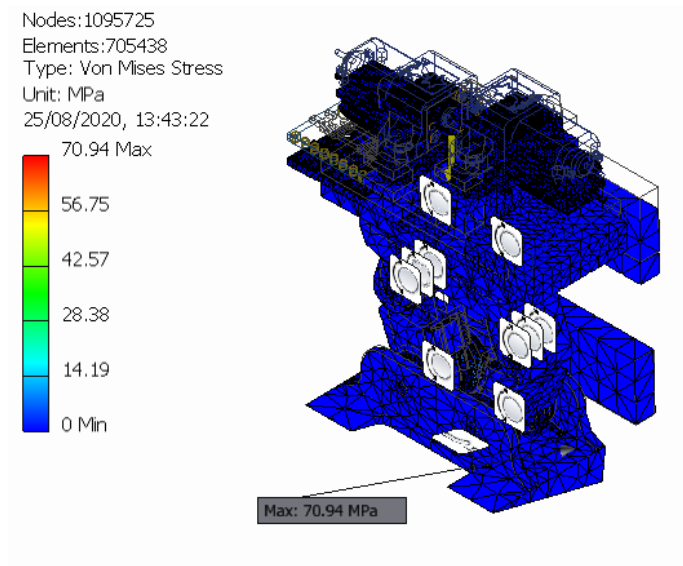


Figura 22: Análisis estático de esfuerzos.

8.8.1. Diseño final

En las Figuras 23 y 24 se muestra el diseño final del robot, el cual se diferencia del mostrado en la Figura 21 en el diseño de las patas. El motivo por el cual se realizó este cambio se explica en el capítulo de control.

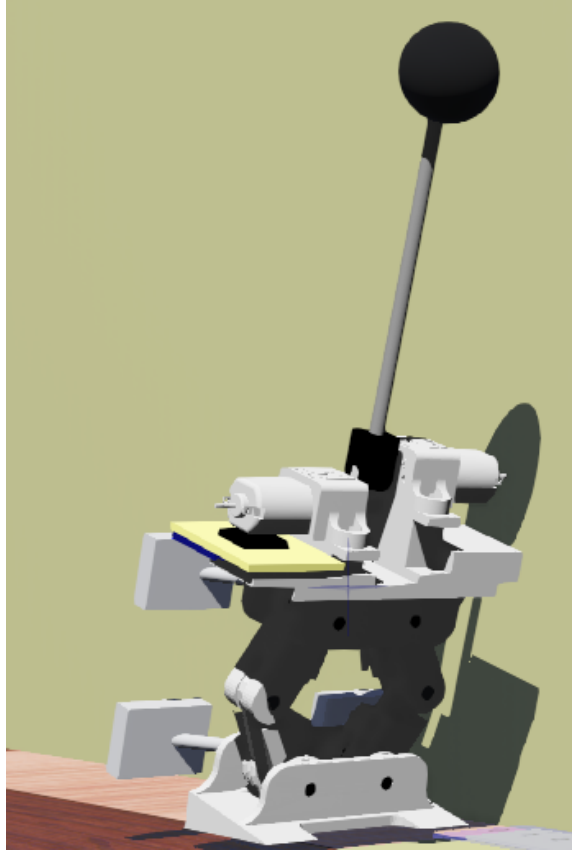


Figura 23: Diseño final del robot vista 1.

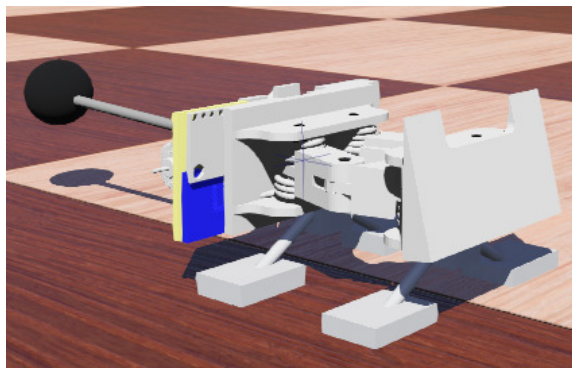


Figura 24: Diseño final del robot vista 2.

Modelado del sistema

Para llevar a cabo el control sobre el sistema primeramente se requiere conocer cómo se comporta el mismo, es decir, se debe conocer su dinámica. Sin embargo, cuando la deducción de las ecuaciones que gobiernan dicha dinámica se torna sumamente compleja, existen métodos heurísticos que permiten obtener información suficiente para plantear una metodología de control.

9.1. Modelo inicial

Como primer acercamiento para comprender la dinámica del sistema en estudio, se consideró un modelo inicial sencillo que constaba de una geometría cúbica de masa m y dimensiones $a \times b \times c$ metros, a la cual se le aplica una fuerza \vec{F} a \vec{r} metros de su centro de masa (COM) de modo que el cuerpo experimenta un giro alrededor de su eje z a la vez que viaja en un movimiento parabólico, siempre y cuando la fuerza \vec{F} tenga una componente horizontal. Puesto que se supone que el cuerpo solamente experimentará cambio de orientación alrededor de su eje z , solamente la inercia alrededor de este eje es de interés, la cual por su geometría es $I_{zz} = \frac{1}{12}m(a^2 + b^2)$. Esta situación se ilustra en la Figura 25. Cabe mencionar que este modelo, al ser un primer acercamiento para adquirir intuición acerca del comportamiento del sistema, se trata de un sistema híbrido que está compuesto por dos comportamientos: la dinámica del cuerpo rígido en una trayectoria que depende de la fuerza aplicada y el lugar de aplicación de la misma, y un comportamiento que consiste en añadir restricciones a dicha dinámica, más en específico, a la componente vertical de la posición y velocidad del cuerpo rígido y su velocidad angular. Estas restricciones tienen lugar cuando se cumplen ciertas condiciones, las cuales se especificarán más adelante, y su objetivo principal es evitar que el cuerpo rígido experimente caída libre hasta el infinito.

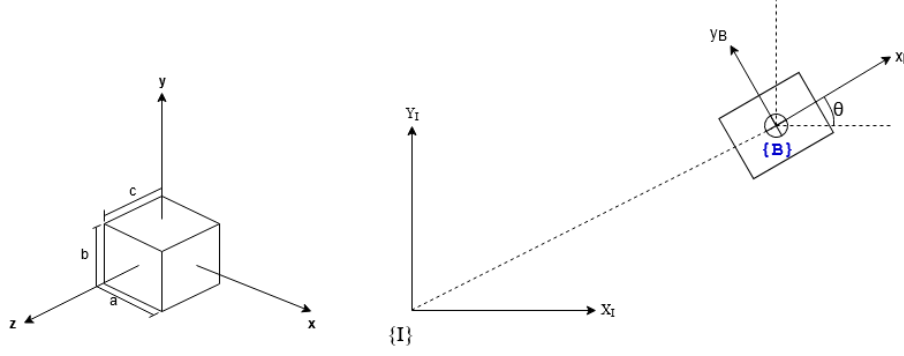


Figura 25: Modelo inicial.

El cuerpo B tiene una posición $[x, y]$ y una rotación θ con respecto al marco inercial I como se muestra en la figura. Estas son las variables de interés del sistema ya que se desea conocer cómo evoluciona la posición y orientación del sistema luego de que la fuerza le ha sido aplicada. Se forma entonces el vector de coordenadas generalizadas respecto al marco inercial como sigue

$${}^I \mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (34)$$

Empleando ahora las ecuaciones de Newton - Euler para el modelado se tiene lo siguiente

$$\sum F_{x_I} = m\ddot{q}_1 = f_{x_I} \quad (35)$$

$$\sum F_{y_I} = m\ddot{q}_2 = f_{y_I} - mg \quad (36)$$

$$\sum \tau_z = I_{zz}\ddot{q}_3 \quad (37)$$

Despejando para las variables de interés se tiene que

$$\ddot{q}_1 = \frac{1}{m} f_{x_I} \quad (38)$$

$$\ddot{q}_2 = -g + \frac{1}{m} f_{y_I} \quad (39)$$

$$\ddot{q}_3 = \frac{1}{I_{zz}} \tau_z \quad (40)$$

En donde f_{x_I} y f_{y_I} son las componentes horizontal y vertical de la fuerza \vec{F}_I , respectivamente, respecto del marco de referencia inercial y τ_z es el torque alrededor del eje z que genera el aplicar dicha fuerza a una distancia $\vec{r} = [r_H \ r_V]^T$ del centro de masa del cuerpo rígido, donde r_H y r_V son las componentes horizontal y vertical del vector de posición \vec{r} , respectivamente.

La fuerza que sufre el cuerpo rígido está referenciada con respecto al cuerpo rígido y se denota $\vec{B}u = [u_1 \ u_2]^T$ y dado que la fuerza se requiere respecto del marco inercial se realiza la siguiente transformación

$$\begin{bmatrix} f_{xI} \\ f_{yI} \end{bmatrix} = R_B^I(q_3)^B u \quad (41)$$

Donde R_B^I es la matriz de rotación del marco de referencia B con respecto al marco de referencia I y está dada por

$$R_B^I = \begin{bmatrix} \cos(q_3) & -\sin(q_3) \\ -\sin(q_3) & \cos(q_3) \end{bmatrix} \quad (42)$$

de lo que se tiene que

$$\begin{bmatrix} f_{xI} \\ f_{yI} \end{bmatrix} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) \\ -\sin(q_3) & \cos(q_3) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (43)$$

El torque τ_z es hallado multiplicando la distancia por la fuerza como sigue

$$\tau_z = r_H u_2 - r_V u_1 = [-r_V \ r_H] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (44)$$

Colocando (28)-(30) en forma matricial se tiene

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} \cos(q_3) & -\sin(q_3) \\ -\sin(q_3) & \cos(q_3) \\ -r_V & r_H \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (45)$$

Si se define

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \quad (46)$$

Entonces

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \quad (47)$$

Y el sistema dinámico no lineal es

$$\dot{x} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} & & & 0 & 0 \\ & & & 0 & 0 \\ & & & 0 & 0 \\ \frac{1}{m} & 0 & 0 & \cos(q_3) & -\sin(q_3) \\ 0 & \frac{1}{m} & 0 & -\sin(q_3) & \cos(q_3) \\ 0 & 0 & \frac{1}{I_{zz}} & -r_V & r_H \end{bmatrix} u \quad (48)$$

El cual está sujeto a la restricción de que si $x_2 = 0$ y $x_5 < 0$, entonces

$$\begin{aligned} x_2 &= 0 \\ x_4 &= x_4 - b_x x_4 \\ x_5 &= 0 \\ x_6 &= 0 \end{aligned}$$

donde b_x es el coeficiente de fricción lineal en dirección del movimiento horizontal.

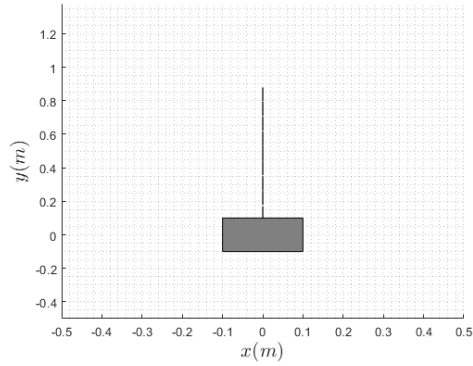
Dicha restricción provoca que el modelo represente a una colisión inelástica perfecta dado que el cuerpo rígido, al entrar en contacto con el suelo, se desplaza en la dirección del movimiento horizontal y no rebota, lo cual no es lo que sucede en la realidad. Sin embargo, a pesar de esto, el modelo permite adquirir intuición del comportamiento del sistema al aplicársele una fuerza y de cómo éste evoluciona mientras se encuentra en el aire, lo cual es el aspecto más relevante para los objetivos de la investigación.

Con el modelo derivado se procedió a realizar simulaciones del mismo en el entorno MATLAB, de lo cual se obtienen los gráficos mostrados en la Figura 26. En la Figura 26a se observa la trayectoria que toma el cuerpo rígido al aplicársele al mismo una fuerza completamente vertical, con lo que el cuerpo solamente se eleva hasta una altura máxima y posteriormente cae en caída libre. En la Figura 26b se muestran las variables de estado de dicha trayectoria en función del tiempo, se observa cambio solamente en la posición y velocidad vertical como se esperaba.

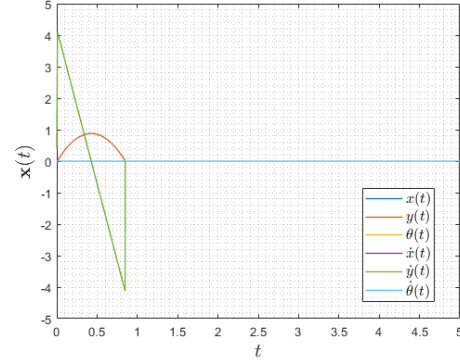
En la Figura 26c se muestra la trayectoria del cuerpo cuando este es sometido a una fuerza completamente horizontal, con la que el cuerpo simplemente experimenta cambio en su coordenada x tal y como lo muestra la Figura 26d, en la que se observan los cambios en función del tiempo de la posición y velocidad en la coordenada horizontal.

Finalmente, en la Figura 26e se observa la trayectoria seguida por el cuerpo cuando se aplica una fuerza con componente vertical y horizontal. Se observa que su trayectoria corresponde a un tiro parabólico tal y como se esperaba y como se muestra en la Figura 26f en sus variables de estado, en la que se observa que la velocidad horizontal es constante y la velocidad vertical varía linealmente debido a la acción de la aceleración de la gravedad.

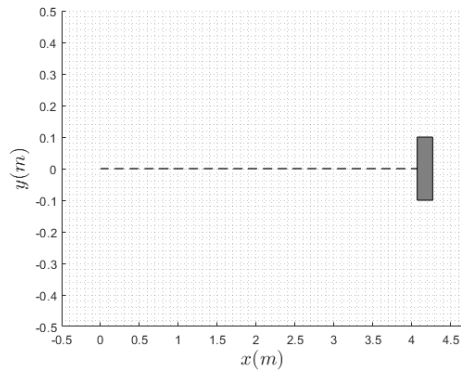
Cabe señalar que todas estas pruebas se realizaron aplicando la fuerza en el centro de masa del cuerpo rígido, por lo que no existe cambio en la posición y velocidad angular del mismo. Dicha prueba aplicando la fuerza a una distancia $\vec{r} = [-0.01 \ 0]^T$ del centro de masa se muestra en la Figura 27. La Figura 27a muestra la trayectoria seguida y la Figura 27b muestra el cambio tanto en las posiciones x y y como en su posición angular θ , así como las velocidades que se experimentan



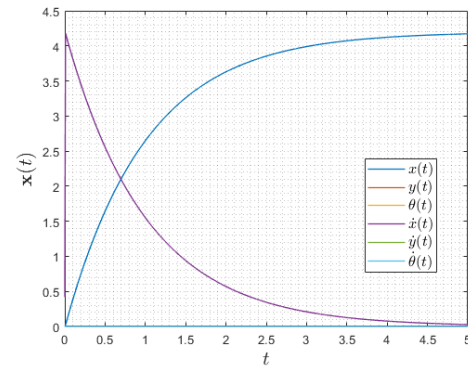
(a) Trayectoria del cuerpo rígido aplicando una fuerza completamente vertical.



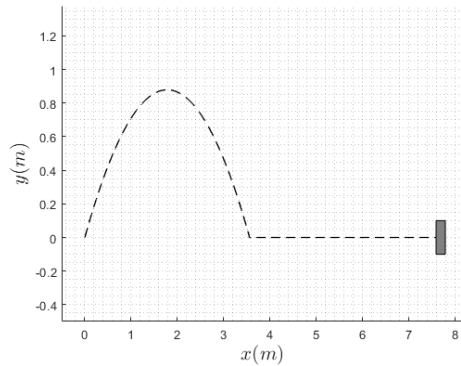
(b) Variables de estado del sistema aplicando una fuerza completamente vertical.



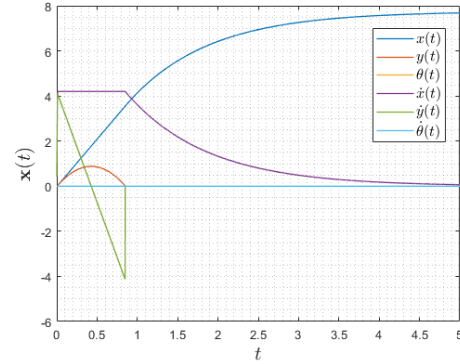
(c) Trayectoria del cuerpo rígido aplicando una fuerza completamente horizontal.



(d) Variables de estado del sistema aplicando una fuerza completamente horizontal.

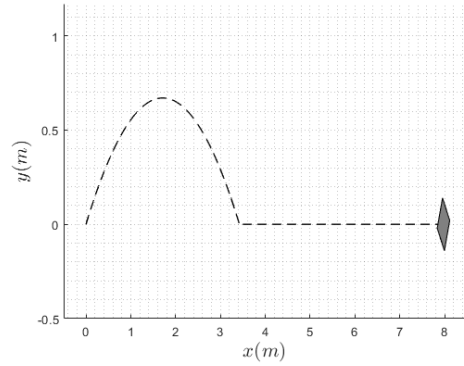


(e) Trayectoria del cuerpo rígido aplicando una fuerza con componentes horizontal y vertical.

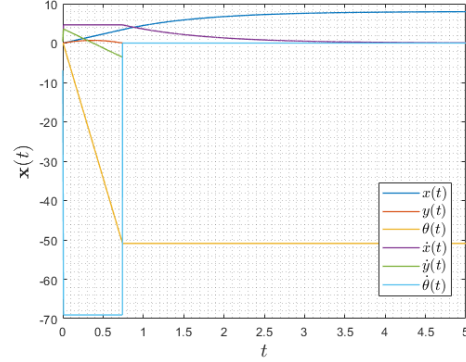


(f) Variables de estado del sistema aplicando una fuerza con componentes horizontal y vertical.

Figura 26: Simulación del sistema dinámico.



(a) Trayectoria del cuerpo rígido aplicando una fuerza con componentes horizontal y vertical a una distancia r del centro de masa.



(b) Variables de estado del sistema aplicando una fuerza con componentes horizontal y vertical a una distancia r del centro de masa.

Figura 27: Simulación del sistema dinámico.

Con este modelo fue posible obtener resultados que validaron la fase de diseño. Para ello primeramente se calcula la velocidad a la que el sistema se moverá una vez se libere la energía almacenada en los resortes torsionales. Este cálculo puede ser llevado a cabo con el concepto de conservación de la energía mecánica como sigue

$$\Delta E_{mec} = \Delta K + \Delta U = 0 \quad (49)$$

$$\frac{1}{2}m\Delta \vec{v}^2 + mg\Delta h + \frac{1}{2}k\Delta\theta^2 = 0 \quad (50)$$

Tomando como puntos de referencia el instante (1) en el que toda la energía del sistema corresponde a la almacenada por los resortes se tiene que

$$E_{instante1} = \frac{1}{2}k\Delta\theta^2 \quad (51)$$

en la fase de diseño se halló que la magnitud de esta energía es de 1.13J y que la masa del sistema es 0.134kg. En el instante en el que la energía de los resortes se libere la energía potencial se transforma en energía cinética, por lo que el otro punto de referencia será dicho momento, de lo que se tiene que

$$E_{instante2} = \frac{1}{2}m\Delta \vec{v}^2 \quad (52)$$

$$E_{instante1} = E_{instante2}$$

$$\frac{1}{2}m\Delta \vec{v}^2 = 1.124J \quad (53)$$

dado que cuando el robot está en reposo su velocidad es 0, $v_o = 0m/s$ de lo que se tiene que la velocidad final v_f es

$$v_f = \sqrt{\frac{2 * 1.124J}{0.119kg}} = 4.339m/s \quad (54)$$

Como segundo paso para la validación de los cálculos de diseño, se tiene el cálculo de la fuerza resultante que actúa sobre el sistema, el cual se puede realizar a través del análisis de la cantidad de movimiento lineal \vec{p} como se muestra a continuación.

Se sabe que la razón de cambio con el tiempo de la cantidad de movimiento lineal del objeto en estudio es igual a la fuerza resultante que éste experimenta

$$\sum \vec{F} = \frac{d\vec{p}}{dt} \quad (55)$$

Tomando como referencia los mismos dos instantes utilizados para el cálculo de la velocidad se tiene que

$$\sum \vec{F} = \frac{\vec{p}_{instante2} - \vec{p}_{instante1}}{dt} \quad (56)$$

$$\vec{p}_{instante1} = m\vec{v}_o \quad (57)$$

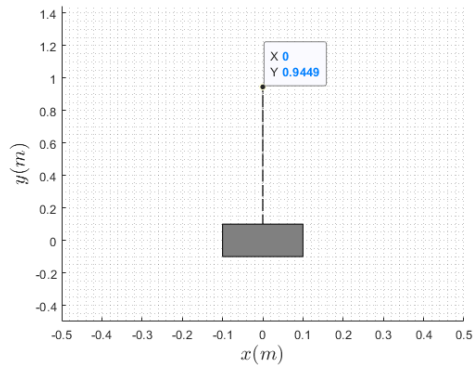
$$\vec{p}_{instante2} = m\vec{v}_f \quad (58)$$

Dado que la velocidad inicial v_o es igual a 0, el cambio en la cantidad de movimiento lineal es igual a $\vec{p}_{instante2}$, de lo que, suponiendo que la fuerza es aplicada al sistema durante 10ms, se puede hallar la fuerza resultante como

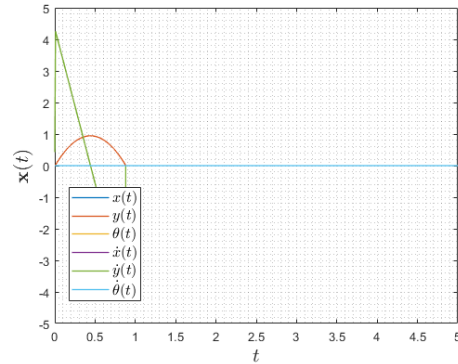
$$\vec{F} = \frac{\vec{p}_{instante2}}{dt} = \frac{(0.119kg)(4.339m/s)}{0.010s} = 51.826N \quad (59)$$

Ahora es posible introducir esta fuerza a la simulación del modelo derivado en MATLAB y observar si la altura máxima alcanzada por el sistema concuerda con la calculada en la fase de diseño. Sin embargo, antes de llevarlo a cabo es importante tomar en cuenta que esta fuerza es aplicada en el centro de masa del sistema dado que el cálculo por medio de la cantidad de movimiento lineal asume que se trata de una partícula. Además, debido a la configuración elegida para el robot, que hace que éste tenga un ángulo de 75° con respecto a la horizontal, este experimentará un movimiento parabólico. Sin embargo, la fuerza hallada en los calculos anteriores no toma en cuenta dicha consideración, por lo que el robot alcanzará la altura estipulada solamente si la fuerza es aplicada de forma completamente vertical. Este no es el caso en el modelo real, por lo que la altura que alcanzará el robot será inferior, dado que la fuerza será separada en componentes horizontal y vertical. Con estas aclaraciones

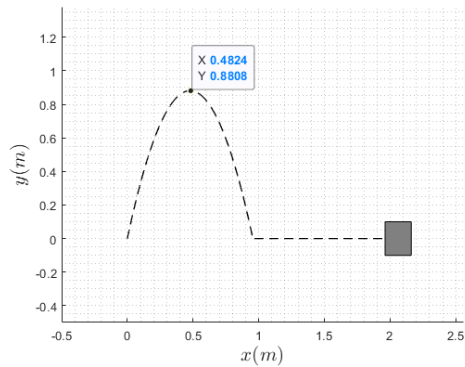
se procedió a realizar la simulación de la cual se obtuvieron los resultados mostrados en la Figura 28. En las Figuras 28a y 28b se observa que el cuerpo rígido alcanza la altura esperada, con un error de 1.70%, dado que la fuerza se aplicó verticalmente. En las Figuras 28c y 28d se observa que la altura alcanzada es menor que la esperada debido a que la fuerza se aplicó como en realidad la experimentará el robot, por lo que se sabe ahora que la altura máxima que el robot alcanzará es de aproximadamente 0.88 m y la razón de salto será de 11, con lo que se cumple con el requerimiento del salto.



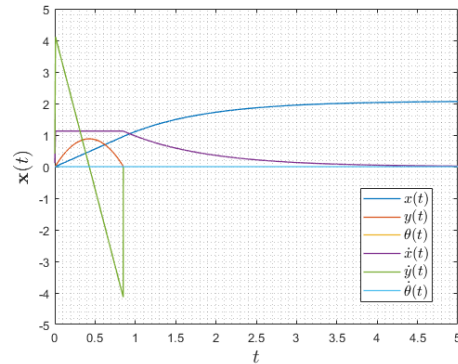
(a) Trayectoria del cuerpo rígido aplicando la fuerza completamente vertical.



(b) Variables de estado del sistema aplicando la fuerza completamente vertical.



(c) Trayectoria del cuerpo rígido aplicando la fuerza con un ángulo de 75° con respecto a la horizontal.



(d) Variables de estado del sistema aplicando la fuerza con un ángulo de 75° con respecto a la horizontal.

Figura 28: Simulación del sistema dinámico aplicando la fuerza que experimentará el sistema real.

9.2. De MATLAB a Webots

En la sección anterior se describió el modelo inicial con el que fue posible validar el diseño mecánico. Sin embargo, este sistema no incluye la barra de inercia con la que fue posible controlar la orientación del robot, por lo que no era un modelo que describiera la dinámica del sistema de interés. Por ello fue necesario obtener otro sistema que describiera el comportamiento del robot incluyendo la barra de inercia. La obtención de este nuevo modelo se tornó compleja de forma analítica, por lo que se decidió obtener una aproximación del mismo por medio de métodos heurísticos, como ya se había previsto.

Lo anterior se llevó a cabo por medio del simulador Webots para lo cual fue necesario, como primer paso, definir o construir el robot en dicho entorno. Dicho procedimiento se detalla en la siguiente sección. Una vez con el robot definido la primera prueba realizada en el entorno Webots fue corroborar los resultados obtenidos de MATLAB con el modelo inicial. La forma de realizarlo fue aplicando al robot la fuerza hallada, observando la altura que éste alcanzaba y comparándola con la altura que alcanzó en MATLAB. La forma de aplicar la fuerza que simule el salto en Webots se detalla en la siguiente sección. Se halló que los resultados (ver Figura 29) mostraban cierta diferencia, alcanzando en Webots la altura de 0.694 m contra los 0.88 m que alcanzó el modelo de MATLAB. Sin embargo, se halló que la diferencia fue debido a la forma física del robot, ya que mientras en MATLAB el sistema se aproximó como un cubo, en Webots el robot ya contaba con la forma compleja real, lo cual, dada la presencia de la barra de inercia, provocó que el centro de masa del robot cambiara durante el salto por lo que se generó un pequeño giro en el sistema que restó altura al salto. Sin embargo, como se explicará en el capítulo de Control, se halló que al implementar el control al robot, su centro de masa permanece más estable y se alcanza la altura estimada.

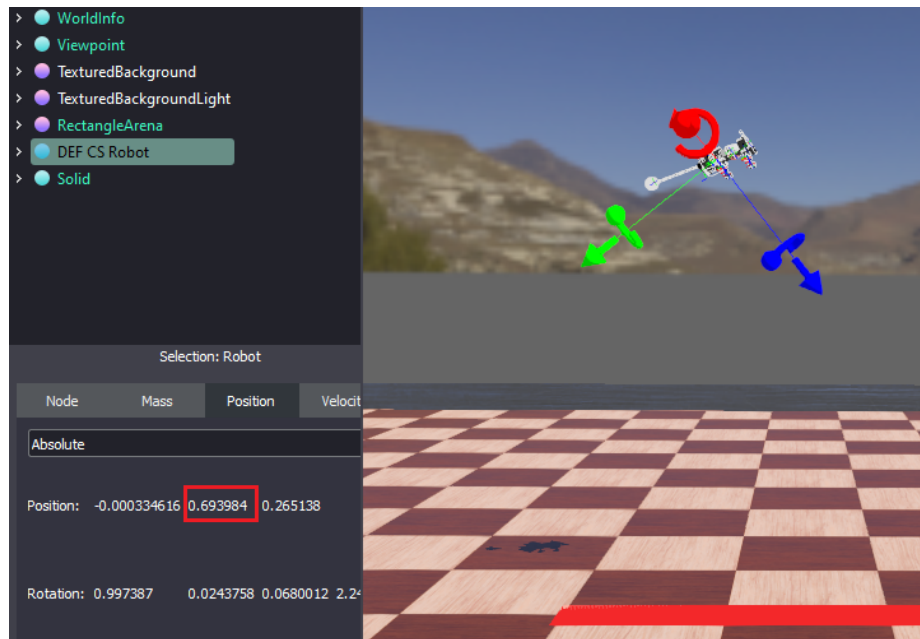


Figura 29: Comprobación de resultados de MATLAB en Webots.

Una diferencia notable, y que vale la pena mencionar, entre la simulación de MATLAB y la de Webots se halló en el comportamiento del robot al momento de entrar en contacto con el suelo. En MATLAB, dada la definición del sistema híbrido, el robot simplemente se deslizaba en la dirección de la fuerza al impactar con el suelo, mientras que en Webots, dado que cuenta con un motor de física, éste rebotaba tal y como se espera que suceda en la realidad. Aunque este comportamiento no fue necesario tomarlo en cuenta para el modelado y control del robot, si fue de gran importancia para asegurar el buen aterrizaje del mismo, ya que gracias a esto fue posible notar que el primer diseño mecánico no permitiría un correcto aterrizaje. Este problema en el aterrizaje así como la forma de solucionarlo se ampliará en el capítulo de Control.

La selección del simulador tanto para la obtención del modelo como para la validación del control desarrollado, se basó en primera instancia en los simuladores utilizados por el autor. Sin embargo, al notar la importancia que en este caso representaba una buena simulación de contactos, se decidió investigar si Webots era, si no la mejor opción, al menos una que entregara resultados suficientemente buenos. Tras esta investigación se halló en [28] una comparación entre distintos motores de física, entre los que se encontraba el utilizado por Webots (Open Dynamics Engine (ODE:)) y en la que se concluye que, si bien ODE: entregó de los mejores resultados en las pruebas realizadas, todos los simuladores cuentan con ventajas y desventajas unos sobre otros dependiendo de la aplicación. Con esto, y tomando en cuenta el buen resultado obtenido para los contactos internos definidos en la construcción del robot (descrita en la sección siguiente), se concluye que Webots es un simulador que entrega resultados fiables, con la ventaja que es un entorno con el que el autor está familiarizado y que es de código abierto.

9.3. Modelo de Autodesk Inventor en Webots

Posteriormente a realizar la validación de los cálculos de diseño a través del modelo inicial, se procedió a importar al diseño al entorno de simulación Webots. Para ello se realiza el procedimiento de migración del diseño explicado en la sección 7.2. Dicho procedimiento se resume en la Figura 30.

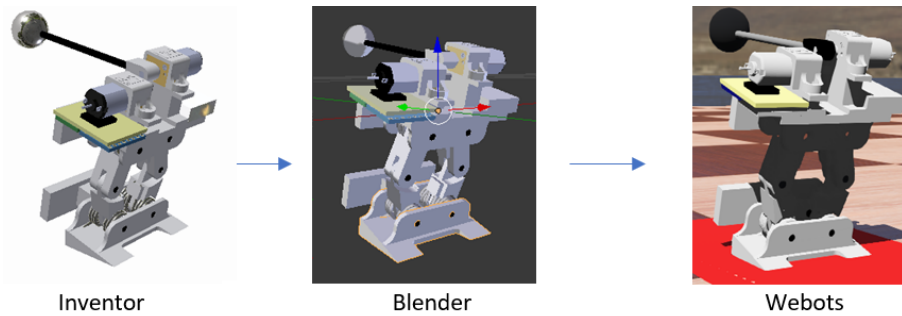


Figura 30: Proceso de migración del diseño de Inventor a Blender y a Webots.

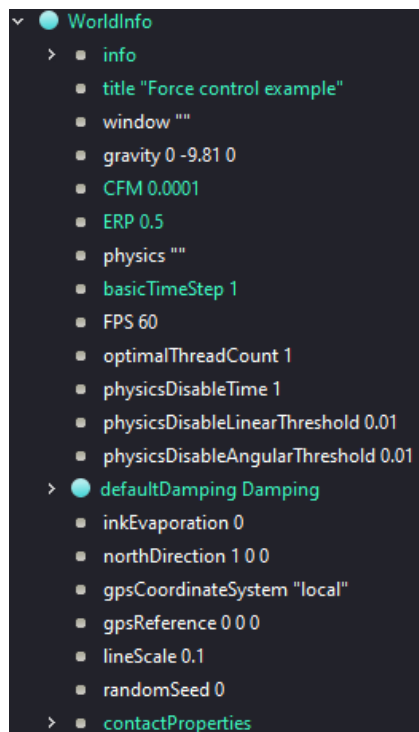
A continuación, se describe el proceso llevado a cabo para incorporar el diseño del robot a Webots de forma que fuese posible realizar pruebas del comportamiento del mismo.

9.3.1. Configuración del entorno

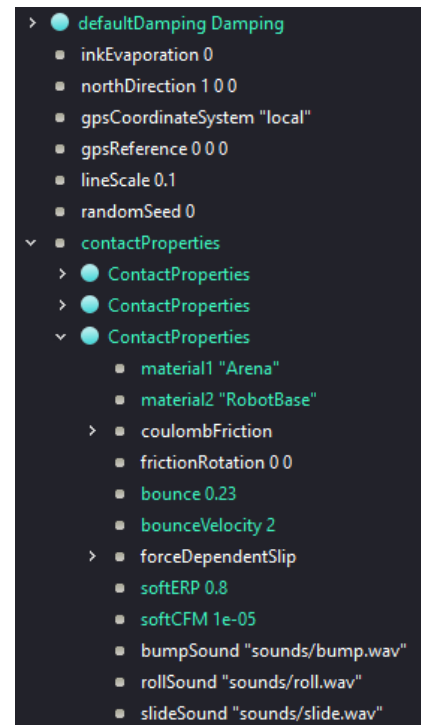
Para definir el entorno de simulación en Webots se modificó el nodo WorldInfo del mismo accediendo a los campos que este posee, tales como el basicTimeStep, ERP, northDirection, damping y el ContactProperties. Cada uno de estos campos fueron configurados con los siguientes valores

- `basicTimeStep = 1`, para utilizar un período de muestreo de 10 ms.
- `ERP = 0.8`, para asegurar que el comportamiento de la junta del robot sea lo más realista posible sin representar errores para Webots, ya que valores más altos de ERP pueden provocarlos.
- `CFM = 1e-05`, de forma que el contacto entre el robot y el suelo no sea tan “esponjoso”.
- `northDirection = 1 0 0`, con lo que la dirección del norte apunta en dirección del eje x positivo.
- `damping = 0.2` tanto para el amortiguamiento lineal como el angular, de forma que la simulación sea más realista permitiendo que el objeto no se deslice o gire sin desacelerar como si no existiese fricción con el suelo o con el aire.
- `ContactProperties`, en el cual se modifica el campo `bouncing` y `bouncingVelocity` de forma que el comportamiento del robot al caer al suelo sea apegado a la realidad, sin un rebote pronunciado dada la característica de rigidez que posee el material PLA del que está hecho el robot.

Estas modificaciones son mostradas en la Figura 31.



(a) Configuración del nodo WorldInfo de Webots.



(b) Configuración del nodo WorldInfo de Webots (continuación).

Figura 31: Definición del entorno en Webots.

9.3.2. Definición del robot

Forma física del robot

Esta acción se llevó a cabo por medio del nodo Robot de Webots, al cual fue posible añadir, por medio del nodo Shape, la geometría diseñada en Inventor. Además, se agregó al mismo dispositivos como acelerómetro y una unidad de medición inercial IMU para realizar las mediciones pertinentes del ángulo roll; esta acción se observa en la Figura 32. Cabe aclarar que en referencias como [4] y [22] el ángulo de interés es el pitch; en este caso, dada la definición del campo *northDirection*, el ángulo de interés es el roll. Estas medidas son brindadas por la IMU como las mediciones convencionales en un plano cartesiano, es decir, el ángulo es medido respecto al eje horizontal e incrementa en sentido contrario de las agujas del reloj. Esto se ilustra en la Figura 33, en la que, dada la configuración definida en la fase de diseño del robot, se muestra efectivamente que el mismo tiene un ángulo de -15° ya que la IMU es paralela a la base superior del robot.

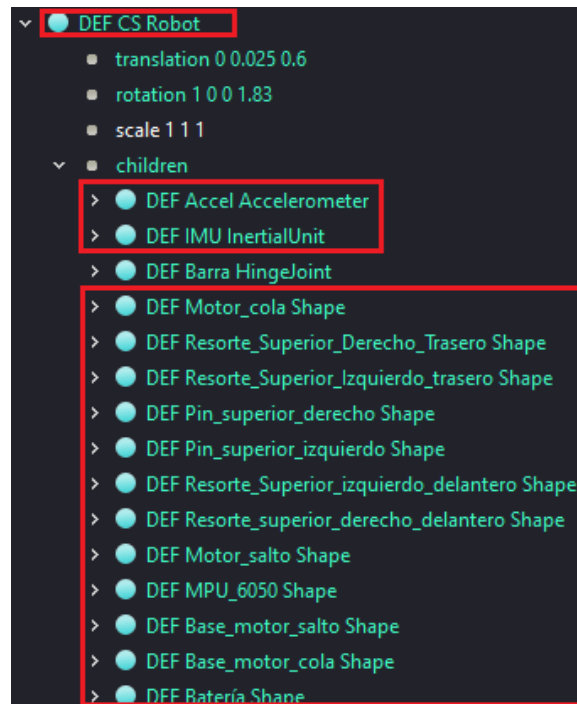


Figura 32: Forma física del robot en Webots.

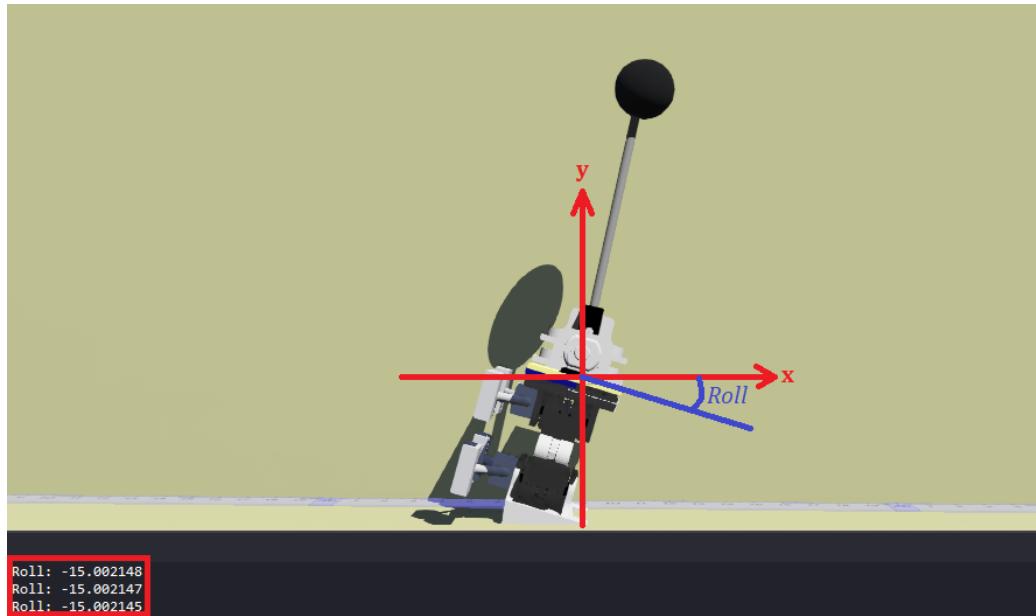


Figura 33: Mediciones del ángulo de roll en Webots.

Interacción del robot con el entorno

Otro aspecto de suma relevancia en el nodo Robot radica en la importancia de definir los campos *physics* y *boundingObject*, con los cuales se definen propiedades como la masa y la forma geométrica del robot que el entorno tomará en cuenta para la detección de colisiones, respectivamente. Si estos campos se dejan vacíos, el robot estaría estático en cualquier punto sin sufrir el efecto de la gravedad, y podría ser atravesado por otros objetos. Si solamente se define el campo *physics* el objeto experimenta caída libre hasta el infinito debido a que el simulador no posee información para la detección de colisiones que brinda el campo *boundingObject*, por lo que el robot atravesaría el suelo. En la Figura 34 se muestra la definición de estos campos. Cabe señalar que la masa definida para el robot no son los 119g ya que muchos de los elementos, como la cola, la base inferior y los eslabones se definieron dentro del robot con sus propias características físicas.

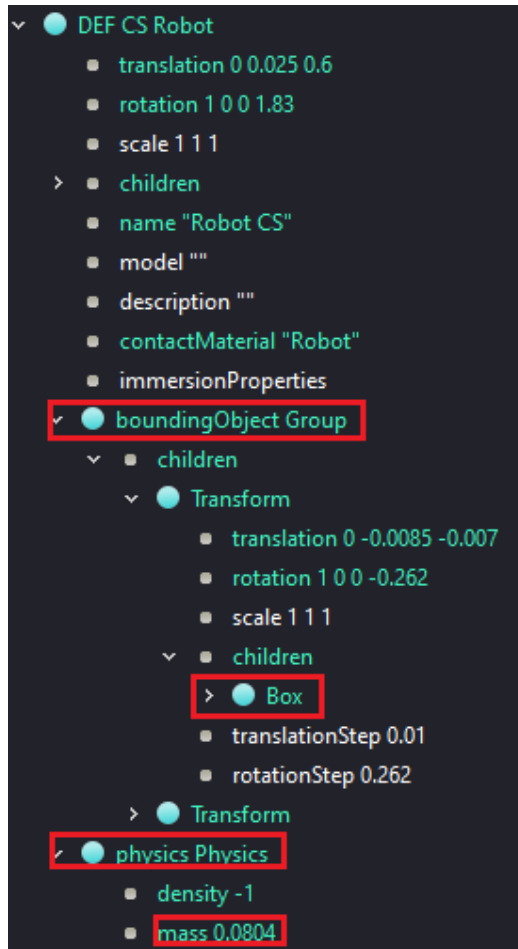


Figura 34: Definición de la interacción del robot con el entorno en Webots.

Compresión del robot

Para simular la compresión y descompresión del robot se utilizó una junta prismática o junta deslizante, Figura 35, actuada por medio de un motor lineal. Esta junta está conformada por dos elementos: la base superior, que corresponde al *parent*, y la base inferior, que corresponde al *end point*, del robot, de forma que el motor lineal pueda actuarla para juntar y separar a ambas bases, generando así la simulación de una compresión y descompresión. En la Figura 35 se muestra el parámetro *axis* que define el eje a lo largo del cual se moviliza la junta, que en este caso es un vector unitario u que está a 15° del eje vertical y , como se muestra en la Figura 36. En la Figura 37 se muestra la definición de dicha junta.

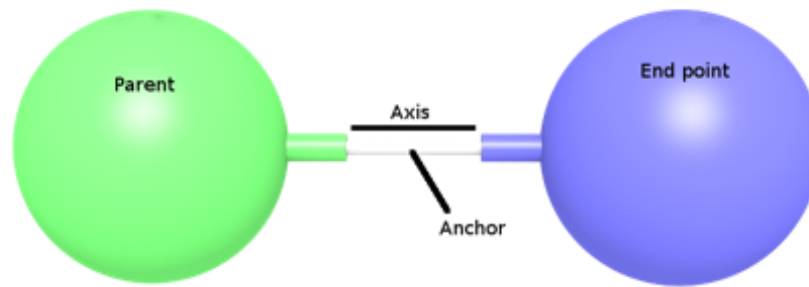


Figura 35: Junta deslizante en Webots.

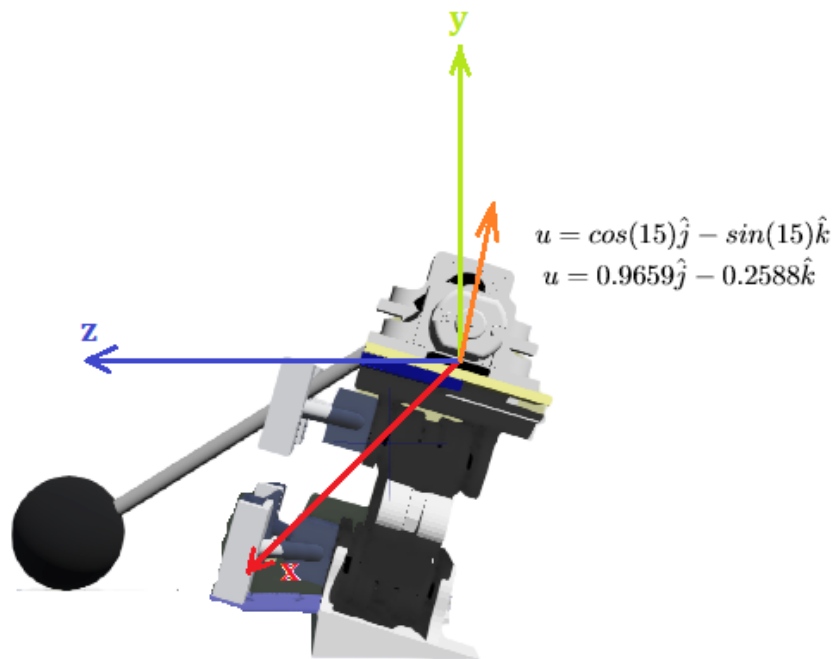


Figura 36: Eje de junta deslizante en Webots.

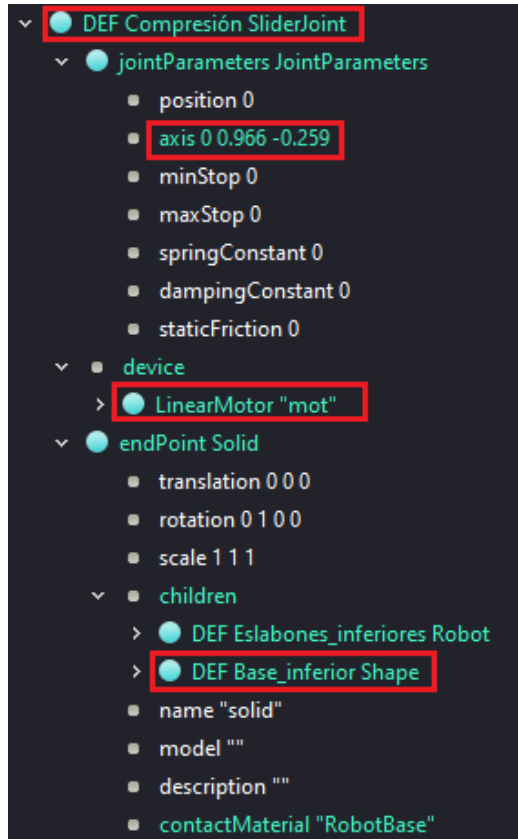


Figura 37: Definición de la simulación de compresión del Robot.

Salto del robot

Dada la complejidad para simular resortes torsionales en Webots, se emplearon las funciones especiales del *Supervisor* del nodo Robot.

En la Figura 38 se muestra una simplificación del análisis del robot sufriendo la fuerza que provoca su salto. En este análisis se considera al robot como un sistema de dos masas, la base superior m_s y la base inferior m_i , unidas por medio de un resorte que almacena la energía. Cuando el resorte se comprime se almacena la energía que lo impulsará en el aire. Esta fuerza, mientras el resorte se descomprime, es aplicada solamente en la base superior, la cual experimentará momento lineal $\vec{p} = m_s v$ el cual, cuando el resorte se expanda por completo, se transformará en $\vec{p} = (m_s + m_i)v$. Por lo anterior se considera que el punto de aplicación de la fuerza para la simulación del salto es el centro de masa del robot conformado por ambas masas, lo cual en Webots se lleva a cabo con la función `wb_supervisor_node_add_force`, la cual permite aplicar una fuerza al robot, precisamente, en su centro de masa. La magnitud y dirección de la fuerza se definen dentro de esta función y sus valores son los mismos que los hallados en el modelo inicial.

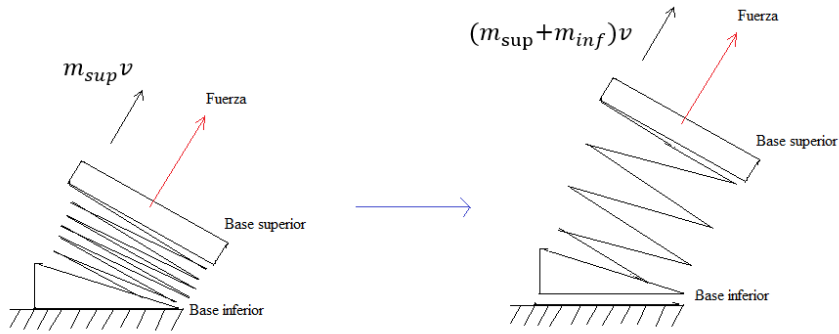


Figura 38: Análisis para aplicación de la fuerza en Webots.

Definición de la barra de inercia

Para la definición de la barra de inercia se empleó el nodo Joint de Webots, con el cual fue posible definir la unión entre el robot y la barra de inercia como una junta revoluta (HingeJoint), cuya geometría corresponde al conjunto del sujetador de la barra de inercia, la barra de inercia y el peso de 10 g. La barra es actuada por un motor cuyas características se definieron en la fase de diseño. Estas operaciones se muestran en la Figura 39.

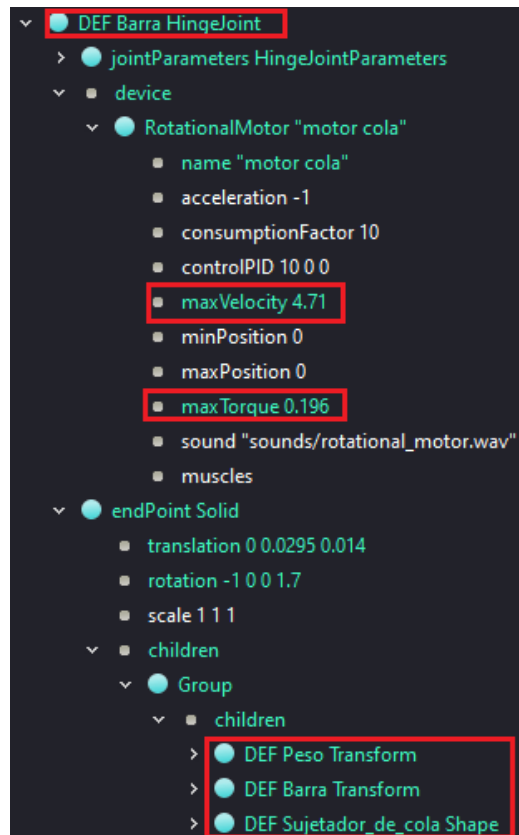


Figura 39: Definición de la barra de inercia.

9.3.3. Obtención de datos y mediciones con MATLAB

Como se mencionó en la parte de metodología, se realizaron mediciones de interés con MATLAB para obtener una aproximación del sistema y determinar qué método de control utilizar. Esto se llevó a cabo perturbando al sistema (el modelo del robot en Webots) con un escalón de entrada (torque generado por la barra de inercia) y observando las mediciones obtenidas en la variable de interés (velocidad angular del pitch:); estos experimentos y sus resultados se muestran en las Figuras 40 - 42. Es importante mencionar que se utilizó la velocidad angular dado que su magnitud tiende a estabilizarse a un valor constante, como se muestra en las figuras, mientras que la posición angular diverge. Las mediciones de la velocidad angular fueron obtenidas utilizando un giroscopio en Webots.

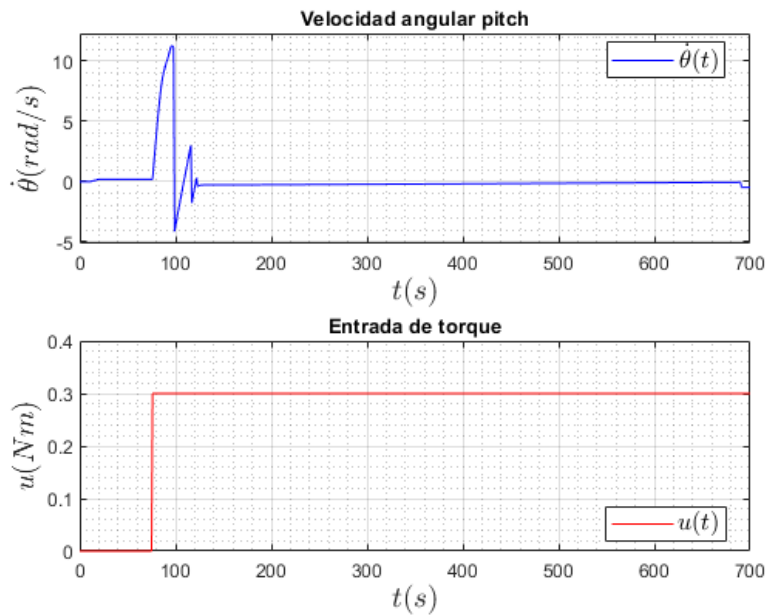


Figura 40: Respuesta de la velocidad al escalón de 0.3 Nm.

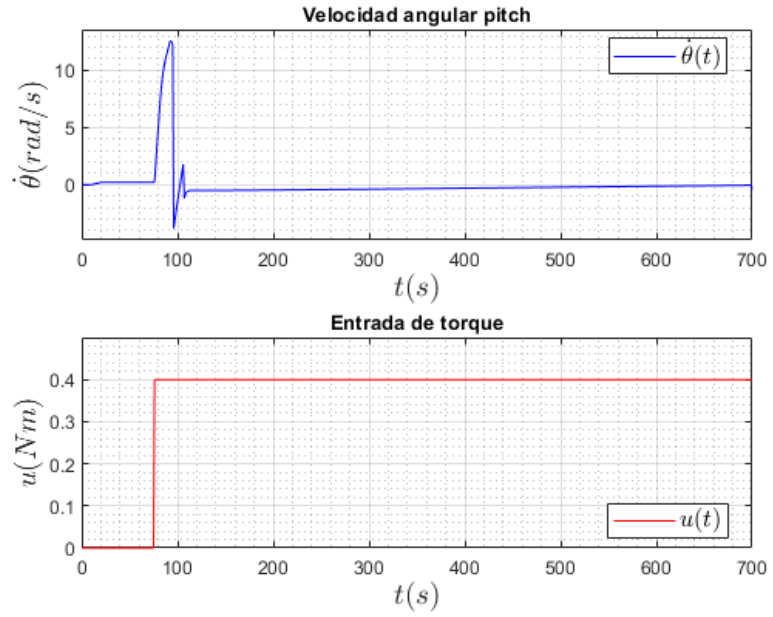


Figura 41: Respuesta de la velocidad al escalón de 0.4 Nm.

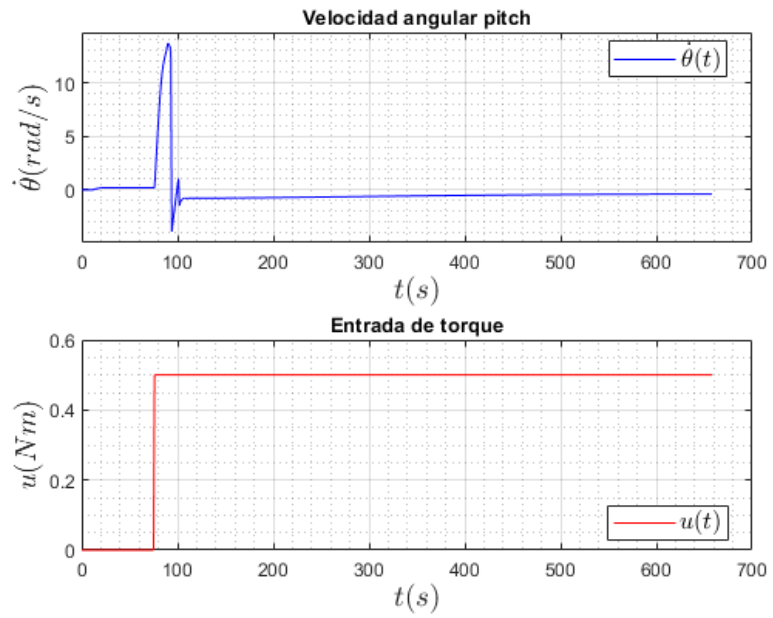


Figura 42: Respuesta de la velocidad al escalón de 0.5 Nm.

Posteriormente, con cada uno de los gráficos anteriores se obtuvo una aproximación del sistema, y esto se llevó a cabo con la *System Identification Tool* de MATLAB. Luego, dado que la variable de interés para el control es el ángulo pitch y no su tasa de cambio, se procedió a integrar al sistema para obtener la expresión del sistema al cual se desea controlar. Al aplicar un escalón unitario a estos sistemas se obtienen las Figuras 43 - 45, en las que se observa el comportamiento esperado y que se mencionó con anterioridad: la posición angular diverge.

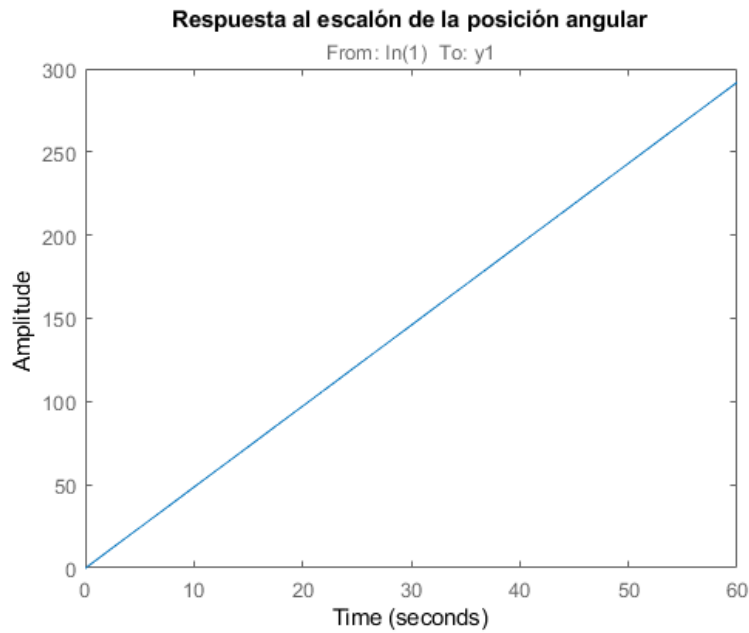


Figura 43: Respuesta del sistema de posición al escalón de 0.3 Nm.

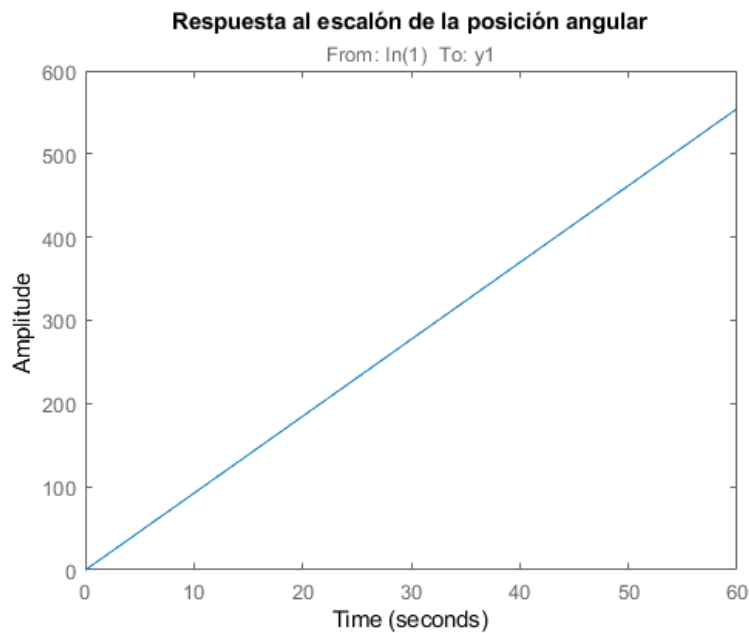


Figura 45: Respuesta del sistema de posición al escalón de 0.5 Nm.

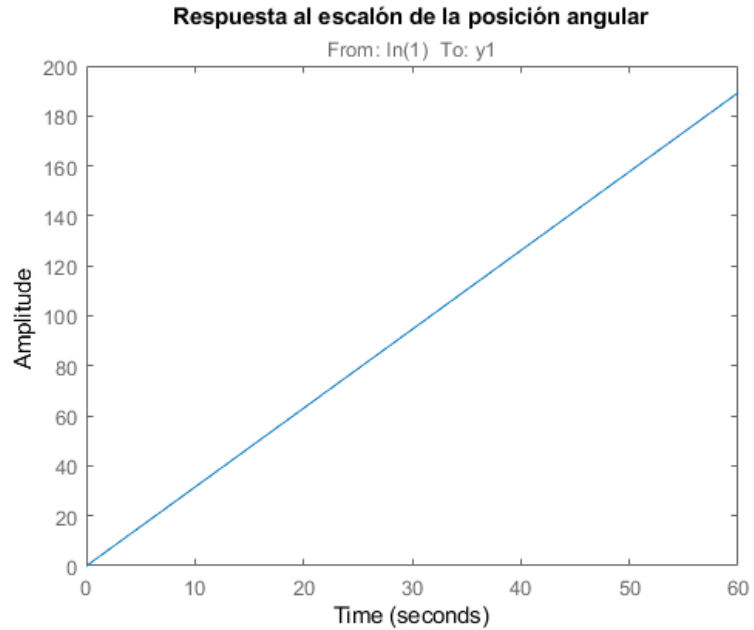


Figura 44: Respuesta del sistema de posición al escalón de 0.4 Nm.

Finalmente, con la herramienta de MATLAB *PID Tuner* se comprobó si era posible controlar a cada sistema obtenido mediante un controlador P, PD o PID. Los resultados se muestran en las Figuras 46 - 50, de las que se concluyó que, efectivamente, un control P o PD puede controlar al sistema.

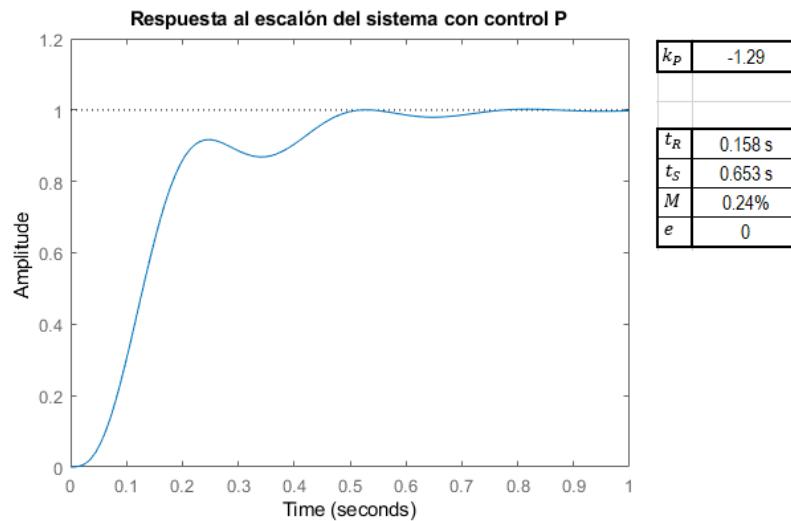


Figura 46: Respuesta al escalón del sistema 1 controlado.

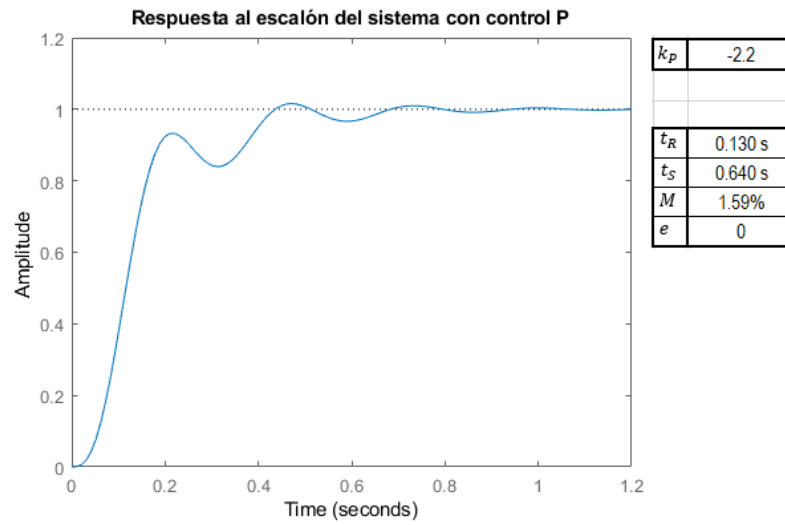


Figura 47: Respuesta al escalón del sistema 2 controlado con $C = kP$

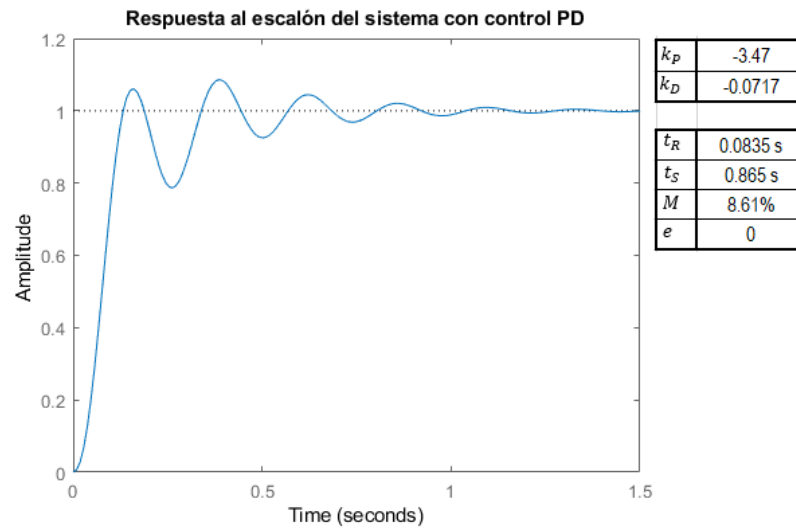


Figura 48: Respuesta al escalón del sistema 2 controlado con $C = kP + kDs$

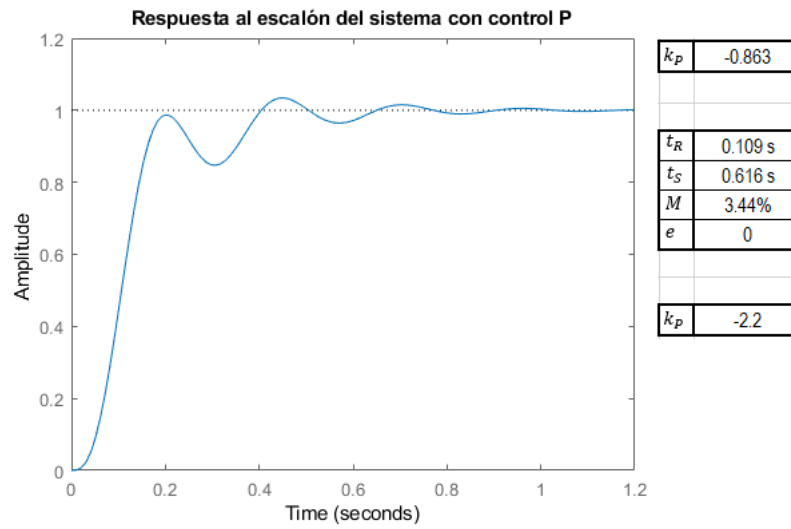


Figura 49: Respuesta al escalón del sistema 3 controlado con $C = kP$.

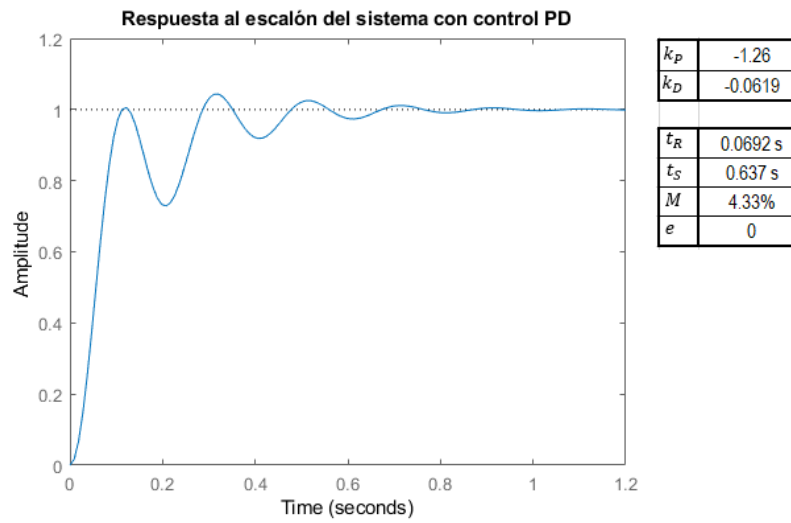


Figura 50: Respuesta al escalón del sistema 3 controlado con $C = kP + kDs$.

Con los resultados obtenidos en la fase de modelado se procedió a aplicar el control al sistema. De los resultados brindados por la herramienta *PID Tuner* de MATLAB se observó que el sistema puede controlarse bien con un control proporcional o con un control proporcional derivativo. A continuación, se muestran los resultados empleando ambas opciones.

10.1. Buen control y mal aterrizaje

10.1.1. Control proporcional

Utilizando la constante proporcional que se muestra en la Figura 49, se obtuvo el resultado mostrado en la Figura 51 para el control de la orientación. Cabe señalar que los primeros 2 segundos corresponden al tiempo en que el robot se posiciona y prepara para el salto; en el lapso de tiempo de 2 a 2.7 segundos se realiza el salto y por consiguiente el control; el resto del tiempo corresponde al impacto del robot con el suelo. Se observa que, cuando el robot está en el aire, el sistema oscila alrededor de la posición angular deseada. Estas oscilaciones son producto del uso exclusivo de la constante proporcional.

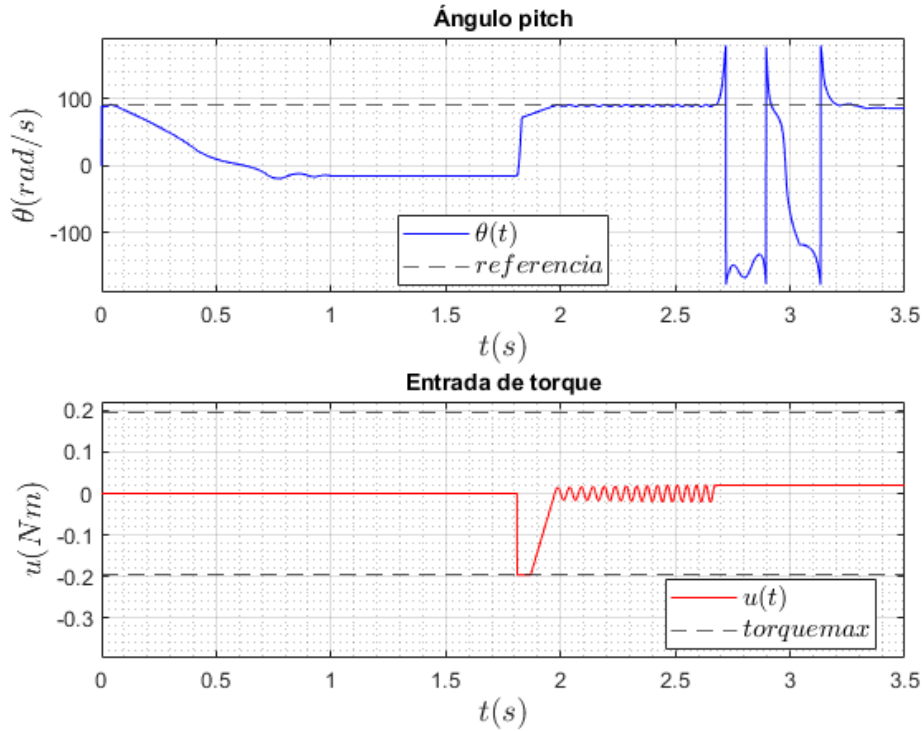


Figura 51: Sistema controlado con $kP = -0.863$.

10.1.2. Control proporcional derivativo

Con el fin de atenuar las oscilaciones que presenta el sistema anterior, se agregó una pequeña ganancia derivativa al mismo y se observa el resultado mostrado en la Figura 52. En esta se aprecia que la amplitud de las oscilaciones se reduce al igual que el tiempo que estas duran, lo que significa que el tiempo de asentamiento del sistema se reduce y éste logra reducir el error en un menor lapso de tiempo.

En los resultados hasta ahora mostrados se observa que se logra controlar la orientación del sistema en la forma deseada. Sin embargo, se hace notorio que la orientación del robot es inestable cuando este aterriza debido al impacto que experimenta con el suelo. Este comportamiento se muestra de forma más explícita en la Figura 53.

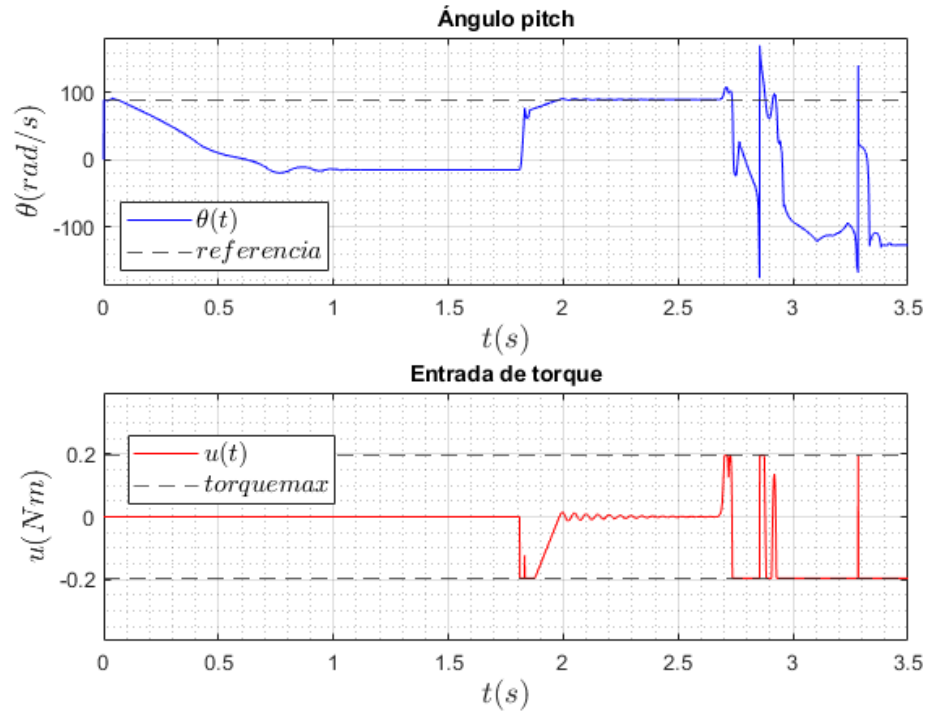


Figura 52: Sistema controlado con $k_P = 0.863$ y $k_d = 0.0055$.

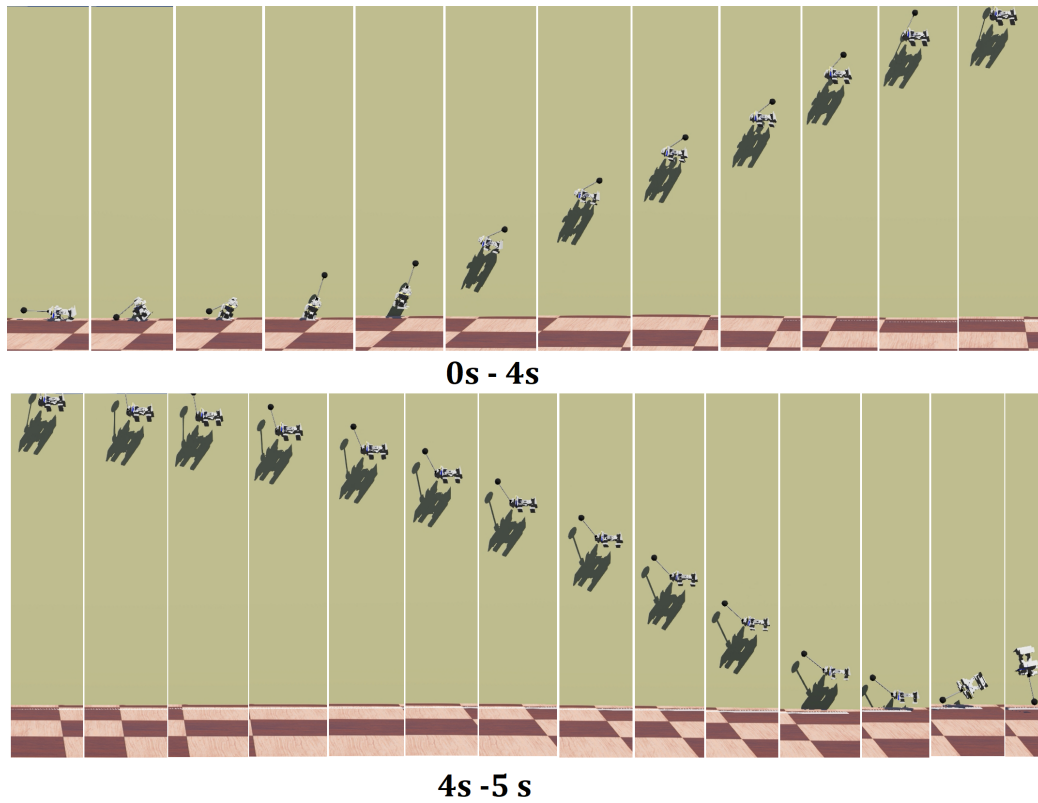


Figura 53: Trayectoria del robot con mal aterrizaje.

10.2. Buen control y buen aterrizaje

Para brindar una solución al mal aterrizaje se propone hallar la forma de reducir o amortiguar el impacto del robot con el suelo. Para esto se redefinió el diseño de las patas del robot, las cuales se diseñan ahora de forma que pueden deformarse al momento de la colisión, de forma que en dicha deformación se absorbe la energía que hace rebotar de forma abrupta al robot. La forma de definir dichas extremidades en Webots fue por medio de juntas revolutas pasivas con cierta constante de resorte y de amortiguamiento lo suficientemente altas para soportar el peso del robot y lo suficientemente bajas para absorber en el impacto elástico la mayor cantidad de energía. Las constantes que cumplen con estos requerimientos en Webots tienen los valores que se muestran en la Figura 54. En dicha imagen también se muestra que el campo *device* está vacío, lo cual Webots lo interpreta como una junta pasiva.

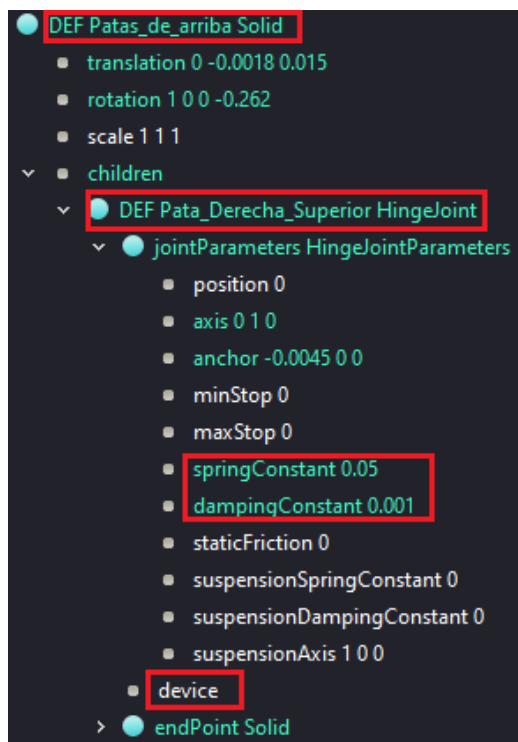
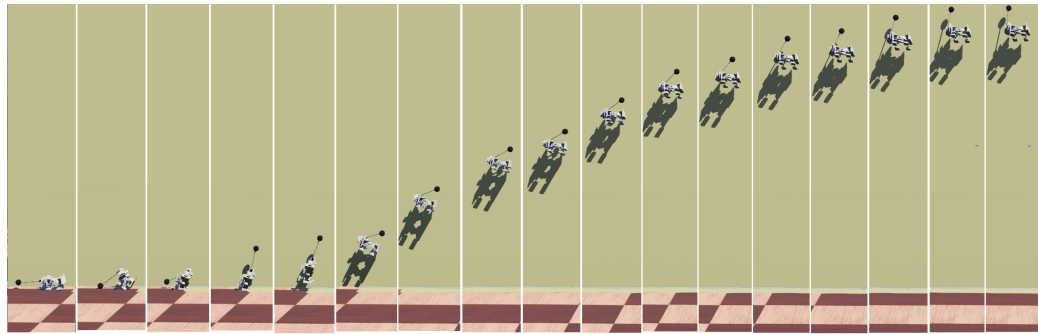
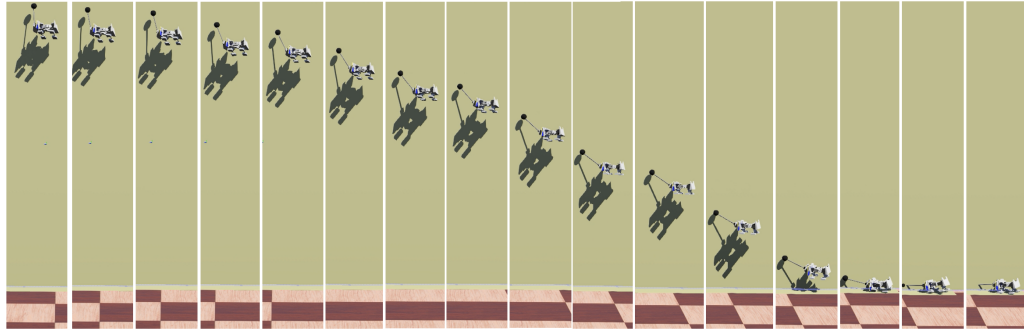


Figura 54: Definición de parámetros de las patas con amortiguamiento.

Posterior a incorporar las nuevas extremidades se procedió a poner a prueba nuevamente al robot para observar si el amortiguamiento era suficiente como para permitir un correcto aterrizaje. El resultado se muestra en la Figura 55, en la que se observa que el robot, efectivamente, aterriza de una buena manera y mantiene la orientación deseada. En la Figura 56 se muestran las gráficas de la orientación y la entrada de control del sistema con las nuevas extremidades, en la que se comprueba la buena postura del robot luego del aterrizaje.



0s - 4s



4s - 5s

Figura 55: Trayectoria del robot con buen aterrizaje.

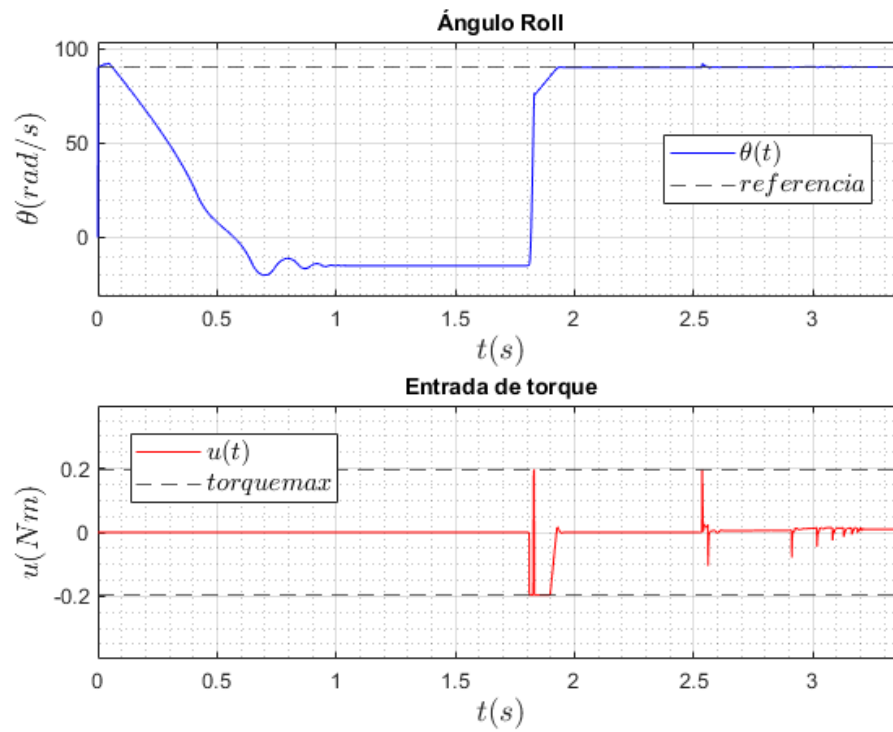


Figura 56: Ángulo roll y entrada de control del robot con buen aterrizaje.

Como se mencionó con anterioridad estas extremidades se definieron directamente en Webots, por lo que no se definieron en la fase de diseño en un modelo CAD desarrollado con Inventor. Sin embargo, la forma en que se propone implementarlas es por medio de dos geometrías sencillas que se muestran en las Figuras 57 y 58, las cuales servirían para sujetarlas al mecanismo y como pata, respectivamente, y que estarían fabricadas de PLA. El amortiguamiento se propone realizarlo a través de alambre galvanizado de zinc (como el utilizado para fabricar serchas) o un alambre de menor dureza que se deforme elásticamente con mayor facilidad. En la Figura 59 se ilustra esta propuesta.

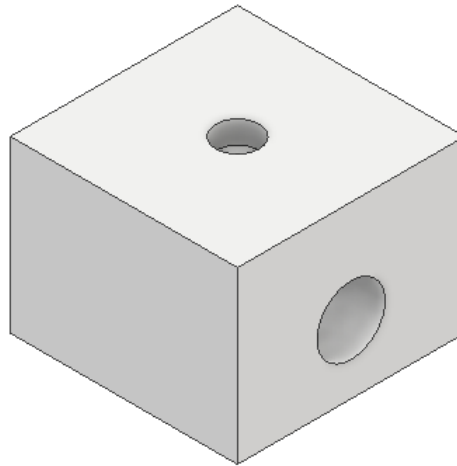


Figura 57: Geometría para sujetar las patas al mecanismo.

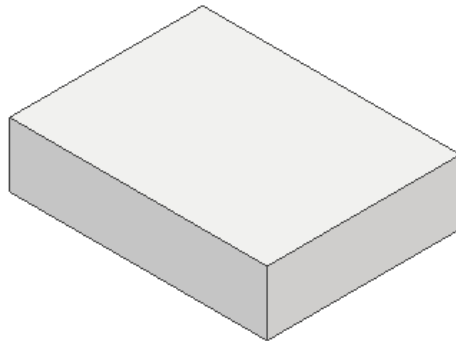


Figura 58: Geometría para simular la pata del robot.

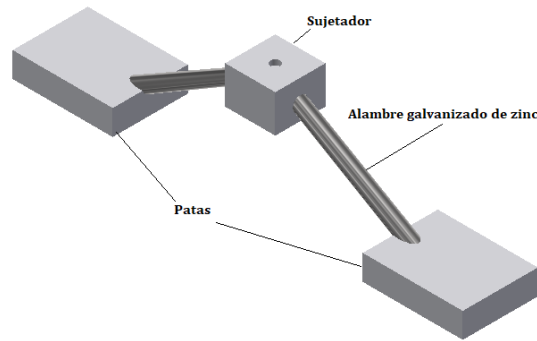


Figura 59: Propuesta para implementar las extremidades con amortiguamiento.

En los resultados anteriores se muestra que, si bien un control proporcional P es suficiente para mantener la orientación deseada del robot, un control proporcional derivativo PD mejora la respuesta del sistema permitiendo que éste se estabilice en un menor lapso de tiempo, lo cual también es beneficioso para un mejor aterrizaje. Este beneficio se hace notorio en la gráfica de la Figura 51, en la que se observa que el sistema con control proporcional no logra estabilizarse antes que el robot impacte con el suelo, lo cual puede provocar que éste se desestabilice aun más cuando, por ejemplo, golpea el suelo solamente con las patas traseras o solamente con las patas delanteras. Con el control PD se observa, en la Figura 52, que el sistema se estabiliza durante el tiempo que se encuentra en el aire, lo cual significa que las patas traseras y delanteras entran en contacto con el suelo al mismo tiempo y el aterrizaje mejora.

Comparando las gráficas de las Figuras 51 y 52, se observa en la primera que, después del impacto, existe una mayor cantidad de picos en el roll lo cual significa que el robot está experimentando revoluciones completas, esto es, el impacto ha generado que el sistema gire descontroladamente. En la segunda gráfica se observa que la cantidad de estos picos es menor y la amplitud disminuye tal y como se esperaba pues, si tanto las patas traseras como delanteras impactan al mismo tiempo con el suelo, se espera que la fuerza de reacción esté mejor distribuida provocando que el impacto desestabilice menos al robot. Sin embargo, a pesar que mejoró, el sistema continúa aterrizando de forma incorrecta, por lo que se concluye que la solución a este problema no se encuentra en la parte del software del robot, tratando de estabilizar la orientación más rápidamente.

Por lo anterior, se propuso una solución con un enfoque en la parte física del robot, por lo que se rediseñaron las patas del mismo, de forma que éstas fueran capaces de deformarse y absorber parte de la energía que se genera en el impacto. En las Figuras 55 y 56, en la que ya no se observan picos en el ángulo roll, se muestra que al realizar dichas modificaciones se resuelve efectivamente el problema y el robot es capaz ahora de atenuar el impacto y mantener la orientación deseada tanto en el aire como al aterrizar.

Finalmente, cabe señalar que la altura máxima que alcanzó el robot en el salto fue de 0.8984 m como se muestra en la Figura 60, lo cual generó un error del 2.09 % al compararla con la altura esperada de 0.88 m. Con lo anterior se confirmó una vez más que el mecanismo de salto logró su objetivo y cumplió con los requerimientos, y se observó que el movimiento de la cola o barra de inercia afectan no solamente el control del robot sino que también la altura que este puede alcanzar, ya que cuando esta no se actúa se alcanza una altura inferior (ver Figura 29) a diferencia de cuando sí se actúa (Figura 60).

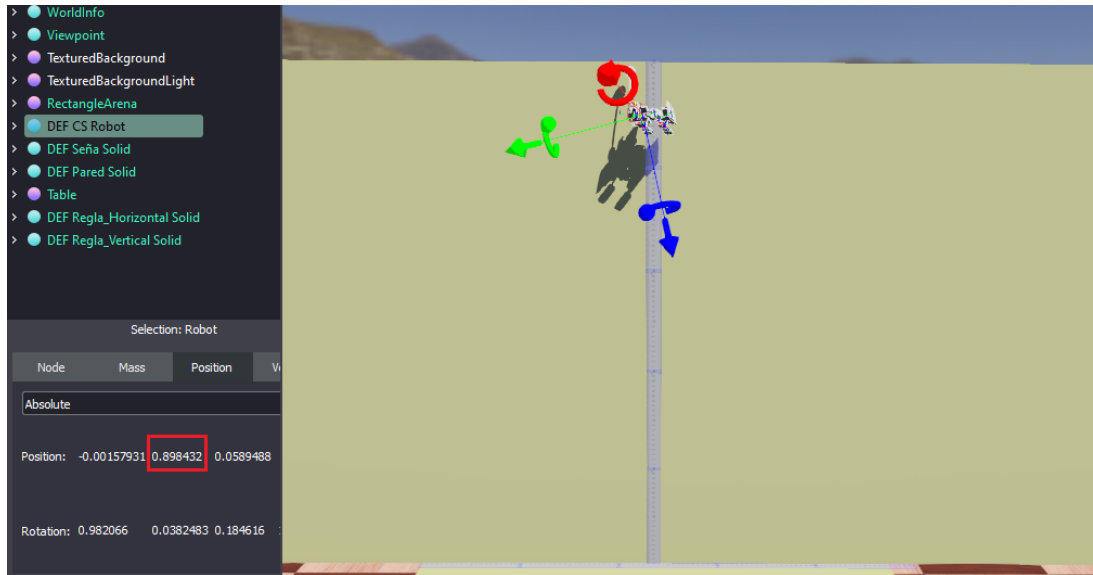


Figura 60: Altura máxima alcanzada por el robot.

El diseño se llevó a cabo en el software Autodesk Inventor, al cual de ahora en adelante se referenciará simplemente como Inventor. En la Figura 1 se muestra el proceso de diseño utilizado para la selección de componentes.

El diseño de robot presentado en esta investigación constó de dos partes básicas: un mecanismo de salto que replica la capacidad de las langostas, y de muchos otros insectos, de saltar una altura de al menos cuatro veces su longitud; y una barra de inercia que permite controlar la orientación del robot durante el salto para un correcto aterrizaje, lo cual replica la habilidad de ciertos animales, como las lagartijas, que utilizan su cola para dichos fines.

El mecanismo de salto diseñado cuenta con un motor, el cual incorpora un juego de engranajes en su caja reductora, y ocho resortes torsionales que, en conjunto, almacenan y liberan energía potencial para generar un salto que permitió al robot alcanzar una altura de al menos 88 cm.

El controlador proporcional derivativo PD implementado en el robot, haciendo uso de un motor para suministrar la entrada de torque requerido y una unidad de medición inercial (IMU) para tomar mediciones del ángulo roll y de la aceleración en el eje vertical, permitió el control de la orientación del mismo durante el salto, posibilitando así un correcto aterrizaje.

Por medio del software MATLAB se validó que el mecanismo de salto desarrollado fue capaz de almacenar suficiente energía para impulsar al robot a una altura mayor a cuatro veces su longitud; y en el entorno de simulación Webots se validó que el controlador desarrollado fue capaz de estabilizar la orientación del robot lo cual permitió su correcto aterrizaje.

Dadas las circunstancias complejas a través de la cuales se desarrolló esta investigación, su implementación física no fue llevada a cabo, por lo que, con base en los resultados obtenidos de los diferentes experimentos llevados a cabo asistidos por los softwares MATLAB y Webots, se recomienda la manufactura del diseño, el desarrollo del código en el controlador sugerido y comprobar que el robot funcione como se espera.

A pesar que el mecanismo desarrollado en esta investigación cumple con los objetivos establecidos de la misma, se observa que este no es el diseño óptimo que permita controlar al robot de la mejor forma posible. Se recomienda por lo anterior realizar un análisis matemático, como el llevado a cabo en [6], para optimizar todos los parámetros físicos del robot, esto es, que tenga el peso mínimo posible, la altura mínima posible, hallar la longitud de la barra de inercia óptima para el control de orientación, etc.

Finalmente, dado que el control del sistema fue desarrollado con base en un sistema linealizado, es decir aproximado, obtenido a partir de métodos heurísticos, se recomienda implementar un controlador basado en la dinámica real del sistema, esto es, hallar el modelo matemático que describa el comportamiento dinámico del robot, lo cual permitirá incluso aplicar métodos más modernos y óptimos de control como el LQR.

-
- [1] P. Castillo, “Construcción de un Mecanismo Bio Inspirado Para Auto Volteo de Vehículos Terrestres,” en *Trabajo de graduación en modalidad de Tesis*, 2019.
 - [2] J. Koh, S. Jung, M. Noh, S. Kim y K. Cho, “Flea inspired catapult mechanism with active energy storage and release for small scale jumping robot,” en *2013 IEEE International Conference on Robotics and Automation*, 2013, págs. 26-31.
 - [3] W. Liu, F. Li, X. Fu, C. Stefanini, G. Bonsignori, U. Scarfogliero y P. Dario, “Jumping Like an Insect: From Biomimetic Inspiration to a Jumping Minirobot Design,” ene. de 2013, págs. 207-221. DOI: 10.1007/978-1-4419-9985-6_11.
 - [4] J. Zhao, J. Xu, B. Gao, N. Xi, F. J. Cintrón, M. W. Mutka y L. Xiao, “MSU Jumper: A Single-Motor-Actuated Miniature Steerable Jumping Robot,” *IEEE Transactions on Robotics*, vol. 29, n.º 3, págs. 602-614, 2013.
 - [5] G. Jung, C. S. Casarez, J. Lee, S. Baek, S. Yim, S. Chae, R. S. Fearing y K. Cho, “JumpRoACH: A Trajectory-Adjustable Integrated Jumping-Crawling Robot,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, n.º 3, págs. 947-958, 2019.
 - [6] J. Zhao, T. Zhao, N. Xi, F. J. Cintrón, M. W. Mutka y L. Xiao, “Controlling aerial maneuvering of a miniature jumping robot using its tail,” en *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, págs. 3802-3807.
 - [7] E. Zadunaisky, *Sistemas dinámicos, teorías y métodos numéricos computacionales*, first. Universidad de Buenos Aires., 2009.
 - [8] M. B. Ortiz-Moctezuma, *Sistemas dinámicos en tiempo continuo: Modelado y simulación*. dic. de 2015, ISBN: 978-84-944673-2-5. DOI: 10.3926/oss.25.
 - [9] E. H. Castellanos, *Análisis comparativo entre técnicas de diseño sistemas de control clásico y moderno. Aplicación al modelado y control de un sistema de tanques*. Universidad Pontificia Bolivariana, 2012.
 - [10] K. Ogata, *Ingeniería de Control Moderna*. Tom Robbins, 1998.
 - [11] L. P. y J. Martínez, *Diseño e implementación de un control óptimo LQR con la tarjeta Raspberry Pi*, Universidad Distrital Francisco José de Caldas, 2016.

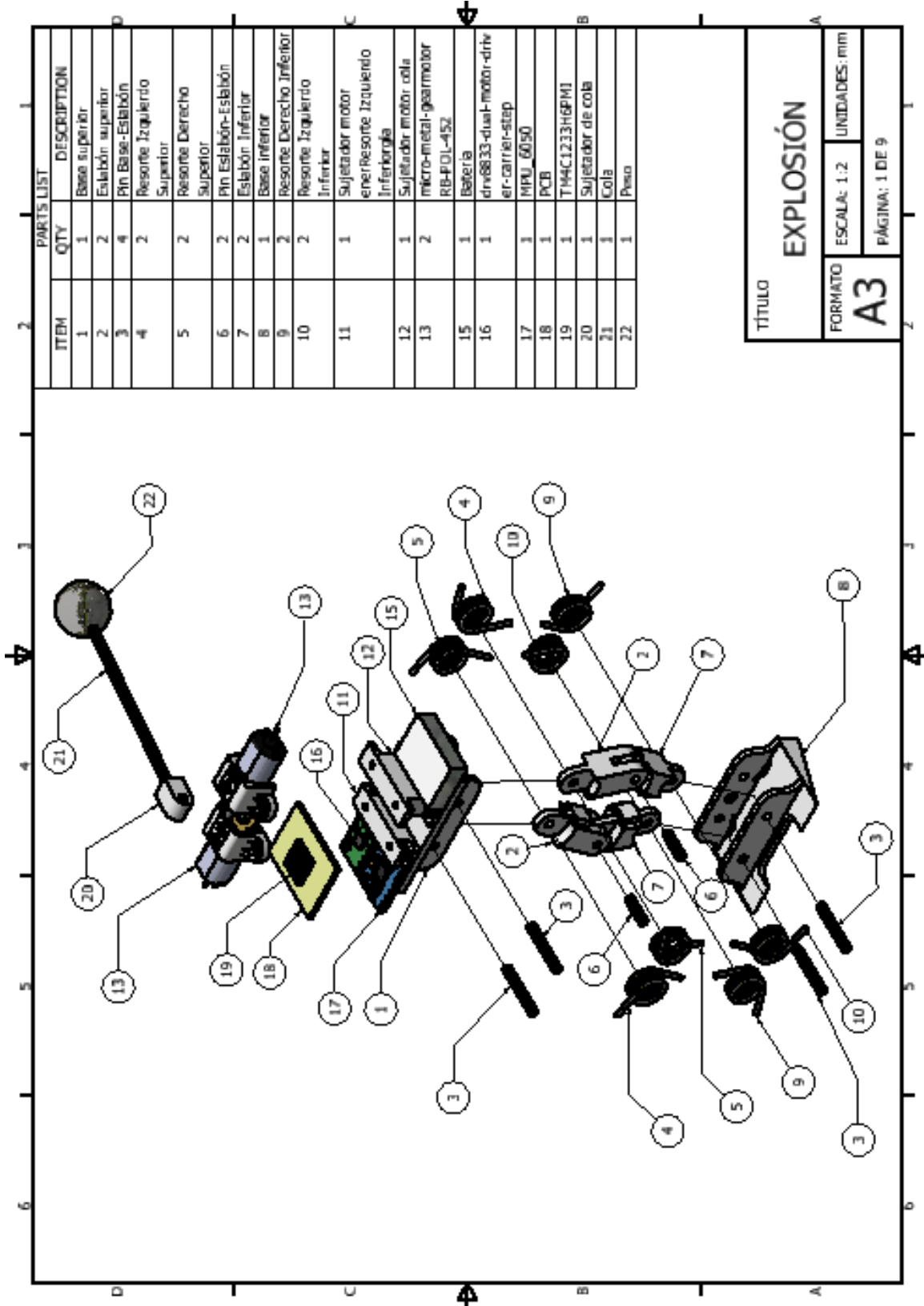
- [12] L. R. y E. Alcorta, “De la representación de sistemas Euler-Lagrange a la Hamiltoniana generalizada,” *Revista Electrónica Nova Scientia*, vol. 17, n.º 14, 2015.
- [13] M. H. Raibert, *Legged Robots that Balance*. The MIT Press, 1986.
- [14] J. Serway A. y Jewett, *Física para ciencias e ingeniería*, ninth. CENGAGE Learning, 2015, vol. 1.
- [15] A. I. en Español. (jul. de 2017). “Autodesk Inventor en Español DESDE CERO,” dirección: <https://www.youtube.com/channel/UCYqLEBIXaVSXQVUoIKk9gSw/playlists>.
- [16] C. Ltd. (). “Webots User Guide R2020a revision 2: Tutorials,” dirección: <https://cyberbotics.com/doc/guide/tutorials>.
- [17] M. Juston. (jul. de 2019). “Exporting Autodesk Inventor CAD models to Webots,” dirección: <https://www.youtube.com/watch?v=L0FVsFD2rS4>.
- [18] C. Ltd. (2020). “Webots User Guide R2020a revision 2: World Info,” dirección: <https://cyberbotics.com/doc/reference/worldinfo>.
- [19] P. Corporation. (2020). “Comparing Products in 7 Categories,” dirección: <https://www.pololu.com/search/compare/60>.
- [20] 3. C. Central. (2020). “Piezas, ensamblajes (y mucho más) de modelos CAD en 2D y 3D de usuarios y proveedores certificados,” dirección: <https://www.3dcontentcentral.es>.
- [21] W. Heitler. (2007), dirección: <https://www.st-andrews.ac.uk/~wjh/jumping/perform.htm>.
- [22] T. Libby, T. Moore, E. Chang-Siu, D. Li, D. Cohen, A. Jusufi y R. Full, “Tail-assisted pitch control in lizards, robots and dinosaurs,” *Nature*, vol. 481, págs. 181-4, ene. de 2012. DOI: 10.1038/nature10710.
- [23] P. Corporation. (2020). “DRV8833 Dual Motor Driver Carrier,” dirección: <https://www.pololu.com/product/2130>.
- [24] Energia. (2020). “Language Reference,” dirección: <https://energia.nu/reference/>.
- [25] ADDICORE. (2020). “GY-521 MPU6050 3-Axis Gyroscope and Accelerometer IMU,” dirección: <https://www.addicore.com/GY-521-MPU6050-p/170.htm>.
- [26] P. Corporation. (2020). “100:1 Micro Metal Gearmotor LP 6V with Extended Motor Shaft,” dirección: <https://www.pololu.com/product/2204>.
- [27] D. P. S.A.S. (2020). “Tornillo M2 Diametro 2mm + tuerca,” dirección: <https://demosspro.com/cb/inicio/427-40-tornillo-m2-diametro-2mm-largo-5-8-10mm-tuerca-pac-x-2.html>.
- [28] N. Chung S. y Nancy Pollard, “Predictable behavior during contact simulation: a comparison of selected physics engines,” *COMPUTER ANIMATION AND VIRTUAL WORLDS*, pág. 19, 2016.

14.1. Códigos

Enlace a github de los códigos utilizados en MATLAB y Webots: <https://github.com/JerryRivera30/Codigos>

14.2. Planos de construcción

Los planos se muestran a partir de la página siguiente.



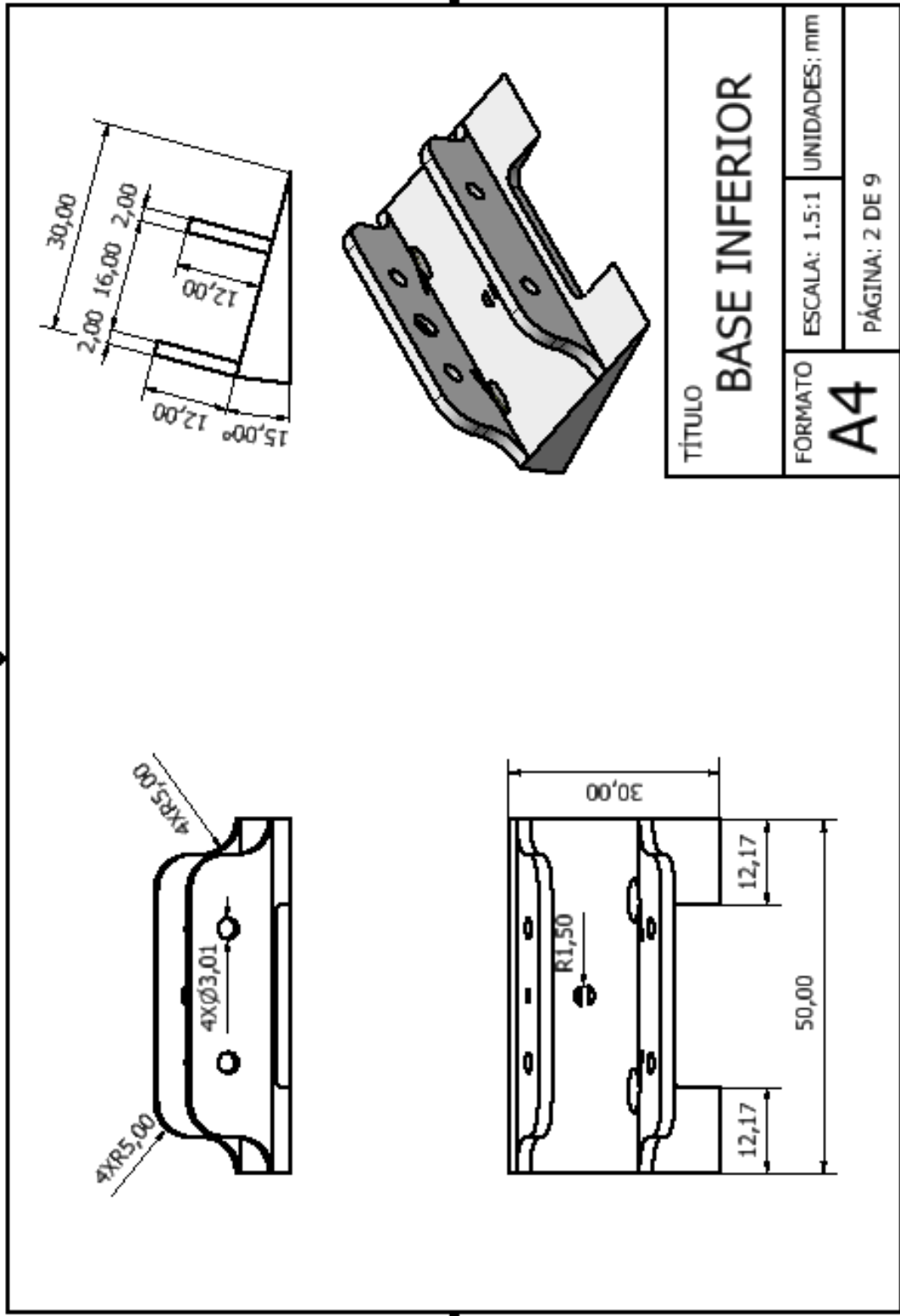
PARTS LIST	
ITEM	DESCRIPTION
1	Base superior
2	Eslabón superior
3	Pin Base-Eslabón
4	Resorte Izquierdo Superior
5	Resorte Derecho Superior
6	Pin Eslabón-Eslabón
7	Eslabón Inferior
8	Base inferior
9	Resorte Derecho Inferior
10	Resorte Izquierdo Inferior
11	Sujetador motor emer/Resorte Izquierdo Inferior
12	Sujetador motor cola micro-metal-gearmotor
13	RB-POL-452
15	Batería
16	drv6833-dual-motor-drive-er-carri-el-step
17	MPU_6050
18	PCB
19	TM4C1233H6PM1
20	Sujetador de cola
21	Cola
22	Peso

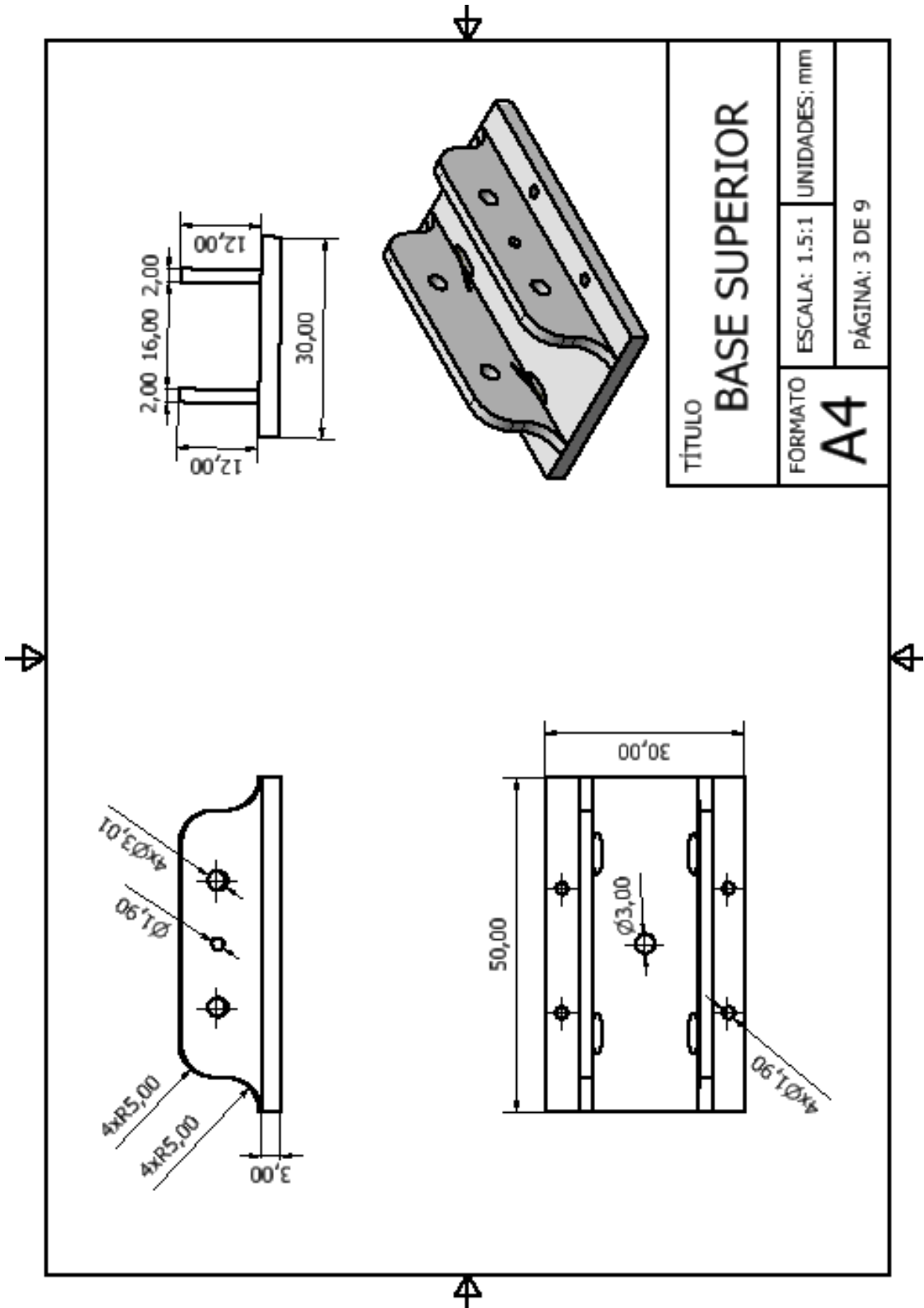
TÍTULO
EXPLOSIÓN

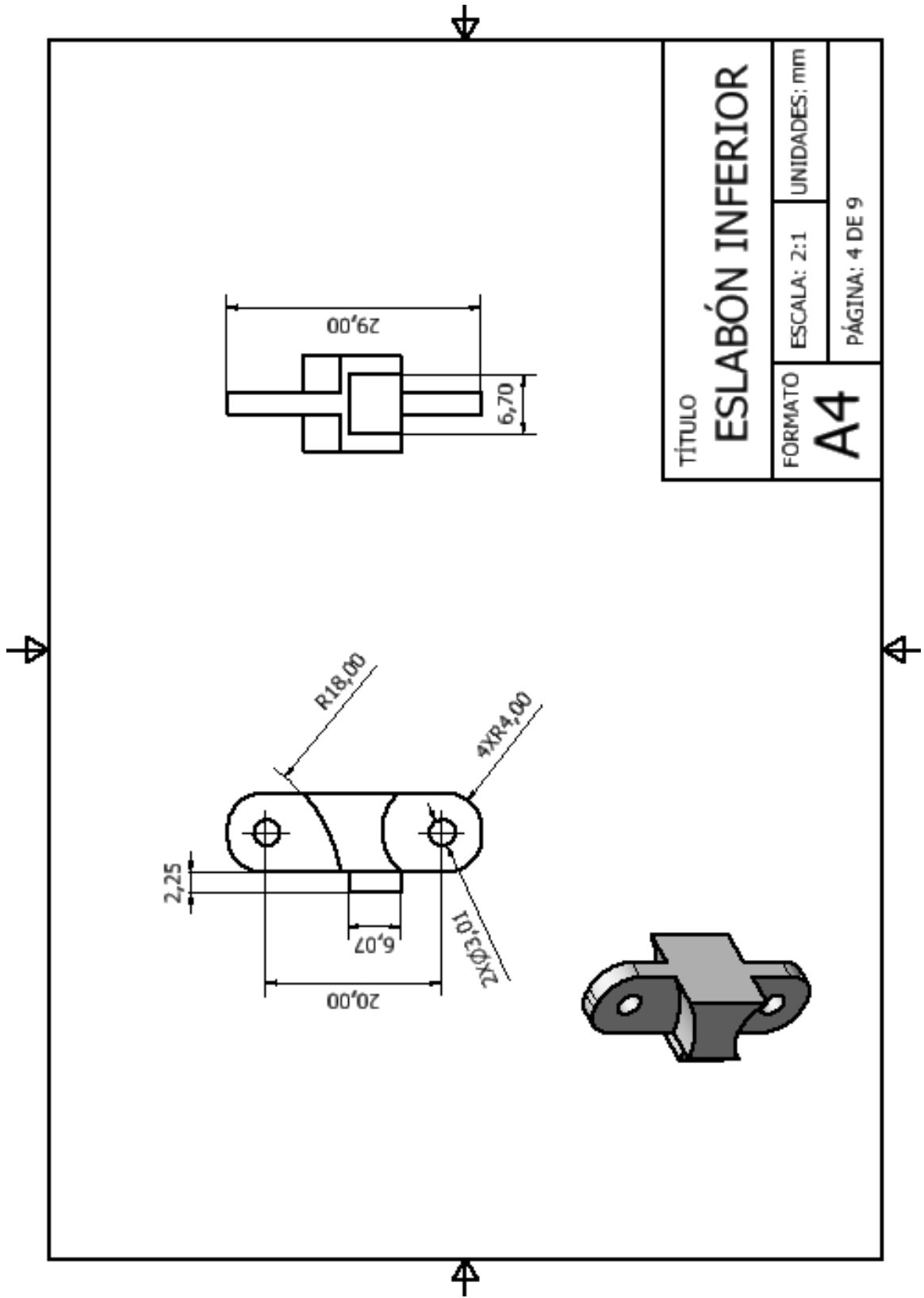
FORMATO
A3

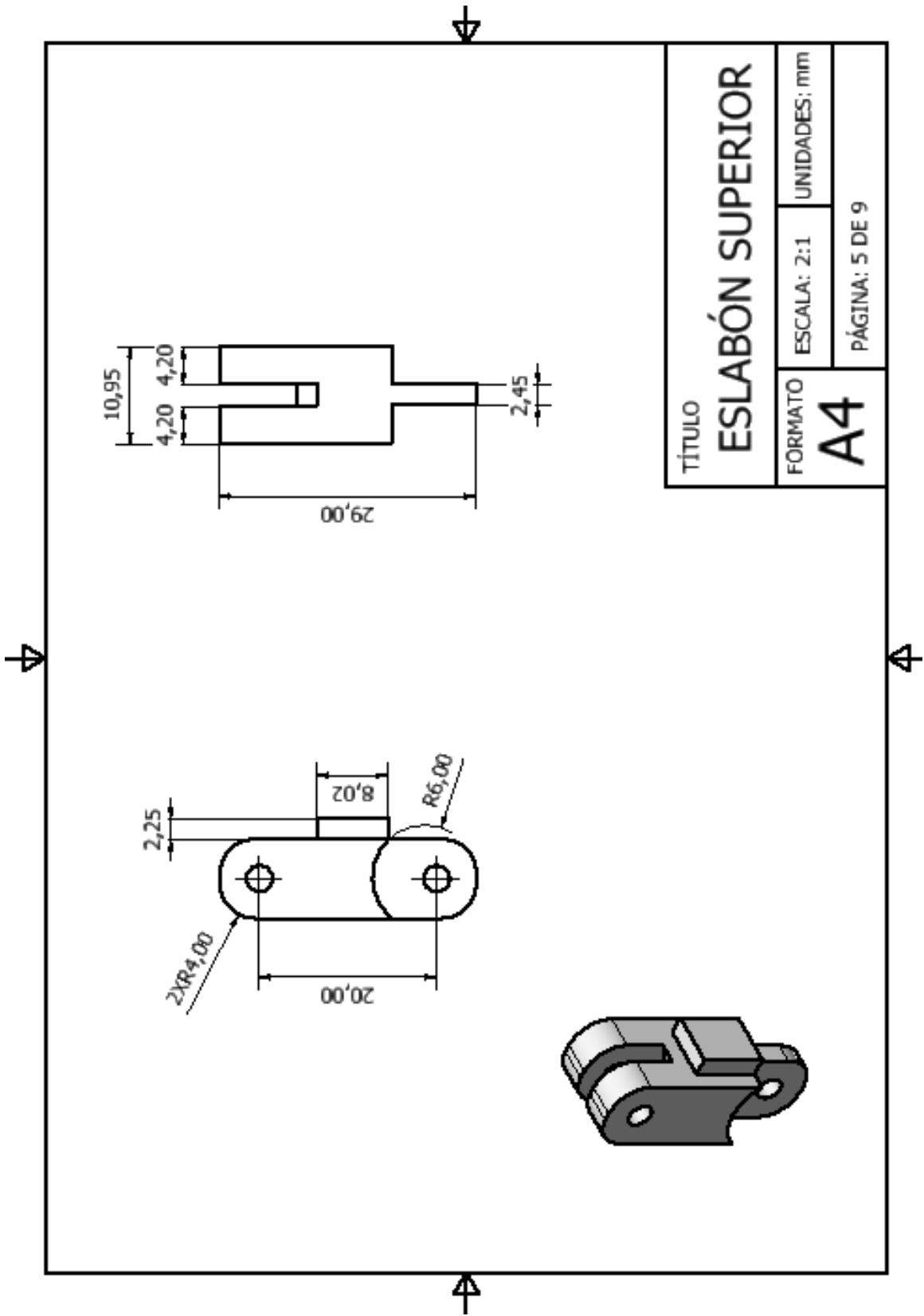
ESCALA: 1:2 UNIDADES: mm

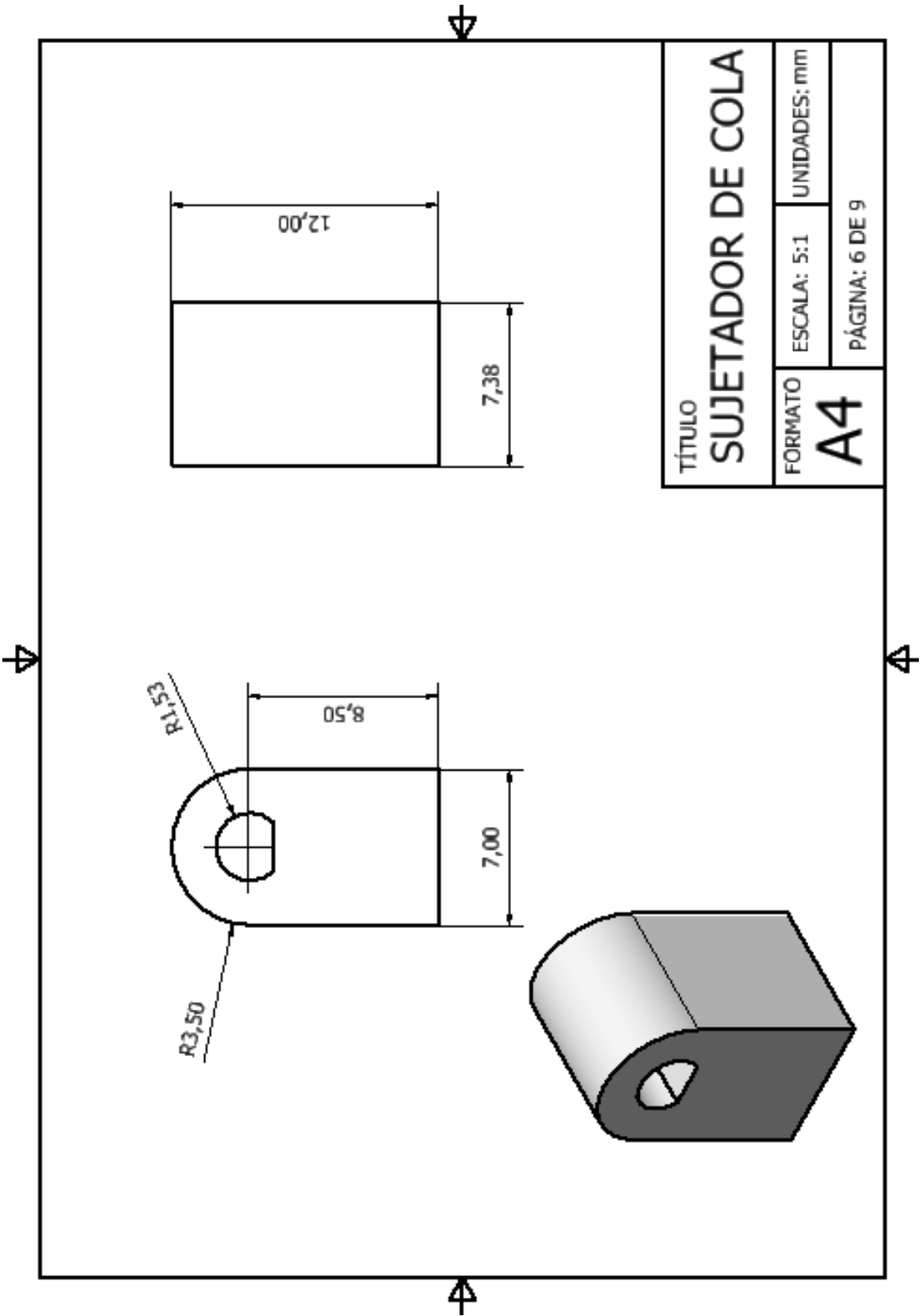
PÁGINA: 1 DE 9

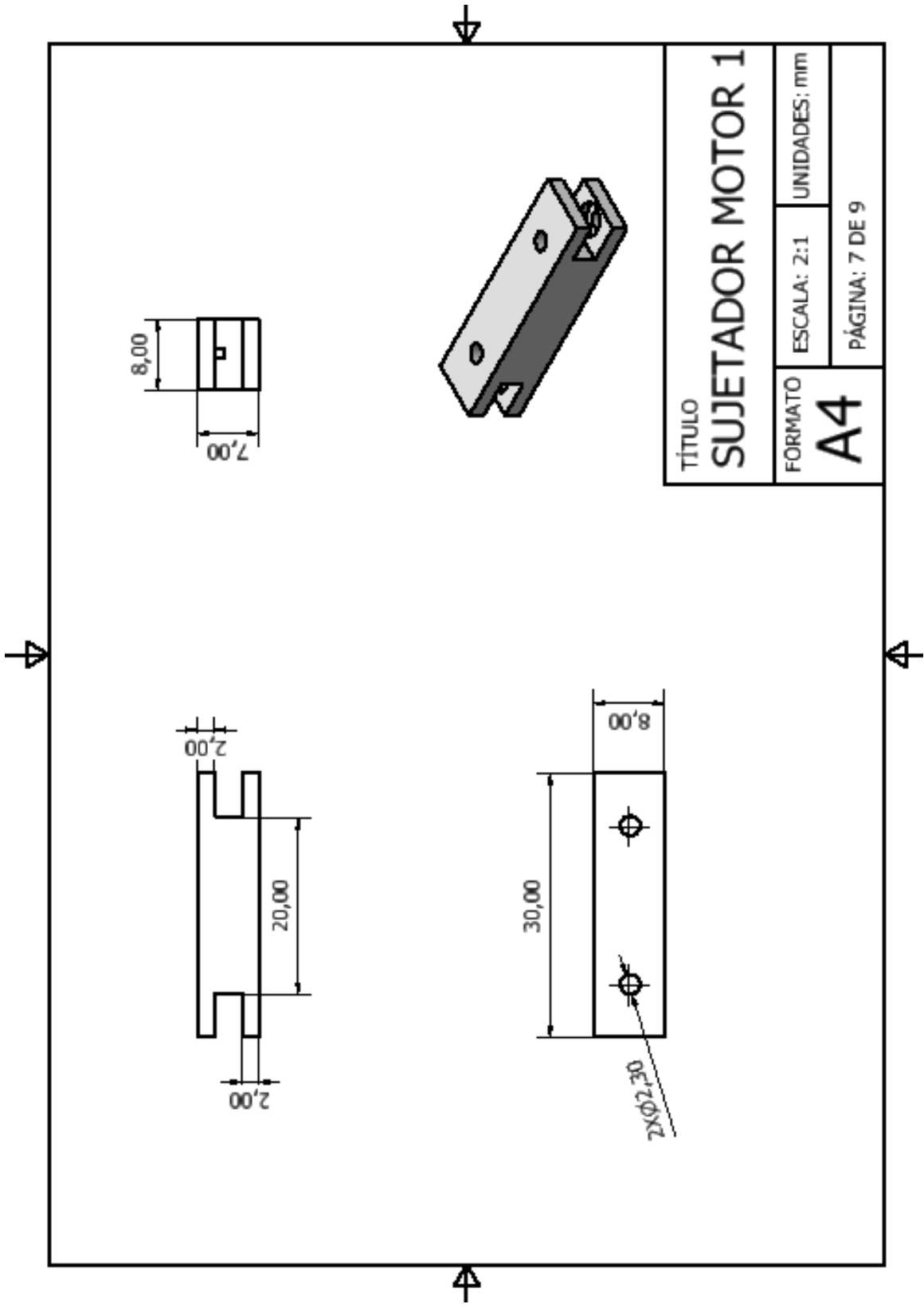




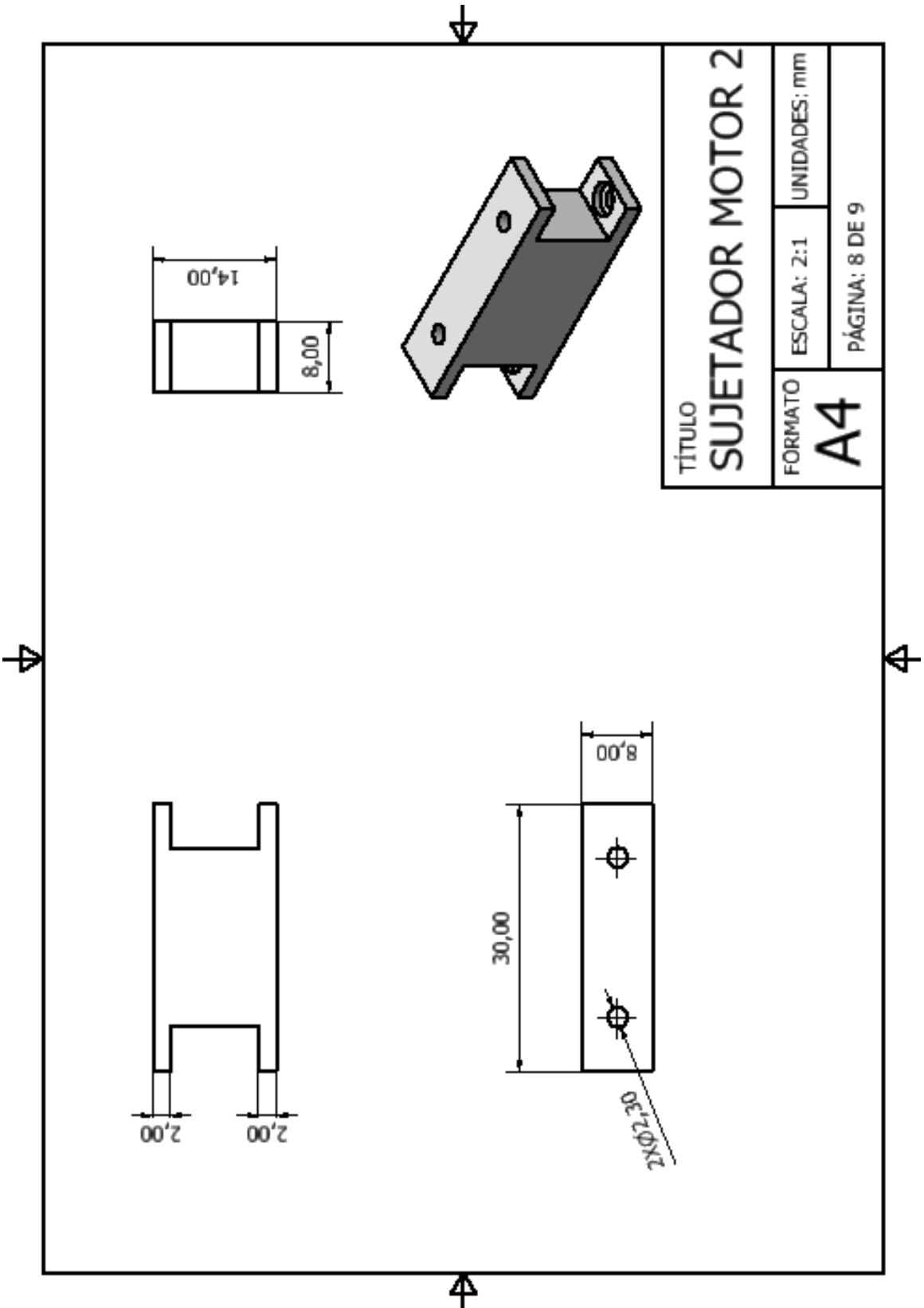


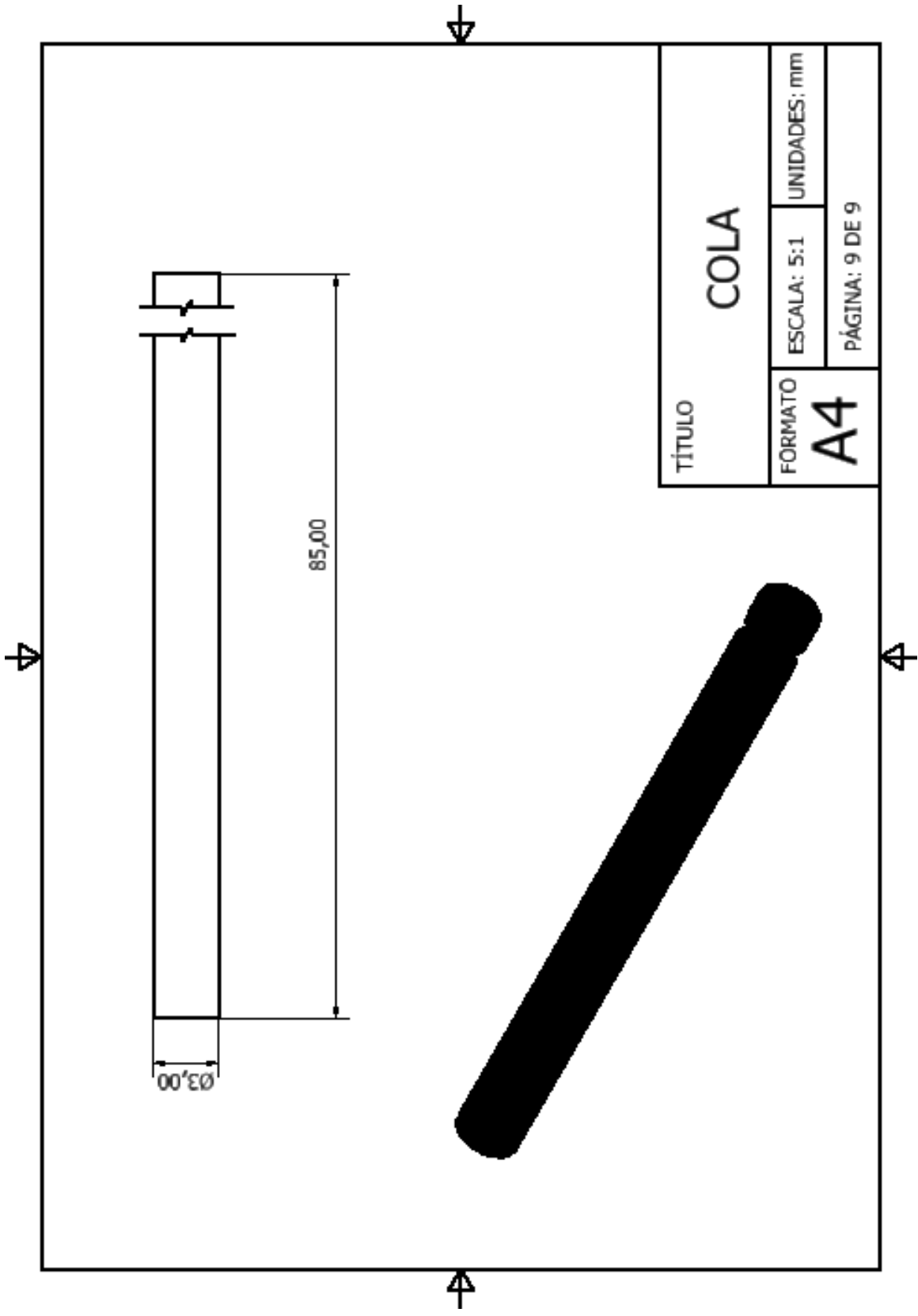






TÍTULO		
SUJETADOR MOTOR 1		
FORMATO	ESCALA: 2:1	UNIDADES: mm
A4		PÁGINA: 7 DE 9





TÍTULO			COLA	
FORMATO	ESCALA: 5:1	UNIDADES: mm		
A4		PÁGINA: 9 DE 9		

ODE: Open Dynamics Engine. 56

pitch: ángulo de elevación.. 24, 58, 64

roll: ángulo de alabeo.. 24, 58, 71

SISO: Single Input Single Output. 25

SMA: Shape Memory Alloy. 3

yaw: ángulo de cabeceo.. 24