



Facultad de Ingeniería Eléctrica
Universidad Michoacana de San Nicolás de Hidalgo

DESARROLLO DE UN EXOESQUELETO DE EXTREMIDADES INFERIORES PARA REHABILITACIÓN

TESIS

Que para obtener el grado de
LICENCIATURA EN INGENIERÍA ELÉCTRICA

presenta

María Luisa Romero Muñoz

Asesor de Tesis

Dr. Leonardo Romero Muñoz

Noviembre 2017

Esta tesis se la dedico a mi mamá Adriana y a mi hijo Christyan.

Agradezco a todas las personas que me han brindado su valioso apoyo y compañía, en especial:

- A mi mamá, mis abuelos y mis hermanos, por estar siempre a mi lado, demostrar incondicionalmente su amor hacia mi y apoyarme en todos los aspectos.
- A mis profesores, que a lo largo de la licenciatura me brindaron su conocimiento y guía.
- A mi asesor de tesis, mi tío Leonardo, por todo su apoyo y paciencia.

Agradezco a Dios, por la vida y por las personas que me rodean, y por permitirme realizar el sueño de prepararme en la ingeniería eléctrica.

Resumen

Es frecuente que las personas en la actualidad presenten distintos tipos de discapacidad motriz ocasionada por enfermedad, accidentes o discapacidades congénitas. Esta situación ha originado el desarrollo de dispositivos llamados exoesqueletos para apoyar a éstas personas en su rehabilitación o en su actividad cotidiana. Esta tesis aborda el desarrollo de un exoesqueleto no antropomórfico de extremidades inferiores para ser utilizado en la rehabilitación de personas adultas.

El exoesqueleto desarrollado utiliza como base un mecanismo de línea recta llamado mecanismo de Peaucellier-Lipkin (P-L), compuesto de 8 barras y 6 uniones o articulaciones. En una de las articulaciones se inserta un actuador rotacional y al hacerlo girar, el extremo del mecanismo describe una trayectoria de línea recta. El mecanismo de P-L se extiende a tres grados de libertad (GDL), permitiendo que dos de las barras puedan modificar su longitud, de manera que se pueden generar tanto trayectorias rectas como cóncavas o convexas, que pueden aprovecharse para los movimientos de las piernas del exoesqueleto.

El exoesqueleto utiliza 6 servomotores comerciales: 4 lineales y 2 rotacionales, además de 2 servomotores rotacionales de alto par que fue necesario implementar, debido a que no se encontraron servomotores comerciales con las características requeridas. Al coordinar los movimientos de los servomotores el exoesqueleto puede generar trayectorias de caminado.

Esta tesis se enfoca en la cinemática directa del exoesqueleto, el control de los servomotores y de la estrategia de caminado utilizando un microcontrolador arduino; así como el desarrollo de un servomotor de alto par (que no fue posible encontrar en el mercado) compuesto por un motor de corriente directa, un encoder y un microcontrolador.

Palabras clave: Exoesqueletos, Exoesqueletos no Antropomórficos, Sistemas de Rehabilitación, Servomotores, Sistemas de Caminado.

Abstract

It is common that people present different types of motor disability caused by illness, accidents or congenital disabilities. This situation has led to the development of devices called exoskeletons to support people in their rehabilitation or in their daily activities. This thesis addresses the development of a non anthropomorphic exoskeleton of lower extremities to be used in the rehabilitation of adults.

The developed exoskeleton uses a straight-line mechanism called the Peaucellier-Lipkin mechanism (P-L) as a base, composed of 8 bars and 6 joints. In one of the joints, a rotational actuator is inserted and by rotating it, the end of the mechanism describes a straight line path. The P-L mechanism extended to three degrees of freedom (GDL), allows that two bars can modify their length. In this way, straight and concave or convex paths can be generated and used for the movements of the legs of the exoskeleton.

The exoskeleton uses 6 commercial servomotors: 4 linear and 2 rotational, plus 2 high-torque rotational servomotors developed, because there is not commercial servomotors with the required characteristics. By coordinating the motions of the servomotors the exoskeleton can generate trajectories of walking.

This thesis focuses on the direct kinematics of the exoskeleton, the control of servomotors and the walking strategy using an Arduino microcontroller; as well as the development of a high torque servomotor (which was not possible to find in the market), composed of a direct current motor, an encoder and a microcontroller.

Lista de Publicaciones

“Exoesqueleto bípedo basado en mecanismo de 8 barras articuladas como sistema de rehabilitación”

Jorge Curiel Godoy y María Luisa Romero Muñoz.

Primer Lugar de la Categoría de Prototipos de Desarrollo Tecnológico Nivel Superior. 6to Encuentro Estatal de Prototipos de Robótica y Desarrollo Tecnológico. Realizado en Morelia Michoacán el 23 de febrero de 2017 en las instalaciones del CIAC-UMSNH.

Contenido

Dedicatoria	III
Resumen	V
Abstract	VII
Lista de Publicaciones	IX
Contenido	XI
Lista de Figuras	XIII
Lista de Tablas	XV
 1. Introducción	 1
1.1. Planteamiento del problema	1
1.2. Antecedentes	3
1.3. Clasificación y aplicaciones de los exoesqueletos	5
1.4. Contexto del desarrollo realizado	7
1.5. Objetivo de la Tesis	8
1.6. Justificación	8
1.7. Logros Alcanzados	9
1.8. Organización de la tesis	9
 2. Mecanismo Peaucellier-Lipkin extendido	 11
2.1. Mecanismo de P-L	11
2.2. Mecanismo de P-L extendido a 3 grados de libertad	13
2.3. Cinemática Directa del Mecanismo Peaucellier-Lipkin extendido a 3 GDL	15
2.4. Experimentos con el modelo matemático	19
2.4.1. Experimentos variando los parámetros a y b	21
2.5. Comentarios finales	25
 3. Elementos del exoesqueleto	 27
3.1. Actuadores lineales y rotacionales	28
3.2. Fuente de alimentación	30
3.3. Funcionamiento de los servomotores	30
3.3.1. Servomotores rotacionales	31
3.3.2. Servomotores lineales	33
3.4. Control de un servomotor utilizando un microcontrolador arduino	34
3.5. Comentarios finales	36

4. Implementación de un servomotor rotacional de alto par	37
4.1. Elementos del servomotor	37
4.1.1. Motor y caja de engranes	38
4.1.2. Encoder absoluto de 10 bits	38
4.1.3. Driver del motor de CD	40
4.1.4. Fuentes de alimentación	41
4.1.5. Microcontrolador	41
4.1.6. Código gray	42
4.1.7. Lectura del encoder usando un microcontrolador	43
4.2. Implementación del control utilizando un microcontrolador arduino	46
4.3. Comentarios finales	49
5. Estación para la generación de trayectorias de caminado	51
5.1. Microcontrolador utilizado	51
5.2. Señales de entrada y salida del microcontrolador	52
5.3. Algoritmos de generación de trayectoria de caminado	52
5.3.1. Subir y bajar piernas	53
5.3.2. Secuencia para dar un paso	53
5.4. Comentarios finales	55
6. Pruebas	57
6.1. Lectura del encoder absoluto de 10 bits	57
6.2. Servomotor de alto par	59
6.3. Secuencia de caminado	60
6.4. Comentarios finales	61
7. Conclusiones y trabajos futuros	63
7.1. Conclusiones	63
7.2. Trabajos futuros	64
A. Programa de caminado	65
Referencias	73

Lista de Figuras

1.1. Porcentaje de población con discapacidad	2
1.2. Porcentaje de población con discapacidad por edad	3
1.3. Exoesqueleto en su estado actual.	7
2.1. Mecanismo Peaucellier-Lipkin de línea recta exacta.	12
2.2. Caso 2: $L_{AB} < L_{BC}$. Mecanismo P-L describiendo un arco cóncavo.	14
2.3. Caso 3: $L_{AB} > L_{BC}$. Mecanismo P-L describiendo un arco convexo.	14
2.4. Casos 4 y 5: Trayectorias rectas a diferentes distancias de A.	15
2.5. Mecanismo P-L en un plano, con origen en el punto A.	16
2.6. Mecanismo P-L cuando la barra BC gira un ángulo θ	16
2.7. Sistema de referencia X', Y' alineado al lado w	17
2.8. Mecanismo P-L en el plano X', Y'	17
2.9. Gráfica obtenida por Matlab.	21
2.10. Gráfica de la trayectoria del punto F obtenida por Matlab.	22
2.11. Gráfica de mecanismo para $a > b$	23
2.12. Gráfica de mecanismo para $a < b$	24
2.13. Gráfica de mecanismo para $a = 0.95, b = 0.95$	24
2.14. Gráfica de mecanismo para $a = 1.05, b = 1.05$	25
3.1. Vista lateral del exoesqueleto.	27
3.2. Servomotor lineal HDLS-4-30-12V marca Servocity 2017.	28
3.3. Servomotor rotacional Torxis Servo i00600 marca Gearwurx [Gea17].	30
3.4. Fuente de alimentación conmutada DC12V40A	30
3.5. Componentes de un servomotor rotacional [ser17a].	31
3.6. Señal PWM para controlar el servomotor.	32
3.7. Posiciones de un servomotor para pulsos de 0.5 a 2.5 ms.	33
3.8. Un servomotor lineal por dentro [ser17b].	34
3.9. Control de un servomotor utilizando un microcontrolador arduino	34
3.10. Diagrama de conexiones [Ard17b].	35
4.1. Motor de CD Sumotor utilizado [motorGW11416517].	38
4.2. Encoder absoluto de 10 bits [TRD-NA1024NW17].	39
4.3. Características del encoder.	39
4.4. Características del encoder.	40

4.5. Driver de motor de CD Pololu VNH5019 [VNH501917].	41
6.1. Prueba del encoder absoluto.	58
6.2. Prueba del servomotor utilizando un esquema de control PID.	59
6.3. Prueba del servomotor utilizando un esquema de control Supertwisting. . .	59
6.4. Secuencia de imágenes del exoesqueleto dando un paso hacia adelante. . . .	60

Lista de Tablas

3.1.	Datos del servomotor lineal HDLS-4-30-12V.	29
3.2.	Datos del servomotor rotacional del zapato Torxis Servo i00600.	29
4.1.	Ejemplo de un código Gray de 3 bits.	42

Capítulo 1

Introducción

La discapacidad ha sido siempre un factor que afecta la calidad de vida de una sociedad, pues influye directamente en las oportunidades de las personas para acceder a sus derechos fundamentales como educación y empleo.

Esta tesis aborda el desarrollo de un exoesqueleto para apoyar la rehabilitación de personas con discapacidad al caminar. Enseguida se describe el problema de discapacidad y su dimensión a nivel nacional.

1.1. Planteamiento del problema

De acuerdo a los resultados de la Encuesta Nacional de la Dinámica Demográfica (ENANID) del Instituto Nacional de Estadística y Geografía (INEGI), en 2014 había en México 120 millones de habitantes, de los cuales 7.2 millones reportan tener mucha dificultad para poder realizar alguna de las actividades básicas o no poder hacerlas (personas con discapacidad) y alrededor de 15.9 millones tienen dificultades leves o moderadas para realizar las mismas actividades (personas con limitación) [INEGI15]. Esto significa que para ese año el 6 % de la población presentó discapacidad de algún tipo y el 13 % tuvo alguna limitación. Es decir, alrededor de 13 de cada 100 personas tiene alguna limitación, un número bastante alto.

Según la ENANID 2014, las discapacidades más frecuentes a nivel nacional son: 64.1 % en caminar, subir o bajar usando sus piernas; con un 58 % en ver. La limitación

para hablar y comunicarse ocupa 28 %, siendo la discapacidad con menor porcentaje a nivel nacional. En la Figura 1.1 se muestran los porcentajes de población por tipo de discapacidad. Es importante tener en cuenta que la pregunta sobre discapacidad de la encuesta permite una respuesta múltiple, es decir, una persona puede declarar tener dificultad o limitación en más de una actividad. Por esta razón la suma de porcentajes es mayor al 100 %.

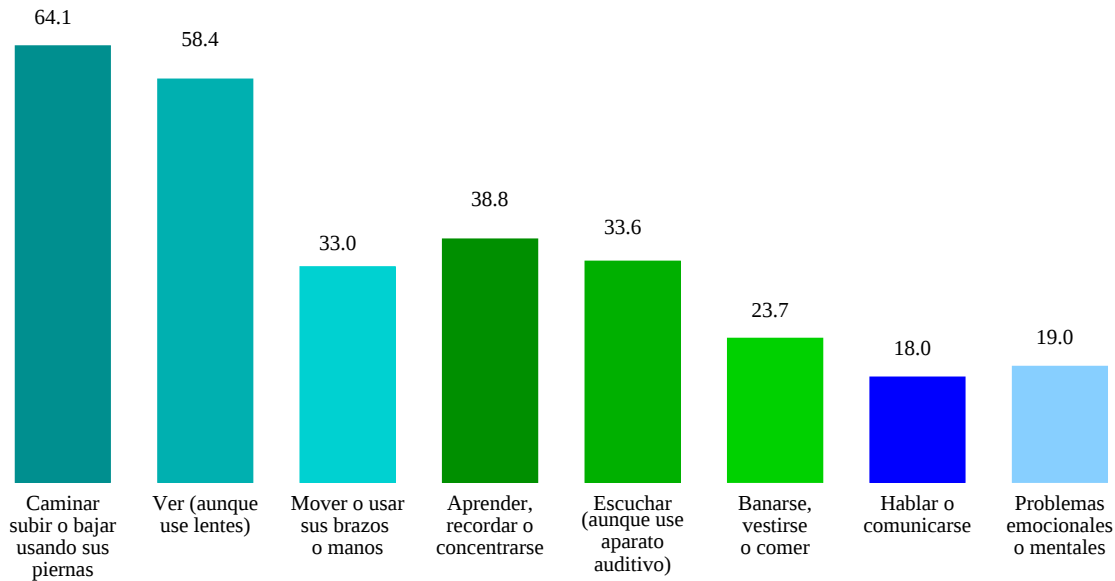


Figura 1.1: Porcentaje de población mexicana con discapacidad, por tipo de discapacidad, en 2014. Fuente: ENANID 2014.

Es conveniente destacar que un porcentaje de personas nace con discapacidad y otro la adquiere durante su vida. Las causas de estas discapacidades son variadas, pero son cuatro las principales: enfermedad, edad avanzada, nacimiento y accidente. Por cada 100 personas con discapacidad, 41 la adquieren por enfermedad, 33 por edad avanzada, 11 por nacimiento, 9 por accidente, 5 por otra causa y 1 por violencia.

Los grupos afectados por las discapacidades más frecuentes son los adultos y en particular los adultos mayores. Los mayores porcentajes de personas con dificultad para usar sus piernas se encuentran entre estos grupos mientras que los porcentajes disminuyen para los jóvenes y los niños. La Figura 1.2 presenta los porcentajes de personas con los diferentes tipos de discapacidad según su edad. Se observa que de la población con discapacidad, el porcentaje para los adultos mayores es de 81.3 %, mientras que para los adultos de 30 a 59

años corresponde el 56.2 %.

Tipos de discapacidad	Grupos de edad			
	de 0 a 14	15 a 29	30 a 59	60 +
Caminar, subir o bajar usando sus piernas	36.2	32.1	56.2	81.3
Ver (aunque use lentes)	26.9	44.6	58.2	67.2
Mover o usar sus brazos o manos	14.1	18.2	28.5	42.7
Aprender, recordar o concentrarse	40.8	31.5	32.1	44.6
Escuchar (aunque use aparato auditivo)	13.4	18.5	24.2	46.9
Banarse, vestirse o comer	37.4	16.4	14.5	29.3
Hablar o comunicarse	45.6	28.5	13.4	14.0
Problemas emocionales o mentales	26.6	28.0	20.1	16.3

Figura 1.2: Porcentaje de población con discapacidad, por tipo de discapacidad según grupos de edad en 2014. Fuente: ENID 2014.

De los datos anteriores se puede inferir que la disfunción motriz para caminar es muy frecuente entre las personas que presentan alguna discapacidad. Esta situación ha impulsado el desarrollo de sistemas para auxiliar a éstas personas. En este sentido, esta tesis se une a estos esfuerzos y aborda el desarrollo de un exoesqueleto que apoye la rehabilitación de personas afectadas en el movimiento de sus miembros inferiores, es decir, sus piernas y sus pies. Enseguida se revisan los antecedentes y la clasificación de los exoesqueletos.

1.2. Antecedentes

Facilitar el desplazamiento de personas y objetos no es un problema nuevo, siempre se han buscado formas de hacerlo posible usando animales y vehículos. Además de utilizar las extremidades inferiores propias de la persona, también se han desarrollado un gran número de mecanismos y máquinas como auxiliares para desplazamientos y manipulación de diversas cargas, sentando la base del desarrollo de la tecnología actual para apoyar a personas con discapacidad.

Parte de la historia de los diversos mecanismos de desplazamiento la describe ampliamente Barrientos [Barrientos07], desde los primeros mecanismos de los que se tiene registro, desarrollados por Arquímedes (287-212 a.C), como catapultas y brazos mecáni-

cos accionados por palancas y poleas empleados en la guerra; los dispositivos mecánicos de Leonardo Da Vinci; diferentes autómatas construidos en distintas épocas y mecanismos de los siglos XVII y XVIII desarrollados en su mayoría por relojeros, con algunas características similares a los robots actuales. Para finales del siglo XIX se incrementó de manera considerable el desarrollo de sistemas caminantes trayendo consigo las primeras patentes [Núñez-Altamirano16].

En la actualidad los robots manipuladores son una parte esencial de la industria. Se utilizan en diferentes procesos bajo condiciones controladas, además de que se emplean también robots móviles en ambientes dinámicos para realizar un gran número de tareas de distintos tipos. También se encuentran en desarrollo robots para las áreas doméstica y médica, equipados con cámaras y sensores para dar seguimiento a los signos vitales de uno o varios pacientes, además de prototipos de robots de compañía y de servicio. También se han desarrollado juguetes robóticos como Paro, una foca de peluche sensible al tacto que exhibe comportamientos inspirados en conductas animales [Sabanovic13]. Estos robots han sido ampliamente aceptados entre adultos mayores en Japón. También se han utilizado robots móviles, algunos antropomorfos, para ayudar a niños autistas y con problemas de aprendizaje [Siegwart11, Dudek10, Petric14, Sabanovic13].

Una rama importante de la robótica consiste en el desarrollo de equipos que ayuden a personas con discapacidad, mejorando la calidad de vida de estas personas. Algunos sistemas robóticos funcionan como auxiliares en la manipulación de objetos, o equipos para facilitar el desplazamiento de las personas con discapacidad motriz. Los equipos más comunes son los sistemas que utilizan ruedas para desplazarse, como las sillas robóticas; algunas son sillas móviles controladas por la persona y otras son equipadas con sensores y actuadores para analizar el entorno, tomar decisiones y realizar los movimientos adecuados [Stefanov04].

En los distintos tipos de robots que se han diseñado y desarrollado, los exoesqueletos son aquellos que guardan una interacción cognitiva y física más directa con el ser humano [Godoy17]. A principios de los años 60, el Departamento de Defensa de los Estados Unidos de América mostró interés en el desarrollo de mecanismos que funcionaran como armaduras para proteger a los soldados. Durante ese mismo período, en el Laboratorio de

Aeronáutica de Cornell se comenzó a manejar el concepto de amplificadores de manipulación para los hombres. Durante los años de 1966 a 1971, en la compañía General Electric, se desarrolló el proyecto *Hardiman* en el cual se diseñaron dos exoesqueletos con una configuración maestro/esclavo; uno de éstos utilizaba la fuerza motriz humana y otro era impulsado por energía hidráulica [Pons08]. Esto marcó el inicio del diseño y desarrollo de los exoesqueletos, dando como resultado diversas ramas de investigación, las cuales se describen en la siguiente sección.

1.3. Clasificación y aplicaciones de los exoesqueletos

Una clasificación de los robots que pueden ser utilizados en o sobre el cuerpo humano con fines de rehabilitación motriz los divide en dos categorías, según la función que desempeñan [Chiang13]. La primera categoría son los robots protésicos: elementos electro-mecánicos que sustituyen extremidades que han sido amputadas, con la finalidad de que este dispositivo cumpla la función natural del miembro perdido [Pons08]. La segunda categoría son los robots ortopédicos o exoesqueletos: estructuras mecánicas que actúan sobre las extremidades de la persona.

En la investigación y el desarrollo de los exoesqueletos destacan dos ramas principales: una enfocada en el desarrollo de exoesqueletos que incrementan las capacidades físicas de los usuarios y otra enfocada principalmente a la rehabilitación y recuperación de la movilidad de las personas [Lopez14]. En la primera rama se encuentran las armaduras para el manejo de materiales radioactivos o peligrosos [Anam12, Fontana14, Lo12] o los exoesqueletos para aumentar la fuerza y el desempeño del usuario, facilitándole adaptarse a situaciones de grandes esfuerzos en ambientes hostiles, industriales o militares [Asbeck14, Dollar08, Pons08, Low06, Yu14]. En la segunda rama se ubican los exoesqueletos de rehabilitación y recuperación de movilidad, los cuales han tomado relevancia debido al rápido incremento en el número de personas que sufren trastornos de movimiento, ya sea por accidente, enfermedad o nacimiento. Esto ha promovido el desarrollo de sistemas que permitan a las personas con discapacidad motriz incrementar sus posibilidades de mejorar, recuperando la movilidad en forma parcial o total [Accoto14, Casolo08].

Los exoesqueletos se pueden clasificar de acuerdo a la zona o extremidades a la cual se acoplará el sistema, pudiendo ser de extremidades superiores (hombro, brazo, codo), de extremidades inferiores (cadera, muslo, rodilla, pierna, pie) o de cuerpo completo [Fontana14]. Los exoesqueletos de extremidades inferiores se aplican principalmente en tres sectores: rehabilitación, asistencia en locomoción humana y aumento de la fuerza. Los exoesqueletos de cuerpo completo, son más complejos debido a que intervienen todas las extremidades de la persona [Pons08, Chen16].

Los exoesqueletos para extremidades inferiores tienen como base de movimiento el principio biomecánico de las piernas, lo que ha llevado a una intensa investigación acerca de los problemas biomecánicos en la sección inferior del cuerpo humano, dando lugar a exoesqueletos antropomorfos; es decir, exoesqueletos que siguen la estructura del cuerpo humano [Lopez14]: cadera, muslo, rodilla, pierna y pie. También se han utilizado combinaciones de elementos con apariencia distinta a la estructura del cuerpo, denominados exoesqueletos no antropomorfos [Accoto14].

En los exoesqueletos se han utilizados actuadores neumáticos [Galle13], hidráulicos, magnéticos y eléctricos, siendo éstos últimos los más empleados [Casolo08]. Estos actuadores apoyan el movimiento del cuerpo originado por los músculos de la persona.

De acuerdo al tipo de control que utiliza un exoesqueleto, se tienen dos diseños de control básicos: 1) sistemas de control diseñados para el seguimiento de trayectorias basado en modelos, parámetros físicos o jerarquías [Lopez14]; y 2) sistemas de control basados de acuerdo a su uso [Anam12, Chen16]. En los sistemas del primer tipo, el exoesqueleto guía las extremidades por un patrón geométrico definido matemáticamente, similar al que siguen las piernas al caminar. En los sistemas del segundo tipo, el sistema de control es diseñado de acuerdo a una estrategia de entrenamiento denominada AAN (del inglés “Assistance As Needed”, asistencia según sea necesaria); este sistema de control solo suministra la potencia mínima necesaria para que el paciente lleve a cabo las tareas de rehabilitación, analizando su rendimiento en tiempo real. Este tipo de control ANN fomenta en el paciente la interacción con el sistema [Cao14].

1.4. Contexto del desarrollo realizado

El Cuerpo Académico de “Robótica y Automatización” de la Universidad Michoacana de San Nicolás de Hidalgo integrado por los profesores Luis Eduardo Gamboa Guzmán, Moisés García Villanueva, Ignacio Juárez Campos y Leonardo Romero Muñoz, lleva a cabo un proyecto de desarrollo de un exoesqueleto no antropomorfo de miembros inferiores y en él participa el estudiante de doctorado en ingeniería mecánica Jorge Curiel Godoy y la autora de esta tesis.

El exoesqueleto utiliza una arquitectura mecánica novedosa basada en el mecanismo de Peaucellier–Lipkin (P–L). El mecanismo P–L original de un sólo grado de libertad se ha extendido con dos grados adicionales para adecuarse a las necesidades de utilización en el exoesqueleto. Para esta aplicación del mecanismo P–L de 3 grados de libertad a un exoesqueleto de miembros inferiores se ha solicitado una patente y actualmente está en trámite.

Una fotografía del exoesqueleto en su estado actual se muestra en la Figura 1.3.



Figura 1.3: Exoesqueleto en su estado actual.

1.5. Objetivo de la Tesis

El objetivo de esta tesis es:

- **Instrumentación y control de un exoesqueleto basado en el mecanismo extendido de P–L de tres grados de libertad.**

Este objetivo comprende los siguientes objetivos particulares:

1. Selección de los actuadores del exoesqueleto. Especificación de los requerimientos eléctricos de los actuadores y de las fuentes de voltaje de alimentación.
2. Especificaciones de control de los actuadores del exoesqueleto.
3. Desarrollo de servomotores de alto par.
4. Desarrollo de una estación de control centralizado para la generación de trayectorias de caminado.

1.6. Justificación

El problema de la discapacidad en México, analizado anteriormente, requiere del diseño y desarrollo de sistemas que contribuyan a mejorar la calidad de vida de este sector de la población, de forma eficaz y a un precio accesible.

Por otro lado, los exoesqueletos han demostrado ser herramientas muy útiles para apoyar a las personas que tienen alguna discapacidad, mejorando la calidad de vida tanto en quien la padece físicamente, como también de su familia y su entorno.

Esta tesis va orientada precisamente hacia el desarrollo de un exoesqueleto novedoso, de bajo costo, que tiene su aplicación en la rehabilitación de personas con discapacidad en sus miembros inferiores.

En el proyecto de desarrollo del exoesqueleto, está involucrado el trabajo de investigación doctoral del estudiante Jorge Curiel Godoy, por la parte mecánica; y de la autora de esta tesis, en la parte eléctrica, electrónica y de control.

1.7. Logros Alcanzados

- Se integraron adecuadamente todos los elementos del exoesqueleto.
- El servomotor desarrollado funciona adecuadamente.
- Se logró que el exoesqueleto genere las trayectorias de movimiento de las piernas que le permita caminar.

1.8. Organización de la tesis

El capítulo 2 aborda el mecanismo de Peaucellier–Lipkin extendido a tres grados de libertad, los elementos que lo componen, la cinemática directa del mecanismo y las posibilidades del movimiento que se pueden lograr.

El capítulo 3 presenta los servomotores comerciales lineales y rotaciones utilizados en el exoesqueleto, así como sus requerimientos de energía y control.

El capítulo 4 describe la implementación del servomotor rotacional de alto par, describiendo los elementos que lo componen y su implementación de control usando un encoder absoluto de 10 bits.

El capítulo 5 aborda la estación de generación de trayectorias de caminado. Se describe el microcontrolador usado, las señales de entrada y salida, y los algoritmos utilizados.

En el capítulo 6 se describen las pruebas realizadas al exoesqueleto y se presentan los resultados obtenidos.

El capítulo 7 presenta las conclusiones del trabajo y enumera varias líneas en las cuales se puede ampliar la funcionalidad del exoesqueleto.

Capítulo 2

Mecanismo Peaucellier-Lipkin extendido

El exoesqueleto desarrollado utiliza el mecanismo Peaucellier-Lipkin extendido a 3 grados de libertad para poder desplazarse. En este capítulo se describe el mecanismo P-L original, de un grado de libertad, inventado en 1869. Se describen los dos grados adicionales de libertad que se agregan al mecanismo, se analiza su funcionamiento y las implicaciones que estos dos nuevos grados de libertad tienen en la generación de trayectorias. Se aborda la cinemática directa del mecanismo P-L extendido, para determinar las posibles trayectorias del mecanismo.

2.1. Mecanismo de P-L

El mecanismo de Peaucellier-Lipkin (P-L), mostrado en la Figura 2.1, es un mecanismo de línea recta exacta que tiene 8 barras y 6 puntos de unión o vértices. Este mecanismo fue inventado en 1869 por el ingeniero francés de la milicia Charles Nicolas Peaucellier, pero años más tarde el ruso Yom Tov Lipman Lipkin inventó de forma independiente el mismo mecanismo, razón por la cual el nombre del mecanismo hace referencia a ambos autores[Luquin15]. Las 8 barras o eslabones que forman el mecanismo P-L están articuladas entre sí en los puntos A , B , C , D , E y F . En lo sucesivo, una barra que une

dos puntos, se denotará juntando las letras de los puntos involucrados; y la longitud de una barra se denotará por la letra L con la barra correspondiente como subíndice. Por ejemplo, L_{AB} se refiere a la longitud de la barra AB que une los puntos A y B . El punto F es el punto de salida del mecanismo, el cual describirá una trayectoria recta vertical si se cumplen las siguientes restricciones:

$$L_{AB} = L_{BC}, \quad L_{CD} = L_{CE} = L_{DF} = L_{EF}, \quad L_{AD} = L_{AE} \quad (2.1)$$

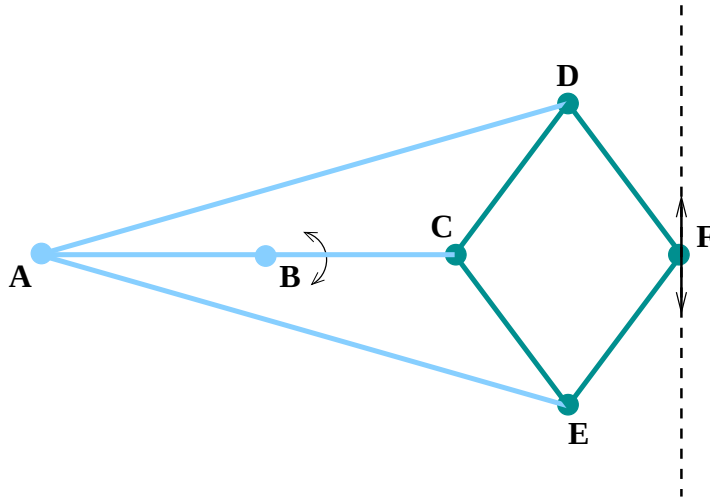


Figura 2.1: Mecanismo Peaucellier-Lipkin de línea recta exacta.

En el mecanismo P-L la barra AB es fija, es decir, su posición y orientación no cambian en ningún momento. En el punto o articulación B el mecanismo P-L incluye un actuador rotacional (indicado por una flecha en la Figura 2.1), que puede variar la inclinación de la barra que une al punto B con el punto C . Al variar este ángulo, el punto F describirá un movimiento en línea recta hacia arriba o hacia abajo. La longitud de la recta será limitada por las longitudes de las barras.

Este mecanismo tiene un único grado de libertad, la rotación de la barra BC respecto al punto B . Cuando la barra BC gira un ángulo positivo (hacia arriba), el punto F empieza a subir, siguiendo la línea recta vertical; por el contrario, cuando la barra BC gira un ángulo negativo (hacia abajo), el punto F empieza a bajar siguiendo también la recta vertical.

Para que el mecanismo sea capaz de realizar movimientos más complejos, como líneas rectas o curvas cóncavas o convexas, se agregaron dos grados de libertad adicionales, como se describe en la siguiente sección.

2.2. Mecanismo de P–L extendido a 3 grados de libertad

En la tesis de doctorado de Diego Alfredo Núñez Altamirano [Núñez-Altamirano16] realizada en la Facultad de Ingeniería Mecánica de la Universidad Michoacana en 2016, se explora la posibilidad de agregar dos nuevos grados de libertad al mecanismo de P–L, para poder tener la posibilidad de generar tanto trayectorias de línea recta como trayectorias curvas cóncavas o convexas. En su tesis doctoral este mecanismo P–L de 3 grados de libertad fue aplicado en vehículos múltipodos, en particular para que vehículos con 6 patas pudieran seguir trayectorias rectas o realizar giros a un lado o al otro.

En el mecanismo extendido se incorporan dos actuadores lineales: uno para poder modificar la longitud de la barra AB y otro para poder modificar la longitud de la barra BC . Ambos actuadores pueden operar en forma independiente.

Al cambiar las longitudes de los eslabones AB y BC se tienen los siguientes casos:

1. $L_{AB} = L_{BC} = K_1$

Este caso corresponde al mecanismo P–L original. La trayectoria de salida (punto F) sigue una línea recta exacta.

2. $L_{AB} < L_{BC}$

Para este caso la trayectoria de F sigue un arco circular cóncavo respecto a la articulación A , como se observa en la Figura 2.2, con centro de rotación a la izquierda de F y alineado con la recta del eslabón AB .

3. $L_{AB} > L_{BC}$

Cuando se cumple esta relación, la articulación F dibuja una curva convexa respecto a la articulación A , como se muestra en la Figura 2.3, con centro a la derecha de F , alineado también con la recta del eslabón AB .

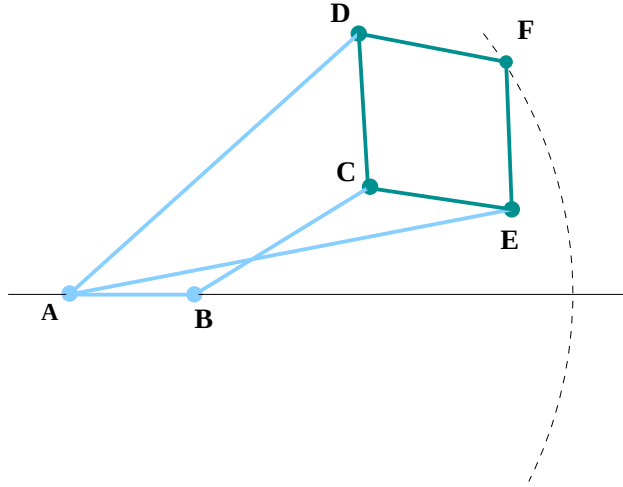


Figura 2.2: Caso 2: $L_{AB} < L_{BC}$. Mecanismo P-L describiendo un arco cóncavo.

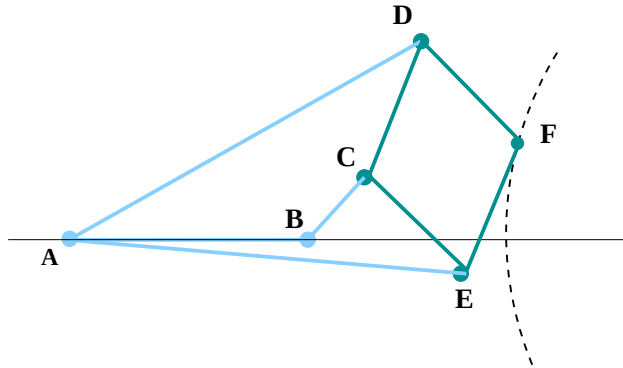


Figura 2.3: Caso 3: $L_{AB} > L_{BC}$. Mecanismo P-L describiendo un arco convexo.

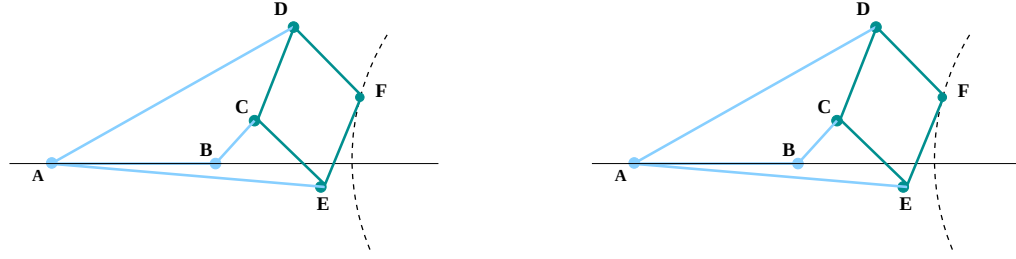
4. $L_{AB} = L_{BC} = K_2$ donde $K_2 > K_1$

Para esta condición la trayectoria de salida es una línea recta, como en el caso 1, pero la distancia del punto A a la línea recta del punto F es menor, como se observa en la Figura 2.4a.

5. $L_{AB} = L_{BC} = K_3$ donde $K_3 < K_1$

En este caso la trayectoria de salida sigue siendo recta, pero el punto F alcanza una distancia mayor respecto al origen, como se muestra en la Figura 2.4b.

Una vez mostrados los distintos casos que se pueden presentar en la operación del



(a) Caso 4: $L_{AB} = L_{BC} = K_2$ donde $K_2 > K_1$ (b) Caso 4: $L_{AB} = L_{BC} = K_2$ donde $K_2 < K_1$

Figura 2.4: Casos 4 y 5: Trayectorias rectas a diferentes distancias de A.

mecanismo de P–L extendido a 3 grados de libertad, enseguida se realiza un estudio de la cinemática directa de dicho mecanismo; es decir el comportamiento del punto de salida F en función del ángulo de rotación del eslabón BC y las longitudes de los eslabones AB y BC .

2.3. Cinemática Directa del Mecanismo Peaucellier-Lipkin extendido a 3 GDL

El análisis de la cinemática directa del mecanismo permite conocer el comportamiento de las articulaciones y eslabones del mecanismo cuando se encuentra en movimiento, particularmente la posición del punto de salida F , partiendo de los valores que puede tomar el ángulo de rotación θ del eslabón BC . Fijemos un sistemas de referencia X, Y , con centro en el punto A del mecanismo de P–L, como se muestra en la Figura 2.5. El ángulo θ se refiere al ángulo del eslabón BC con respecto al eje X positivo y es cero para la configuración inicial mostrada en la Figura 2.5.

Por simplicidad las longitudes de las barras se denotarán por una letra minúscula siguiendo la siguiente convención:

$$a = L_{AB}, \quad b = L_{BC}, \quad c = L_{CD} = L_{CE} = L_{DF} = L_{EF}, \quad d = L_{AD} = L_{AE} \quad (2.2)$$

Si el eslabón AB rota un ángulo θ positivo, la situación del mecanismo se muestra en la Figura 2.6.

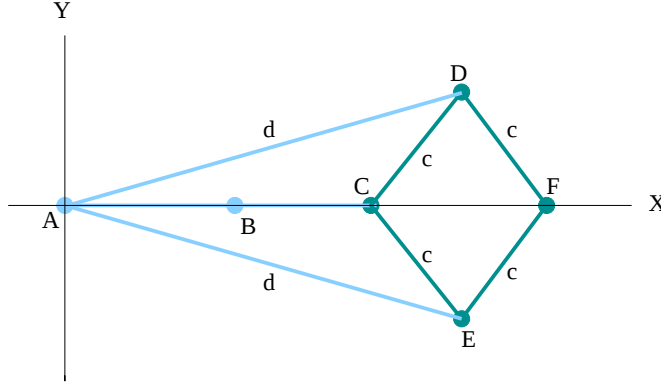


Figura 2.5: Mecanismo P-L en un plano, con origen en el punto A.

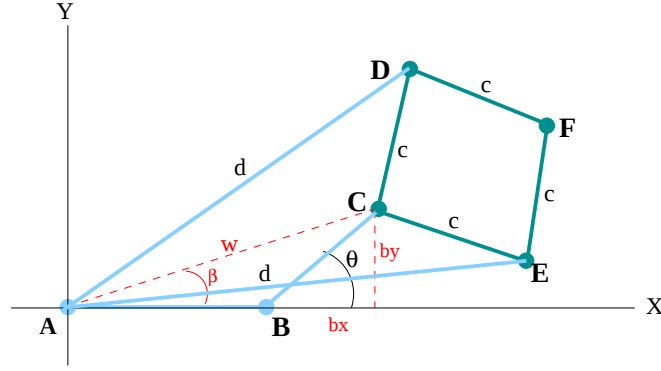


Figura 2.6: Mecanismo P-L cuando la barra BC gira un ángulo θ .

En adelante, la posición de cada punto se denotará de la forma A_x, A_y . De este modo tenemos que los puntos A y B no cambian de posición y tienen coordenadas:

$$A_x = 0, \quad A_y = 0, \quad B_x = a, \quad B_y = 0 \quad (2.3)$$

Para el triángulo (B, C, C_x) , dibujado con línea punteada en la Figura 2.6, tenemos:

$$bx = b \cos(\theta), \quad by = b \sin(\theta)$$

y como $C_y = by$ entonces la posición del punto C está dada por las coordenadas:

$$C_x = a + b \cos(\theta), \quad C_y = b \sin(\theta) \quad (2.4)$$

Cuando el sistema rota, además de los triángulos que conforman los diferentes eslabones, se dibuja uno adicional, el triángulo (A, C, C_x) , como se muestra en la Figura

2.6. Sea w el lado de este triángulo que une el punto A y el punto C y β el ángulo que forma w con respecto al eje X positivo. La distancia w y el ángulo β están dados por:

$$w = \sqrt{C_x^2 + C_y^2}, \quad \beta = \arctan\left(\frac{C_y}{C_x}\right) \quad (2.5)$$

Para cada valor de θ , los puntos de las articulaciones C , D , E y F se mueven ahora en un sistema de referencia X', Y' , que rota un ángulo β , como se observa en la Figura 2.7.

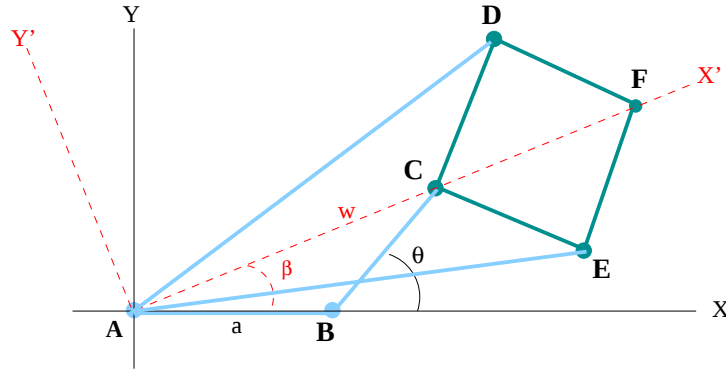


Figura 2.7: Sistema de referencia X', Y' alineado al lado w .

Calculada la longitud de w y el ángulo β , el mecanismo tiene la configuración mostrada en la Figura 2.8, con respecto al sistema de referencia X', Y' .

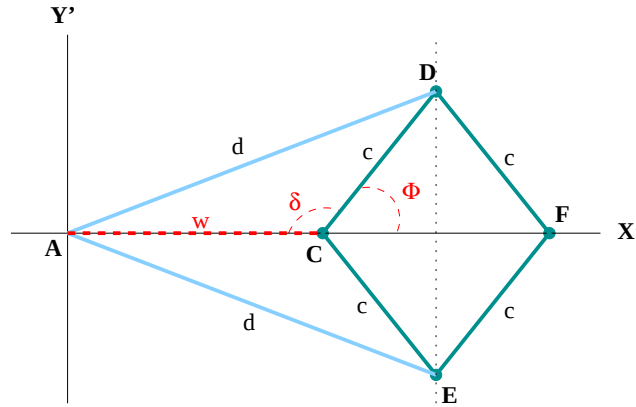


Figura 2.8: Mecanismo P-L en el plano X', Y' .

La posición de la articulación D está dada por:

$$D'_x = w + c \cos(\phi), \quad D'_y = c \sin(\phi) \quad (2.6)$$

donde el ángulo ϕ puede calcularse como sigue:

$$\phi = \pi - \delta \quad (2.7)$$

y el ángulo δ se calcula mediante la ley de cosenos de la forma:

$$\begin{aligned} d^2 &= w^2 + c^2 - 2cw \cos(\delta) \\ \cos(\delta) &= \frac{w^2 + c^2 - d^2}{2cw} \\ \delta &= \arccos\left(\frac{w^2 + c^2 - d^2}{2cw}\right) \end{aligned} \quad (2.8)$$

Puesto que el polígono formado por los puntos (C, D, F, E) es simétrico con respecto al eje X' , la posición del punto E está dada por:

$$E'_x = D'_x, \quad E'_y = -D'_y \quad (2.9)$$

Por último, aprovechando que el polígono también es simétrico con respecto a la línea que une los puntos D y E , las coordenadas del punto de salida F son:

$$F'_x = w + 2c \cos(\phi), \quad F'_y = 0 \quad (2.10)$$

Ahora, siendo que el sistema de coordenadas X', Y' tiene una rotación β con respecto al plano X, Y original, se aplica una matriz de rotación (dependiente del ángulo β) que multiplica al vector de coordenadas $[x', y']^t$ de los puntos D, E , y F , para obtener su posición en el plano original X, Y . Las coordenadas $[x, y]^t$ de los puntos se obtienen de la siguiente forma:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (2.11)$$

Una vez obtenidos estos puntos, se tienen las posiciones de cada una de las articulaciones del mecanismo. Las posiciones dependen de los valores de a, b, c, d y el ángulo θ .

Enseguida se presentan los resultados que predice el modelo matemático de la cinemática directa del mecanismo.

2.4. Experimentos con el modelo matemático

Para observar el funcionamiento del mecanismo, se escribió una función en Matlab [MATLAB10] llamada exoesqueleto.m, la cual se presenta a continuación:

```
function [Fx Fy] = exoesqueleto (a,b,c,d,theta,grafica)
% a = 1; b = 1; c = 1.2; d = 3;           % parametros
BordeGraf = 1;                           % Borde gráfica
Ax = 0;      Ay = 0;                     % punto A
Bx = a;      By = 0;                     % punto B
Cx = a + b * cos(theta); Cy = b * sin(theta); % punto C
w = sqrt(Cx^2 + Cy^2);
beta = atan2(Cy, Cx);
delta = acos((w^2 + c^2 - d^2)/(2*w * c));
Dpx = w + c * cos(pi - delta);           % D'x
Dpy = c * sin(pi - delta);              % D'y
Epx = Dpx;                              % E'x
Epy = - Dpy;                            % E'y
Fpx = w + 2 * c * cos(pi - delta);       % F'x
Fpy = 0;                                % F'y
Mrot = [cos(beta), - sin(beta);
        sin(beta), cos(beta)];           % Matriz rotación
D = Mrot * [Dpx; Dpy];
Dx = D(1); Dy = D(2);

E = Mrot * [Epx; Epy];
Ex = E(1); Ey = E(2);

F = Mrot * [Fpx; Fpy];
Fx = F(1); Fy = F(2);

X = [Ax, Bx, Cx, Dx, Fx, Ex, Cx, Ex, Ax, Dx];
Y = [Ay, By, Cy, Dy, Fy, Ey, Cy, Ey, Ay, Dy];
```

```

if grafica == 1
    plot(X,Y)
    axis square
    grid on
    hold on
    plot(X, Y, ' .', 'MarkerSize',20);
    Xmin = -Fx - BordeGraf; Xmax = Fx + BordeGraf;
    Ymin = -Fx - BordeGraf; Ymax = Fx + BordeGraf;
    x = [Xmin, Xmax, Xmax, Xmin, Xmin];
    y = [Ymin, Ymin, Ymax, Ymax, Ymin];
    plot(x,y);          % Forza la misma escala en X y Y
end
end

```

En Matlab se puede llamar a la función exoesqueleto de la forma:

```

>> theta=0; grafica=1; a=1; b=1; c=1.2; d=3;
>> [Fx Fy] = exoesqueleto(a,b,c,d,theta,grafica)

```

En este caso se dibuja la configuración del mecanismo P-L con los valores de a , b , c y d contenidos en la función exoesqueleto.m y el ángulo θ . La gráfica mostrada se muestra en la Figura 2.9. Si no se desea graficar el mecanismo y solamente obtener los valores del punto F en las variables F_x y F_y , entonces se puede asignar a la variable grafica el valor de 0.

Para observar la trayectoria que sigue el mecanismo cuando el ángulo está variando, se escribió una función en Matlab llamada grafica_exo.m, la cual se presenta a continuación:

```

function grafica_exo(a,b,c,d,inicio, incremento, fin, angulo_mecanismo)
for ang = [inicio:incremento:fin]
    [Fx, Fy] = exoesqueleto(a,b,c,d,ang,0);
    hold on;
    plot([Fx], [Fy], ' .');
end
[Fx Fy] = exoesqueleto(a,b,c,d,angulo_mecanismo,1);
end

```

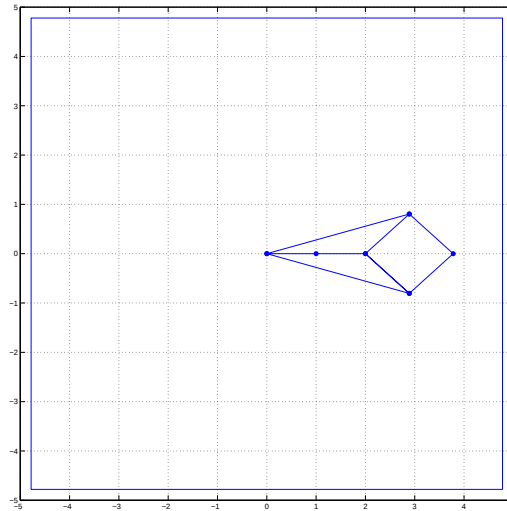


Figura 2.9: Gráfica obtenida por Matlab.

En Matlab se puede llamar a la función `grafica_exo` de la forma:

```
>> angulo_inicio=-pi/4;incremento=0.05;angulo_final=pi/4;angulo_mecanismo=pi/4;
>> a = 1; b = 1; c = 1.2; d = 3;
>> grafica_exo(a,b,c,d,angulo_inicio,incremento,angulo_final,angulo_mecanismo)
```

En este caso, se dibuja mediante puntos la trayectoria que sigue el punto F del mecanismo, desde el valor de la variable `angulo_inicio` hasta el valor de la variable `angulo_final`, en incrementos definidos por la variable `incremento`. El mecanismo se dibuja en el ángulo especificado por la variable `angulo_mecanismo`. La gráfica que se presenta en Matlab se muestra en la Figura 2.10, donde se puede observar claramente que el punto F sigue una trayectoria en línea recta vertical.

2.4.1. Experimentos variando los parámetros a y b

Ahora procederemos a realizar pruebas con diferentes longitudes de los eslabones a y b para ver las posibles trayectorias del punto F del mecanismo. Para observar con claridad la modificación de la trayectoria al variar los parámetros, se escribió una función en Matlab llamada `grafica2_exo.m`, la cual se presenta a continuación:

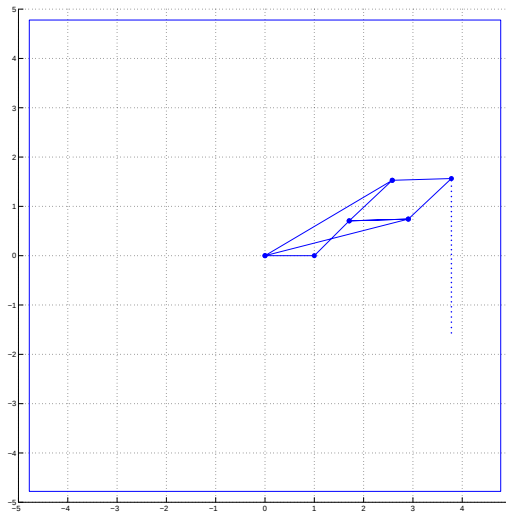


Figura 2.10: Gráfica de la trayectoria del punto F obtenida por Matlab.

```
function grafica2_exo(a1,a2,b1,b2,c,d,inicio,inc,fin,angulo_mecanismo)
for ang = [inicio:incremento:fin]
    [Fx, Fy] = exoesqueleto(a1,b1,c,d,ang,0);
    hold on;
    plot([Fx], [Fy], ' .', 'MarkerSize', 3);
end
for ang = [inicio:incremento:fin]
    [Fx, Fy] = exoesqueleto(a2,b2,c,d,ang,0);
    hold on;
    plot([Fx], [Fy], ' .', 'MarkerSize', 10);
end
[Fx Fy] = exoesqueleto(a2,b2,c,d,angulo_mecanismo,1);
end
```

Esta función grafica el mecanismo con los parámetros $(a2, b2, c, d)$, así como la trayectoria del punto F con una marca de tamaño 10. También se dibuja la trayectoria del mecanismo con los parámetros iniciales $(a1, b1, c, d)$. En Matlab se puede llamar a la función `grafica2_exo` de la forma:

```
>> angulo_inicio=-pi/4; inc=0.05; angulo_final=pi/4; angulo_mecanismo=0;
>> a1=1; a2= 1.5; b1=1; b2=0.5; c=1.2; d=3;
>> grafica2_exo(a1,a2,b1,b2,c,d,angulo_inicio,inc,angulo_final,angulo_mecanismo)
```

En este caso, se cambió de valores $a = b$ a $a > b$ y se obtiene la gráfica de la Figura 2.11, donde se puede observar claramente que el punto F sigue una trayectoria convexa, en lugar de la línea recta vertical.

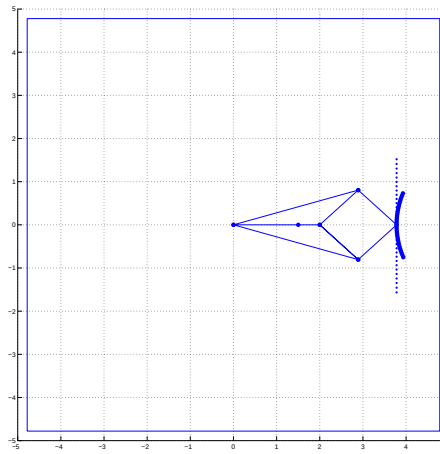


Figura 2.11: Gráfica de mecanismo para $a > b$.

Para observar el caso contrario, cuando $a < b$, se llama a la función `grafica2_exo` con los siguientes valores:

```
>> angulo_inicio=-pi/4; inc=0.05; angulo_final=pi/4; angulo_mecanismo=0;
>> a1=1; a2= 0.5; b1=1; b2=1.5; c=1.2; d=3;
>> grafica2_exo(a1,a2,b1,b2,c,d,angulo_inicio,inc,angulo_final,angulo_mecanismo)
```

En la gráfica obtenida, mostrada en la Figura 2.12, se puede observar que el punto F sigue una trayectoria cóncava, en lugar de la línea recta vertical.

Veamos ahora que pasa cuando $a = b$, pero disminuyen su valor, de 1 a 0.95, conservando los demás valores. En la gráfica obtenida, mostrada en la Figura 2.13, se puede observar que el punto F sigue una trayectoria recta vertical, pero más alejada del punto A

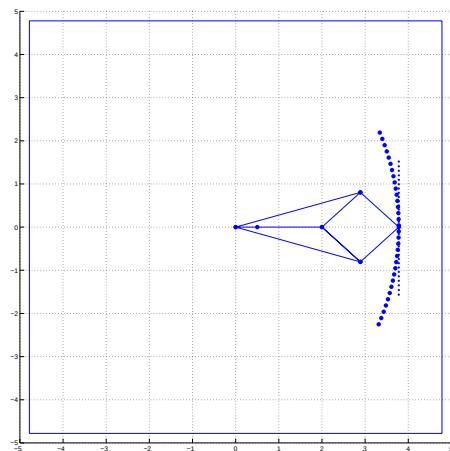


Figura 2.12: Gráfica de mecanismo para $a < b$.

que con los valores originales de a y b . El caso opuesto ocurre cuando a y b aumentan su valor. La gráfica mostrada en la Figura 2.14, corresponde a un valor de 1.05 para a y b .

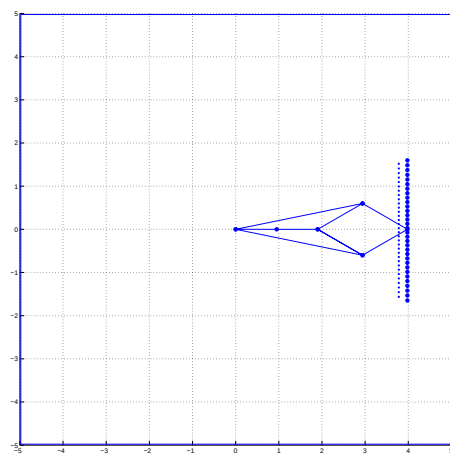


Figura 2.13: Gráfica de mecanismo para $a = 0.95, b = 0.95$.

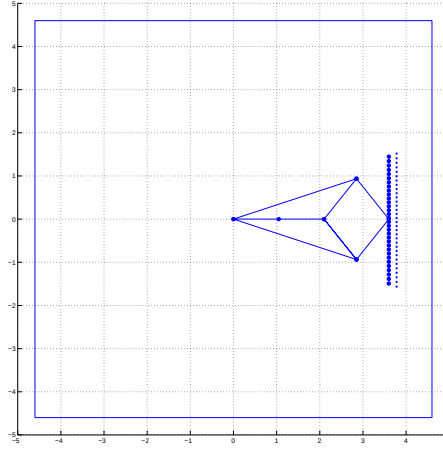


Figura 2.14: Gráfica de mecanismo para $a = 1.05, b = 1.05$.

2.5. Comentarios finales

El mecanismo de P-L extendido a tres grados de libertad (3 GDL) agrega la posibilidad de modificar las longitudes de las barras AB y BC , además del ángulo θ de la barra BC con el eje X positivo. Con estos dos grados de libertad adicionales se pueden generar trayectorias tanto rectas como curvas convexas y cóncavas. En la comprobación experimental, usando Matlab, se pudo comprobar las posibles trayectorias que se pueden generar.

En la aplicación de este mecanismo a un exoesqueleto, se dispone de un mecanismo de P-L de 3 GDL por cada pierna. El punto A del mecanismo se fija a la pelvis del exoesqueleto y se garantiza mecánicamente que la barra AB conserva su orientación. Al punto F se le agrega un zapato. Si lo rotamos 90 grados, en el sentido de las manecillas del reloj, veremos los dos mecanismos P-L de 3 GDL tal como se está utilizando en el exoesqueleto mostrado en la Figura 1.3 de la página 7.

De utilidad para el exoesqueleto son los comportamientos siguientes:

1. Posibilidad de trayectorias rectas. Cuando $a = b$, al modificar el ángulo θ el zapato de la pierna pueda desplazarse hacia adelante o hacia atrás.

2. Posibilidad de trayectorias cóncavas. Cuando $a > b$, la trayectoria cóncava permite que el zapato pueda tocar el piso sólo en los extremos del movimiento, mientras que en el resto de la trayectoria iría en el aire.
3. Posibilidad de levantar o bajar el zapato de una pierna. Cuando en una pierna $a = b$, pero disminuyen su valor, el zapato baja; por el contrario, cuando aumentan su valor, el zapato sube. Esta posibilidad de levantar y bajar el zapato se puede utilizar para el caminado.
4. La posibilidad de levantar o bajar el zapato de ambas piernas. Esta posibilidad habilita al exoesqueleto para adaptarse a la altura de las personas.

Capítulo 3

Elementos del exoesqueleto

El exoesqueleto está compuesto de dos mecanismos de Peaucellier-Lipkin (P-L) extendidos a tres grados de libertad, uno por cada pierna. En la Figura 3.1 se muestra una vista lateral del exoesqueleto en la que se observa uno de los dos mecanismos de P-L utilizados.

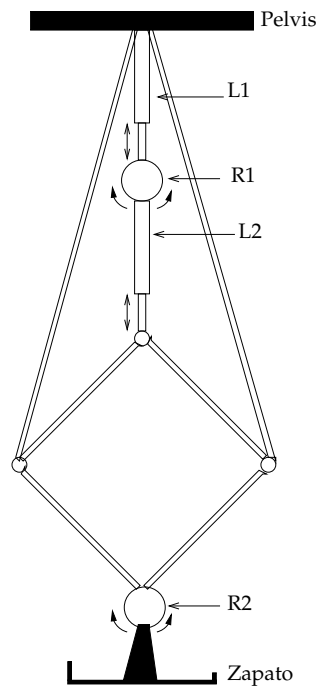


Figura 3.1: Vista lateral del exoesqueleto.

La pelvis está sujeta a un marco-soporte que la mantiene paralela al piso. El actuador lineal, marcado como L1 en la Figura 3.1, mantiene fija su orientación vertical pero puede modificar su longitud. El actuador rotacional R1 permite rotar el eslabón que contiene el segundo actuador lineal, L2. Finalmente el actuador rotacional R2 permite modificar la inclinación del zapato.

Debido a que el actuador rotacional R1 requiere un alto par y no fue posible encontrar uno adecuado en el mercado, se decidió desarrollar uno. El siguiente capítulo describe los detalles de construcción del servomotor desarrollado.

En este capítulo se describen los actuadores lineales L1, L2 y el actuador rotacional R2, sus características y su forma de control. También se presentan los componentes típicos de un servomotor y su forma de funcionamiento y control.

3.1. Actuadores lineales y rotacionales

Los servomotores lineales L1 y L2 son servomotores [HDLS-4-30-12v17] de la marca Servocity modelo HDLS-4-30-12V. La Figura 3.2 presenta una imagen de este servomotor y la Tabla 3.1 muestra sus principales características.



Figura 3.2: Servomotor lineal HDLS-4-30-12V marca Servocity 2017.

El servomotor rotacional R2 es un motor modelo Torxis Servo i00600 [Gea17], mostrado en la Figura 3.3. Las características principales de este servo se muestran en la Tabla 3.2.

Característica	Valor
Voltaje nominal	12v
Peso	40.2 oz
Señal PWM (completamente retraído)	2000 microsegundos
Señal PWM (completamente extendido)	1000 microsegundos
Amplitud de pulso	3-5V
Velocidad en vacío	0.30 pulgadas por segundo
Velocidad con carga máxima	0.20 pulgadas por segundo
Corriente (sin carga)	1A
Corriente (carga máxima)	4.5A
Engranajes	Tren de engranes metálicos y piñón de nylon
Ciclo de trabajo	25 % (25 % encendido, 75 % apagado)
Carrera	4 pulgadas
Empuje dinámico	180 libras
Relación de transmisión	40:1

Tabla 3.1: Datos del servomotor lineal HDLS-4-30-12V.

Característica	Valor
Voltaje nominal	12V
Corriente a plena carga	3A
Tiempo de viaje	90 grados en 1.5 segundos
Par (continuo)	1600 oz-in (11.3 N-m)
Par (pico)	3200 oz-in (22.6 N-m)
Tiempos	1500 microsegundos = 0 grados 500 microsegundos = -90 grados 2500 microsegundos = +90 grados 100 microsegundos = -135 grados 2900 microsegundos = +135 grados
Señal de entrada	RC PWM
Dimensiones	5.5 in X 3.9 in X 2.4 in
Reducción de engranes	1044:1

Tabla 3.2: Datos del servomotor rotacional del zapato Torxis Servo i00600.



Figura 3.3: Servomotor rotacional Torxis Servo i00600 marca Gearwux [Gea17].

3.2. Fuente de alimentación

Los 4 servomotores lineales L1 y L2, y los 2 motores rotacionales R2 de los zapatos, son alimentados con una fuente conmutada modelo DC12V40A mostrada en la Figura 3.4, disponible en <http://www.aliexpress.com>. Las principales características de la fuente son: voltaje de entrada de corriente alterna entre 85 y 265 volts, voltaje de salida 12V, corriente máxima de salida 40A, potencia de 480W, dimensiones de 21.5*11.3*5 cm y peso de 750g.



Figura 3.4: Fuente de alimentación conmutada DC12V40A

3.3. Funcionamiento de los servomotores

Los servomotores son sistemas eléctricos compuestos de un motor de CD, una caja de reducción de engranes y un circuito de control; y pueden ser del tipo rotacional o lineal.

3.3.1. Servomotores rotacionales

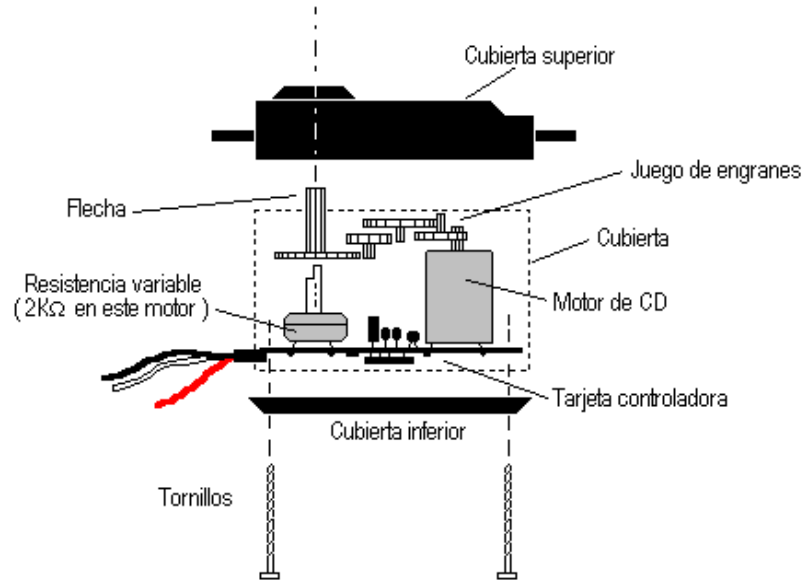


Figura 3.5: Componentes de un servomotor rotacional [ser17a].

La Figura 3.5 muestra los componentes típicos de un servomotor rotacional. El servomotor controla la posición angular de la flecha de acuerdo a la señal de entrada que recibe.

El motor eléctrico se acopla a un juego de engranes para hacer girar la flecha. La flecha también está acoplada a un potenciómetro, para medir el ángulo girado por la flecha. El circuito de control utiliza el potenciómetro para decidir si tiene que mover el motor en un sentido o en el otro y que cantidad. Este sistema permite mantener una posición angular de la flecha dentro del rango de operación del servomotor. La caja de engranes amplía el par del motor; el circuito electrónico se encarga de recibir una señal de entrada y traducirla en movimiento del motor.

Un servomotor rotacional normalmente tiene un rango de operación entre 0 y 180 grados, pero también hay servomotores que pueden posicionarse entre 0 y 360 grados.

Un servomotor pequeño como el de la Figura 3.5 tiene tres cables de entrada: dos para alimentación, normalmente 5v+ y referencia (Gnd) (cables rojo y negro respectiva-

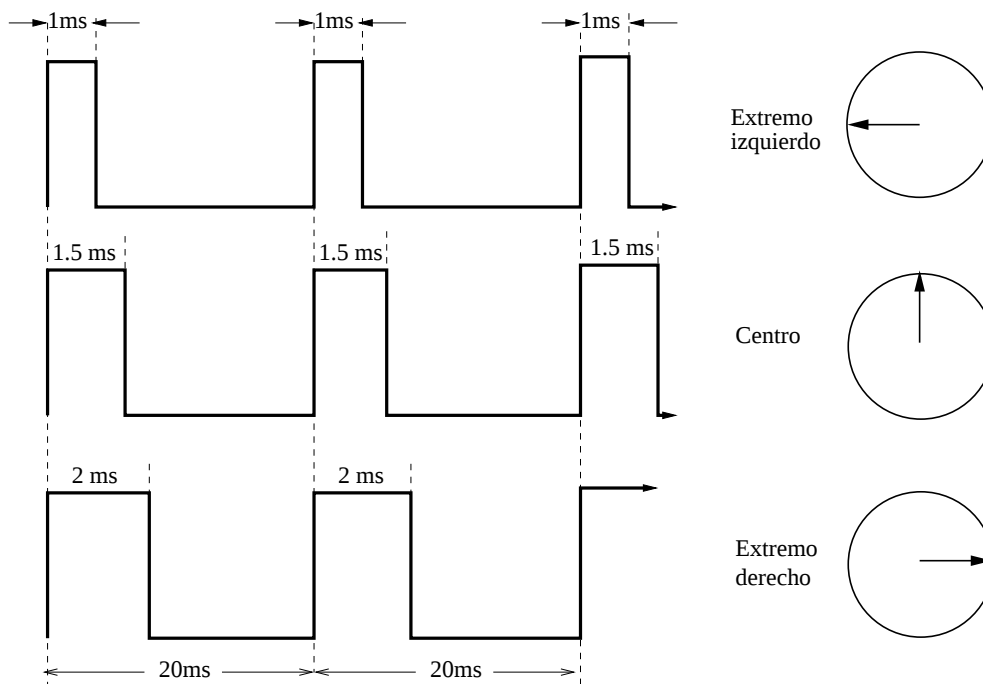


Figura 3.6: Señal PWM para controlar el servomotor.

mente), y uno de control (cable blanco). En los servomotores lineales y rotacionales del exoesqueleto la alimentación tiene un conector especial (cables rojo y negro), mientras que el conector de control tiene dos cables, uno negro y otro blanco, el de referencia (Gnd) y el de control.

La señal de control utiliza un esquema de modulación de la señal llamado modulación de ancho de pulso (PWM, por sus siglas en inglés, "Pulse Width Modulation"). En el esquema PWM se envía un tren de pulsos, típicamente 50 pulsos por segundo, y el ancho del pulso determina el ángulo de giro del servomotor. Típicamente el ancho del pulso varía entre 1 y 2 milisegundos.

La Figura 3.6 presenta trenes de pulsos con diferente duración del pulso para controlar un servo que gira 180 grados. Para la duración de 1ms el servomotor gira hasta su posición extrema izquierda, para la duración de 1.5ms el servomotor gira a su posición central y para la duración de 2ms el servomotor gira a su posición extrema derecha.

Otro ejemplo de movimiento de un servomotor se muestra en la Figura 3.7, en la

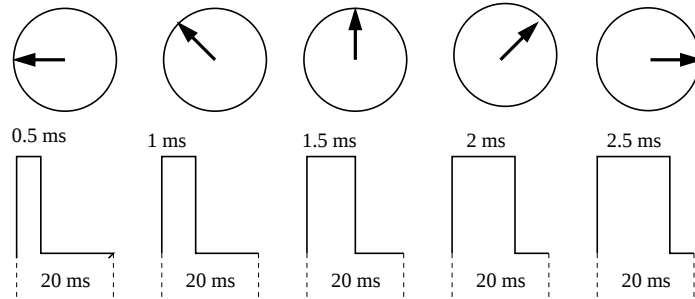


Figura 3.7: Posiciones de un servomotor para pulsos de 0.5 a 2.5 ms.

que se observa la posición de la flecha del servomotor para anchos de pulso que varían de 0.5 a 2.5 milisegundos. El desplazamiento angular es proporcional al ancho del pulso.

3.3.2. Servomotores lineales

Un servomotor lineal es un dispositivo que genera un movimiento rectilíneo [ser17b]. Internamente está formado por un motor eléctrico con un mecanismo reductor que consta por varios engranajes y un tornillo sinfín. El tornillo empuja un émbolo o vástago, que puede extenderse o retraerse en función del sentido de giro del motor. Los actuadores lineales tienen la ventaja de ejercer grandes fuerzas y grandes desplazamientos. Por ejemplo, pueden ejercer de 20 a 150kgf, con desplazamientos de 100 a 800mm. Como desventaja, los actuadores lineales tienen velocidades de desplazamiento lentas, típicamente del orden de 4-20 mm/s, por lo que el tiempo de extensión total puede llegar a ser de varios minutos. Existen actuadores lineales de diferentes tensiones de alimentación, siendo frecuentes modelos de 12V y 24V. El consumo de corriente es, en general, elevado, del orden de 3A a 5A en los modelos de 12V, y de 2A-3A en modelos de 24V.

El funcionamiento de los servomotores lineales es similar al rotacional, la señal de control que recibe determina la longitud que se extiende el émbolo del servomotor, desde la posición totalmente retraída hasta la posición totalmente extendida. La Figura 3.8 muestra los componentes típicos de un servomotor lineal.



Figura 3.8: Un servomotor lineal por dentro [ser17b].

3.4. Control de un servomotor utilizando un microcontrolador arduino

La señal de entrada para un servomotor puede generarse de muchas maneras, usando un circuito oscilador controlado por un potenciómetro o bien utilizar un microcontrolador o una computadora.

El microcontrolador Arduino puede generar señales de control hasta para 12 servomotores en la mayoría de los modelos de tarjetas Arduino [Ard17b]. Las Figuras 3.9 y 3.10 muestran un ejemplo de control de un pequeño servomotor utilizando una tarjeta arduino y un potenciómetro.

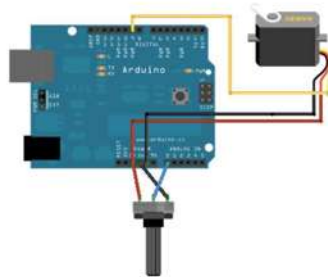


Figura 3.9: Control de un servomotor utilizando un microcontrolador arduino y un potenciómetro [Ard17b].

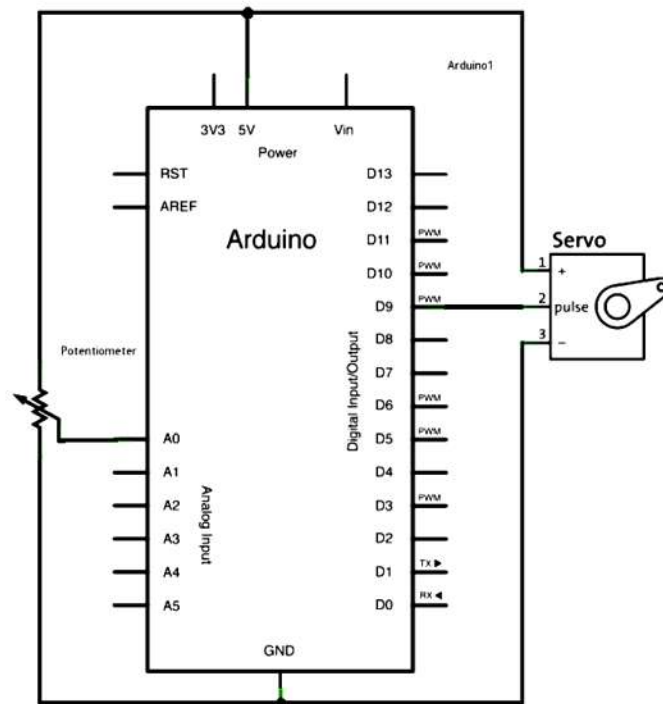


Figura 3.10: Diagrama de conexiones [Ard17b].

El siguiente código [Ard17b] realiza la tarea de control deseada: al girar el potenciómetro se cambia el ancho de pulso de la señal de control que genera el arduino y que recibe el servomotor.

```
/*
  Controlling a servo position using a potentiometer (variable resistor)
  by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
  modified on 8 Nov 2013
  by Scott Fitzgerald
  http://www.arduino.cc/en/Tutorial/Knob
  */
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
```

```
int val;          // variable to read the value from the analog pin

void setup() {
  myservo.attach(9);    // attaches the servo on pin 9 to the servo object
}

void loop() {
  val=analogRead(potpin); //reads the value of the potentiometer (value between 0 and 1023)
  val=map(val,0,1023,0,180);//scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);    //sets the servo position according to the scaled value
  delay(15);             //waits for the servo to get there
}
```

Como ejercicio, se implementó este control para poder mover los servomotores lineales y rotacionales mientras estuvo en la fase de construcción el exoesqueleto.

3.5. Comentarios finales

En este capítulo se ha revisado las características de los servomotores lineales y rotacionales que emplea el exoesqueleto, así como su funcionamiento.

Entender el control de los servomotores nos ayudará a abordar el siguiente capítulo, donde se describe la implementación de un servomotor rotacional de alto par y la señal de control que recibe el servomotor sigue el esquema de señal de control PWM indicado en este capítulo.

Capítulo 4

Implementación de un servomotor rotacional de alto par

En este capítulo se describe los componentes para integrar un servomotor de alto par para utilizarse en la articulación rotacional que gira la pierna. Fue necesario construirlo debido a que no se encontró un servomotor rotacional comercial con el par requerido. El más cercano fue el servo motor H54-100-S500-R Dynamixel Pro Smart, con un par de 21.142 Nm, 32.7 RPM, 24V, 5.93A, con un costo de USD \$2,590.00, disponible en <http://www.robotshop.com>.

El servomotor opera como un servomotor comercial y se puede conectar de la misma forma. Se programó para obedecer pulsos con amplitud entre 1000 y 2000 microsegundos.

4.1. Elementos del servomotor

El servomotor se integra con los siguientes componentes:

- Motor y su caja de engranes.
- Encoder absoluto.
- Driver del motor de CD.

- Fuentes de alimentación.
- Microcontrolador

4.1.1. Motor y caja de engranes

En la Figura 4.1a se muestra el motor seleccionado, un motor industrial de corriente directa, alta potencia y baja velocidad, de la marca china Sumotor, modelo GW114165. EL motor esta acoplado a un engrane tipo gusano y a una corona helicoidal, como se muestra en la Figura 4.1b.



(a) Motor de CD GW114165



(b) gusano y corona helicoidal.

Figura 4.1: Motor de CD Sumotor utilizado [motorGW11416517].

El motor de CD es de 24 volts, 15 amperes, 80 Nm, 17 RPM sin carga y 13 RPM bajo carga, con dimensiones 360x114x167mm y una masa de 9.2 Kg.

4.1.2. Encoder absoluto de 10 bits

Para conocer con precisión la rotación del motor fue seleccionado el encoder rotatorio absoluto marca Koyo modelo TRD-NA1024NW de 10 bits mostrado en la Figura 4.2. Los datos del encoder se pueden consultar en el manual del fabricante [TRD-NA1024NW17].

Las características del encoder se muestran en las Figuras 4.3 y 4.4, las cuales fueron tomadas del manual del encoder.

De la Figura 4.3 se observa que el encoder debe alimentarse con un voltaje entre 10.8 y 26.4V, con un consumo de corriente de 70 mA o inferior. La salida está codificada



Figura 4.2: Encoder absoluto de 10 bits [TRD-NA1024NW17].

Electrical Specifications

Model Number		TRD-NA□NW		TRD-NA□PW
Power Supply	Supply Voltage	10.8 to 26.4 V DC		←
	Allowable Ripple	3% rms or less		←
	Consumption Current ^{*4}	70 mA or lower (2,048 pulses: 100 mA or lower)		100 mA or lower
Output Code		Gray binary ^{*1}		←
Maximum Response Frequency		20 kHz ^{*2}		←
Precision		(360° / (Resolution x 2)) [*]		←
Rotation Direction		Clockwise rotation (CW) or counterclockwise rotation (CCW) ^{*3}		←
Output	Output Form	NPN open collector output		PNP open collector output
	Output Logic	Negative logic (Active low)		Positive logic (active high)
	Residual Voltage	0.4 V or lower	1.5 V or lower	0.5 V or lower
	Inflow-Outflow Current	16 mA	32 mA	30 mA
	Load Supply Voltage	30 V DC or lower		←
Rise / Fall Time ^{*5}		Not larger than 2.0 μs (Load resistance 1kΩ)		Not larger than 3.0 μs (Load resistance 1kΩ)

*1: Excess 38 gray code for 180 resolution, excess 76 gray code for 360 resolution, and excess 152 gray code for 720 resolution

*2: Electrical maximum number of revolutions = [(Maximum response frequency/Resolution) x 60]

Therefore, if the encoder rotates at a speed greater than the electrical maximum number of revolutions, the signals do not electrically follow.

*3: CW means clockwise seen from the shaft side. CCW means counterclockwise seen from the shaft side.

*4: No load

*5: Cable length 2 m, maximum load

Figura 4.3: Características del encoder.

en código gray y por esta razón en la sección 4.1.6 se revisa en detalle este código y su conversión a un código binario estándar. También las salidas del encoder son del tipo de colector abierto NPN, activas en bajo y con un voltaje de 30 volts o inferior; estos detalles son importantes para conectar estas salidas al microcontrolador que va a leer el encoder. De la Figura 4.4 podemos observar la configuración de colector abierto NPN para las salidas correspondientes a los bits 1 a 10 (para el encoder seleccionado), así como los códigos de color de cada cable que sale del encoder. El cable café es para la entrada positiva de voltaje (12 volts en este caso) y el cable azul es para la entrada negativa. Los 10 bits que proporciona el encoder pueden codificar una posición angular entre 0 y 1023 pasos por vuelta.

El conector del encoder tiene físicamente 12 entradas y un arreglo ligeramente

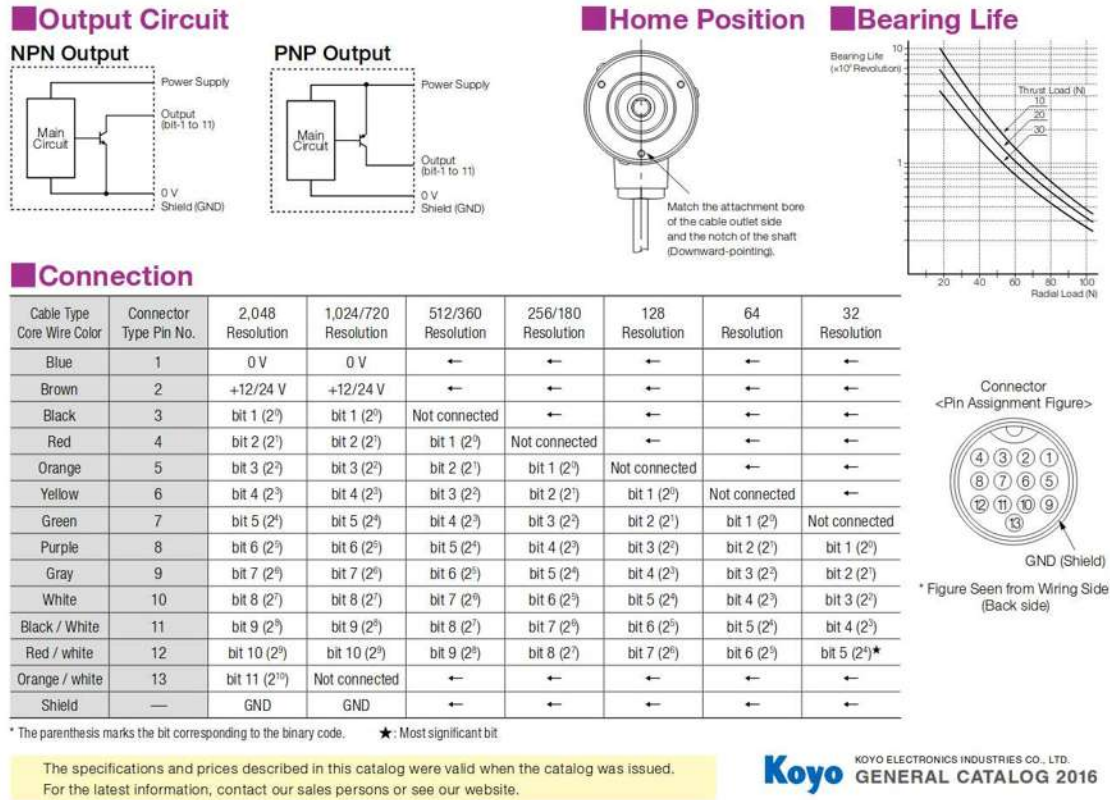


Figura 4.4: Características del encoder.

diferente al del manual, siendo la primera entrada el cable café, seguida del cable negro (el bit 1), el rojo (bit 2), hasta llegar al cable rojo-blanco (bit 10) y la última entrada es el cable azul. En la implementación se alimenta con 12 volts, al igual que el microcontrolador utilizado.

4.1.3. Driver del motor de CD

Para operar el motor de CD se eligió el driver VNH5019 de Pololu [VNH501917]. La Figura 4.5 muestra el driver y la forma como se puede operar. Este driver acepta entradas entre 5.5 y 24 V y puede entregar 12 A continuos (30 A pico). Trabaja con niveles lógicos entre 2.5 y 5 V y soporta control PWM (hasta 20 kHz).

Por medio de las entradas INA e INB se puede controlar la dirección del movimiento. Si INA está en alto e INB en bajo, el motor girará en una dirección y si INA está en

bajo e INB está en alto, el motor girará en dirección contraria.

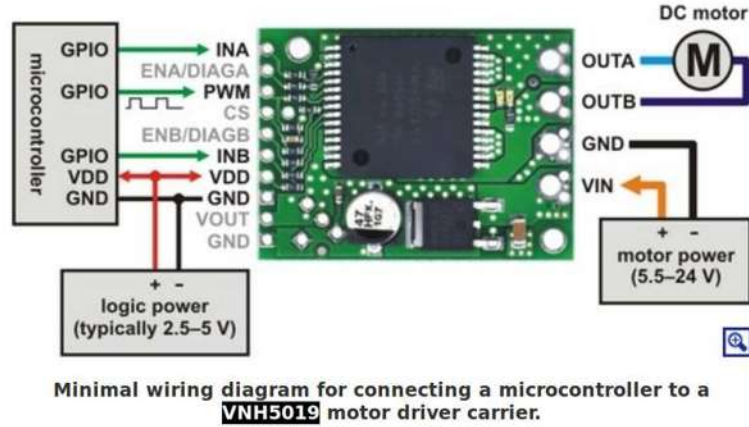


Figura 4.5: Driver de motor de CD Pololu VNH5019 [VNH501917].

4.1.4. Fuentes de alimentación

Para alimentar el motor de CD se emplea una fuente conmutada de 24 volts, 15 amperes. Por otro lado, para alimentar el encoder y el microcontrolador se utiliza la misma fuente de voltaje de 12 volts, 40 amperes que utiliza el exoesqueleto para la alimentación de los otros servomotores comerciales.

4.1.5. Microcontrolador

El microcontrolador requiere recibir 10 entradas digitales del encoder, dos salidas digitales INA e INB del driver del motor, una señal de salida con capacidad PWM y una entrada con capacidad de interrupción para leer la entrada del pulso de control del servomotor.

Se eligió el microcontrolador arduino nano, por ser pequeño y tener las capacidades antes mencionadas. Un poco más adelante se describe la utilización de este microcontrolador para leer el código gray que entrega el encoder, así como efectuar el control del servomotor. A continuación se revisa el código gray.

4.1.6. Código gray

Esta sección está basada en [cod17]. El código Gray, nombrado así en honor del investigador Frank Gray, es un sistema de numeración binario en el que dos palabras consecutivas difieren solamente en uno de sus dígitos.

Este código actualmente es usado para facilitar la corrección de errores en los sistemas de comunicaciones. La Tabla 4.1 muestra un ejemplo de un código gray, donde se puede apreciar que entre dígitos sucesivos sólo cambia un bit; y también esto ocurre entre el último número y el primero.

De esta forma, un lector puede leer fácilmente el valor que recibe, puesto que sólo cambia un bit en cada número consecutivo. En caso de no utilizarse el código gray, se tendría que tener alguna forma de sincronizar el dispositivo emisor con el receptor; por ejemplo utilizando ciertos intervalos de tiempo o una señal de sincronización especial.

Decimal	Gray	Binario
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

Tabla 4.1: Ejemplo de un código Gray de 3 bits.

Conversión de código binario a código Gray

Para convertir un número binario (en Base 2) a código Gray, simplemente se le aplica una operación O exclusiva con el mismo número desplazado un bit a la derecha, sin tener en cuenta el acarreo. La operación OR exclusiva (XOR en inglés) es un 1 cuando únicamente una de sus entradas es 1 y se representa por el símbolo $\oplus$. De esta manera tenemos que $0 \leftarrow 0 \oplus 0$, $1 \leftarrow 0 \oplus 1$, $1 \leftarrow 1 \oplus 0$ y $0 \leftarrow 1 \oplus 1$.

Por ejemplo, el número binario 1010 se convierte en el número 1111 en código gray:

100	
100	(operación XOR)

110	

Conversión de código Gray a código binario

Para convertir un código Gray a un código binario, se define un vector g conteniendo los dígitos en código Gray y otro vector b destinado a contener los dígitos en base 2. Sea g_n el dígito que se encuentra en el extremo izquierdo de la representación en código Gray. Sea b_n el dígito binario de mayor peso y que se encuentra en el extremo izquierdo en la representación en base 2. El algoritmo 1 presenta la conversión deseada.

Data: g en código gray ($g_n, \dots g_0$)

Result: b en código binario ($b_n, \dots b_0$),

$g_n = b_n$;

for $i \leftarrow n - 1$ **to** 0 **do**

 | $b_i = g_i \oplus b_{i+1}$

end

Algorithm 1: Algoritmo para conversión de código Gray a binario.

Por ejemplo, si $g = 110$ en código Gray, tenemos:

$$b_2 = g_2 = 1$$

$$b_1 = g_1 \oplus b_2 = 1 \oplus 1 = 0$$

$$b_0 = g_0 \oplus b_1 = 0 \oplus 0 = 0$$

Es decir, g equivale a código binario 100.

4.1.7. Lectura del encoder usando un microcontrolador

El siguiente código del microcontrolador arduino nano realiza la lectura del encoder absoluto.

```

// Propósito: Leer el encoder absoluto de 10 bits KOYO TRD-NA1024NW

#define LSB 4      // el bit0 (negro) está conectado al pin D4,
                  // el bit1 al D5, ... el bit9 (rojo-blanco) al D13
#define BITS 10    // Define la cantidad de bits (10)

void setup() {
    for (int i = 0; i < BITS; i++)      // Habilita la resistencia PULLUP
        pinMode(i + LSB, INPUT_PULLUP); // para las 10 entradas de bits
    Serial.begin(9600);
    Serial.print("Ready\n");
}

int i;
int sensor[BITS]; // valores reportados en código Gray
int bin[BITS];    // equivalente en binario

int conBinDec(int entrada[], int bits){
    int valor = 1, resultado = 0;
    for (i = 0; i < bits; i++){
        resultado += entrada[i] * valor;
        valor *= 2;
    }
    return resultado;
}

int decodificaGray(){
    bin[BITS-1] = sensor[BITS-1];
    for(i=BITS-2; i >= 0; i--){
        bin[i] = bin[i+1] ^ sensor[i];

        return conBinDec(bin, BITS);
    }
}

int leeEncoder(void){
    for (i = 0; i < BITS; i++)
        sensor[i] = !digitalRead(i+LSB);
}

```

```
    return decodificaGray();
}

void imprimeBits(int arreglo[], int bits){
    for (i = bits - 1; i >= 0; i--){
        Serial.print(arreglo[i]);
    }

    int valorBinario;
    int valorGray;
    void loop() {
        valorBinario = leeEncoder();
        imprimeBits(sensor,BITS);
        Serial.print(" (Valor Gray =");
        valorGray = conBinDec(sensor, BITS);
        Serial.print(valorGray);
        Serial.print(") Binario =");
        Serial.println(valorBinario);
        delay(500);
    }
}
```

En la función *setup* se inicializan las entradas D4, D5, D6, D7, D8, D9, D10, D11, D12 y D13 como entradas con la bandera INPUT_PULLUP, que habilita la conexión a 5V de la entrada usando una resistencia interna del microcontrolador, debido a que las salidas del encoder son colector abierto. La entrada D4 corresponde al bit 1 del encoder, D4 al bit 2 y así sucesivamente.

En la función *leeEncoder* se leen las entradas que vienen del encoder con la función *digitalRead* de la entrada digital y se colocan los valores en el arreglo *sensor*. Debido a que las señales del encoder son activas en bajo, es necesario utilizar el operador lógico ! colocado antes de la llamada a la función.

La función *decodificaGray* implementa el algoritmo 1 para convertir los valores del arreglo *sensor* al arreglo *bin* que contiene el equivalente binario. Antes de terminar la función se llama a la función *conBinDec* para convertir los valores contenidos en el arreglo *bin* a un entero.

Finalmente en la función *loop* del microcontrolador se lee el encoder y se escribe en el puerto serial los bits en código gray, los bits en código binario y su equivalente en decimal.

4.2. Implementación del control utilizando un microcontrolador arduino

En el siguiente código se implementa el control del servomotor. Se omiten las variables y funciones de la función *leeEncoder* explicada anteriormente.

Se utiliza la entrada digital D2 como entrada con capacidad de interrupción en el arduino nano. En la función *setup* se define la rutina de atención a interrupción llamada *atencionInterrupcion* que se llama ante cualquier cambio de estado de la entrada D2. La función *atencionInterrupcion* utiliza la función *micros* para leer la cantidad de microsegundos del timer interno del microcontrolador. Con estos tiempos puede determinar el ancho del pulso que recibe para controlar la posición del servomotor.

Se utiliza la salida digital D3 como señal del PWM del driver del motor. Para tener las salidas digitales INA e INB se utilizan las salidas digitales D14 y D15, las cuales utilizan los mismos pines de las entradas analógicas A0 y A1.

La función *setMotorPWM* recibe un entero en el rango de -255 a 255 y fija las entradas INA, INB y PWM del driver del motor de CD. Si la entrada es positiva gira en una dirección y si es negativa gira en dirección contraria.

La función *control_PID* implementa un controlador discreto PID (Proporcional, Integral, Derivativo) utilizando un tiempo de muestreo de 10 ms (definido con la función *delay* dentro de la función *loop*).

Adicionalmente la función *control_SuperTwisting* implementa un control Supertwisting, el cual ha sido reportado como bueno en una tesis de maestría sobre el desarrollo de un exoesqueleto, en la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana [ÁlvarezZalapa17].

```
// Propósito:  Implementa un servomotor con un motor de CD usando el driver VNH5019 y
//              un encoder absoluto de 10 bits KOYO TRD-NA1024NW
```

```
#define INA A0    // D14
#define INB A1    // D15
#define PWM 3

#define PIN_INTERRUPT 2

int timer_microsegundos = 0;

void atencionInterrupcion(){
    static long time0 = 0, time1 = 0;
    if(digitalRead(PIN_INTERRUPT))
        time0 = micros();
    else {
        time1 = micros();
        timer_microsegundos = time1 - time0;
    }
}

void setup() {
    pinMode(INA, OUTPUT);
    pinMode(INB, OUTPUT);
    pinMode(PWM, OUTPUT);
    digitalWrite(INA, LOW);
    digitalWrite(INB, LOW);
    digitalWrite(PWM, LOW);
    pinMode(PIN_INTERRUPT, INPUT);
    attachInterrupt(digitalPinToInterrupt(PIN_INTERRUPT),
                    atencionInterrupcion, CHANGE);

    Serial.begin(115200);
    Serial.print("Ready\n");
}

int pos_encoder;
int pos_ref = 500;
void loop() {
    pos_ref = lee_timer(timer_microsegundos);
```

```

    pos_encoder = leeEncoder();
    // control_Supertwisting(pos_ref, pos_encoder);
    control_PID(pos_ref, pos_encoder);
    delay(10);
}

int lee_timer(int microsegundos){
    if (microsegundos <= 0)    //error u overflow
        return pos_ref;
    if (microsegundos < 1000)
        microsegundos = 1000;
    if (microsegundos > 2000)
        microsegundos = 2000;
    return (microsegundos - 1500) / 4 + 511;
}

#define MAXPWM 80
void setMotorPWM(int pwm){
    if(pwm > MAXPWM)        pwm = MAXPWM;
    else if(pwm < -MAXPWM)  pwm = -MAXPWM;

    if(pwm == 0) {
        digitalWrite(INA, LOW); digitalWrite(INB, LOW);
    } else if(pwm > 0) {
        digitalWrite(INA, LOW);  digitalWrite(INB, HIGH);
    } else {
        pwm = - pwm;
        digitalWrite(INA, HIGH); digitalWrite(INB, LOW);
    }
    analogWrite(PWM, pwm);
}

void control_PID(int ref, int pos) {
    static float  error_ac = 0.0, error_anterior = 0, error, Kp = 10, Kd = 3, Ki = 0.1;
    static int control, abs_control;

    error = ref - pos;

```

```
    control = Kp * error + Kd * (error - error_anterior) + Ki * error_ac;
    error_anterior = error;
    error_ac += error;
    if(control >= MAXPWM || control < -MAXPWM || error == 0)
        error_ac = 0;
    setMotorPWM(control);
}

void control_Supertwisting(int ref, int pos) {
    static float lambda = 10.0, alfa = 10.0, signo, s_sqrt;
    static int control, s;

    s = ref - pos;
    if(s == 0)        signo = 0;
    else if(s > 0)    signo = 1;
    else              signo = -1;
    control = lambda * sqrt(abs(s)) * signo + alfa * signo;
    setMotorPWM(control);
}
```

4.3. Comentarios finales

En este capítulo se han descrito los elementos necesarios para integrar un servomotor de alto par utilizando un motor de CD, su driver, un encoder absoluto de 10 bits y un microcontrolador arduino.

El costo del motor es de USD \$306.67, del driver VNH5019 es de USD \$24.95, del encoder TRD-NA1024NW es USD \$32.04 y el arduino nano genérico es de \$ USD 2.00.

Capítulo 5

Estación para la generación de trayectorias de caminado

En este capítulo vamos a abordar una estrategia sencilla para la generación de movimientos de las articulaciones del exoesqueleto de tal manera que pueda caminar.

Se consideran únicamente los dos servomotores lineales y el servomotor rotacional superior de cada pierna, debido a que actualmente el exoesqueleto está siendo modificado para acoplar mecánicamente los servomotores rotacionales a los zapatos. Queda como un trabajo futuro incluir dichos servomotores en la generación de trayectorias de caminado. El código escrito se encuentra en el Apéndice A.

Se pueden tener estrategias de caminado más complejas en el exoesqueleto, pero quedan fuera del alcance de esta tesis. En el capítulo 7 se indican varias líneas interesantes que se pueden abordar.

5.1. Microcontrolador utilizado

Se eligió el microcontrolador Arduino Due para controlar los servomotores y disponer de suficiente potencia de cómputo para realizar las tareas necesarias.

El Arduino Due está basado en el procesador Atmel SAM3X8E ARM Cortex-M3 de 32-bits, que ofrece 54 pines digitales de entrada-salida (de las cuales 12 pueden ser usadas

como salidas PWM), 12 entradas analógicas, un reloj 84Mhz y maneja un voltaje de 3.3 volts [ard17a].

5.2. Señales de entrada y salida del microcontrolador

Las salidas digitales D2, D3 y D4 controlan los servomotores rotacional, lineal inferior y lineal superior respectivamente de la pierna derecha (visto el exoesqueleto desde atrás). De manera similar, las salidas digitales D5, D6 y D7 controlan los servomotores rotacional, lineal inferior y lineal superior respectivamente de la pierna izquierda.

El exoesqueleto tiene un interruptor ubicado al frente para indicar si el paso a realizar es hacia adelante o hacia atrás. El estado de este interruptor se ingresa al microcontrolador en la entrada digital D12.

Adicionalmente, junto al interruptor anterior, se dispuso un botón rojo para iniciar un paso. Cuando se oprime este botón, el exoesqueleto da un paso hacia adelante o hacia atrás, dependiendo del interruptor arriba mencionado. El estado de este botón se lee en la señal digital D11.

Adicionalmente se tienen tres potenciómetros cuyo valor es leído en las entradas analógicas A0, A1 y A2. El valor de A0 controla que tanto suben las piernas al caminar. El valor de A1 modifica el tiempo de espera para que los servomotores rotacionales efectúen el giro requerido, modificando así la velocidad de caminado. El valor de A2 controla la longitud del paso.

Finalmente hay una salida digital extra en el pin D9 que opera un relevador para energizar todos los servomotores. Esto fue necesario debido a que los servomotores rotacionales en ocasiones realizaban giros no deseados al energizarse, sin tener la señal de control apropiada.

5.3. Algoritmos de generación de trayectoria de caminado

Inicialmente los servomotores del exoesqueleto adquieren la posición vertical en ambas piernas, tocando el piso. Para realizar los movimientos de las piernas, es conveniente

resaltar que en todas las ocasiones se mantiene el mismo desplazamiento en ambos servomotores lineales de cada pierna, simplificando así la cinemática del exoesqueleto. De esta manera al girar el servomotor rotacional el efecto final es realizar un avance o retroceso del zapato.

5.3.1. Subir y bajar piernas

Se definen las funciones *sube_pierna_derecha* y *sube_pierna_izquierda* que suben el zapato de la pierna respectiva, al incrementar una cierta longitud igual en ambos servomotores lineales de la pierna. De forma similar, se definen las funciones *baja_pierna_derecha* y *baja_pierna_izquierda* que bajan el zapato de la pierna respectiva, al disminuir una cierta longitud igual en ambos servomotores lineales de la pierna.

5.3.2. Secuencia para dar un paso

Se define la función auxiliar *paso_servos_rot* que recibe como entradas el valor inicial *izq0* y final *izq1* del servomotor rotacional izquierdo, y también el valor inicial *der0* y final *der1* del servomotor rotacional derecho. El propósito de esta función es llevar el servomotor rotacional izquierdo de la posición inicial *izq0* a la posición final *izq1*, al mismo tiempo que el servomotor rotacional derecho avanza de la posición inicial *der0* a la posición final *der1*. En ambos casos se tiene un rango de valores de -50 a +50, donde -50 indica que la pierna está totalmente hacia atrás y +50 indica que la pierna está desplazada totalmente hacia adelante.

La función *dar_paso_adelante* se describe en el algoritmo 2.

En la línea L0 se ajusta el límite de 50 dependiendo del valor del potenciómetro en A2, para controlar el tamaño del paso. En la línea L1 se llama a la función que sube la pierna derecha a un valor determinado por el potenciómetro A0. En la línea L2 se llama a la función *paso_servo_rot* para realizar el avance: la pierna izquierda se mueve hacia atrás desde la posición vertical de inicio, y la pierna derecha se mueve hacia adelante. En la línea L4 y L5 se baja la pierna derecha y enseguida se sube la pierna izquierda. En la línea L6 se llama nuevamente a la función *paso_servo_rot* para llevar la pierna izquierda de atrás hacia adelante hasta la distancia límite, y la pierna derecha hacia atrás la misma distancia. En

las líneas L6, L7, L8 y L9 el proceso se repite para terminar el paso, llevando ambas piernas a la posición vertical.

```

L0: limite  $\leftarrow 50 - A2/30$  ;
L1: sube_pierna_derecha();
L2: paso_servos_rot(0, -limite, 0, limite);
L3: baja_pierna_derecha();
L4: sube_pierna_izquierda();
L5: paso_servos_rot(-limite, limite, limite, -limite);
L6: baja_pierna_izquierda();
L7: sube_pierna_derecha();
L8: paso_servos_rot(limite, 0, -limite, 0);
L9: baja_pierna_derecha();

```

Algorithm 2: Algoritmo para dar un paso hacia adelante.

La función *dar_paso_atras* se describe en el algoritmo 3, la cual es similar a la función para dar el paso hacia adelante.

```

L0: limite  $\leftarrow 50 - A2/30$  ;
L1: sube_pierna_derecha();
L2: paso_servos_rot(0, limite, 0, -limite);
L3: baja_pierna_derecha();
L4: sube_pierna_izquierda();
L5: paso_servos_rot(limite, -limite, -limite, limite);
L6: baja_pierna_izquierda();
L7: sube_pierna_derecha();
L8: paso_servos_rot(-limite, 0, limite, 0);
L9: baja_pierna_derecha();

```

Algorithm 3: Algoritmo para dar un paso hacia atrás.

5.4. Comentarios finales

En este capítulo se ha presentado una forma de que el exoesqueleto pueda dar un paso hacia adelante o hacia atrás, controlando el grado de elevación de la piernas, la velocidad del movimiento y la longitud de los pasos. La secuencia de movimientos de cada pie imita los movimientos de la persona al caminar.

La estrategia presentada mantiene en igualdad a las longitudes de los servomotores lineales de cada pierna. Si se permite variar dichas longitudes, se pueden tener otras posibles estrategias de movimiento, como se describió en el capítulo 2.

Capítulo 6

Pruebas

A continuación se describen los resultados de algunas pruebas realizadas al exoesqueleto o sus partes.

Debido a que el exoesqueleto en su versión final, con los servomotores rotacionales colocados en los zapatos, está todavía en construcción, las pruebas consideran la lectura del encoder, el control del servomotor de alto par desarrollado sin carga y la secuencia de caminado del exoesqueleto con la versión inicial que no considera servomotor en los zapatos.

6.1. Lectura del encoder absoluto de 10 bits

El encoder absoluto de 10 bits se probó utilizando el código presentado en el capítulo 4 y girando lentamente la flecha del encoder con la mano. En todos los casos se observó que se leía adecuadamente el giro realizado. La Figura 6.1 muestra el resultado de una de estas pruebas. Debido a que el programa repite la lectura cada 500ms, se puede observar que muchos valores no son consecutivos. También se puede observar que al final se pasa del número 1022 al 12, indicando el inicio de la segunda vuelta.

```

/dev/ttyACM0 (Arduino Due (Programming Port))
0000000000 (Valor Gray =0) Binario =0
0000000000 (Valor Gray =0) Binario =0
0000000000 (Valor Gray =0) Binario =0
0000000001 (Valor Gray =1) Binario =1
0000000001 (Valor Gray =1) Binario =1
0000000110 (Valor Gray =6) Binario =4
0000001110 (Valor Gray =14) Binario =11
0000010001 (Valor Gray =9) Binario =14
0000011000 (Valor Gray =24) Binario =16
0000011010 (Valor Gray =26) Binario =19
0000011100 (Valor Gray =28) Binario =23
0000010101 (Valor Gray =21) Binario =25
0000010011 (Valor Gray =19) Binario =29
0000010000 (Valor Gray =16) Binario =31
0000110011 (Valor Gray =51) Binario =34
0000110100 (Valor Gray =52) Binario =39
0000111010 (Valor Gray =58) Binario =44
0000101001 (Valor Gray =41) Binario =49
0000100101 (Valor Gray =37) Binario =57
0001100010 (Valor Gray =98) Binario =67
0001101010 (Valor Gray =106) Binario =76
0001110111 (Valor Gray =119) Binario =90
0001010110 (Valor Gray =86) Binario =100
0001001000 (Valor Gray =72) Binario =112
0001000001 (Valor Gray =65) Binario =126
0011001111 (Valor Gray =207) Binario =138
0011010111 (Valor Gray =215) Binario =154
0011111001 (Valor Gray =249) Binario =174
0010101110 (Valor Gray =174) Binario =203
0010010001 (Valor Gray =145) Binario =225
0010001000 (Valor Gray =136) Binario =240
0110001100 (Valor Gray =396) Binario =264
0110010101 (Valor Gray =405) Binario =281
0110100010 (Valor Gray =418) Binario =316
0111111010 (Valor Gray =506) Binario =339
0111000100 (Valor Gray =452) Binario =376
0101001000 (Valor Gray =328) Binario =399
0101010001 (Valor Gray =337) Binario =414
0101110101 (Valor Gray =373) Binario =422
0101101010 (Valor Gray =362) Binario =435
0100100100 (Valor Gray =292) Binario =455
0100110001 (Valor Gray =305) Binario =478
0100001010 (Valor Gray =266) Binario =499
1100011010 (Valor Gray =794) Binario =531
1100011101 (Valor Gray =797) Binario =534
1100111101 (Valor Gray =829) Binario =553
1101100111 (Valor Gray =871) Binario =581
1101110101 (Valor Gray =885) Binario =601
1101011011 (Valor Gray =859) Binario =621
1111000100 (Valor Gray =964) Binario =647
1111110111 (Valor Gray =1015) Binario =677
1110100110 (Valor Gray =934) Binario =708
1110000100 (Valor Gray =900) Binario =760
1010101000 (Valor Gray =680) Binario =816
1011101010 (Valor Gray =746) Binario =844
1011111010 (Valor Gray =762) Binario =851
1011110000 (Valor Gray =752) Binario =863
1011011001 (Valor Gray =729) Binario =878
1011001011 (Valor Gray =715) Binario =882
1011001010 (Valor Gray =714) Binario =883
1001000110 (Valor Gray =582) Binario =900
1001010000 (Valor Gray =592) Binario =927
1000100100 (Valor Gray =548) Binario =967
1000110001 (Valor Gray =561) Binario =990
1000000001 (Valor Gray =513) Binario =1022
0000001010 (Valor Gray =10) Binario =12
Autoscroll

```

Figura 6.1: Prueba del encoder absoluto.

6.2. Servomotor de alto par

El servomotor rotacional de alto par desarrollado se probó sin carga. Las Figuras 6.2 y 6.3 muestran los resultados de aplicar los controles PID y Supertwisting al servomotor, respectivamente. En ambos casos la posición inicial es de 300 y la posición de referencia es 500. Se puede observar que el desempeño es muy similar en ambos esquemas. Será muy

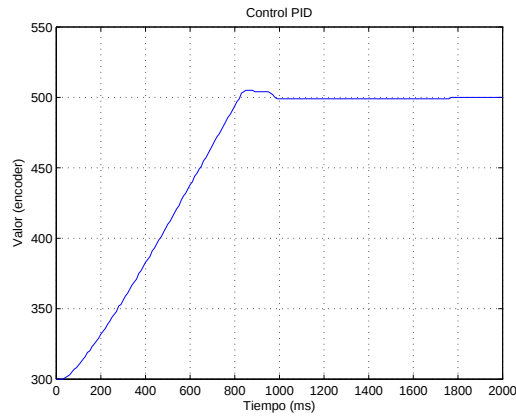


Figura 6.2: Prueba del servomotor utilizando un esquema de control PID.

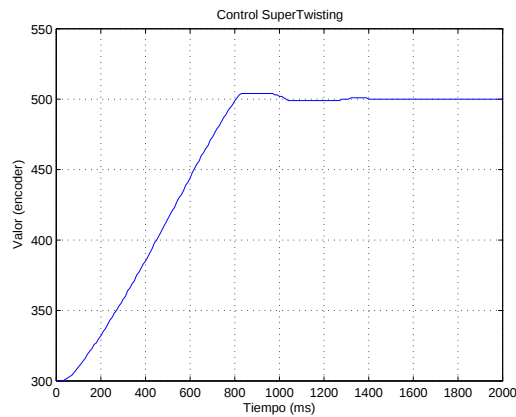


Figura 6.3: Prueba del servomotor utilizando un esquema de control Supertwisting.

interesante probar su desempeño cuando el servomotor este bajo carga en el exoesqueleto.

6.3. Secuencia de caminado

Para lograr que caminara el exoesqueleto fue necesario ponerle una suela antiderrapante a los zapatos, para tener una fuerza de fricción suficiente para lograr los movimientos. La Figura 6.4 muestra una secuencia de imágenes cuando el exoesqueleto da un paso hacia adelante.



(a)



(b)



(c)



(d)

Figura 6.4: Secuencia de imágenes del exoesqueleto dando un paso hacia adelante.

6.4. Comentarios finales

De acuerdo a las pruebas realizadas, se pudo comprobar la correcta operación del servomotor de alto par sin carga. También fue posible comprobar que el exoesqueleto puede caminar.

Capítulo 7

Conclusiones y trabajos futuros

Para realizar esta tesis fue necesario involucrarse con el análisis cinemático del mecanismo P-L extendido de 3 grados de libertad, para comprender las posibilidades de movimiento; adentrarse en el tema de los microcontroladores; y aplicar esquemas de control en el desarrollo del servomotor de alto par. Enseguida se presentan los resultados principales de este trabajo y las líneas posibles de desarrollo de trabajos futuros.

7.1. Conclusiones

Se alcanzaron las siguientes metas:

1. Se desarrolló la cinemática directa del mecanismo de P-L de 3 grados de libertad, a partir de trigonometría básica. A partir del modelo se realizaron experimentos en Matlab, comprobando las diferentes trayectorias de movimiento que se pueden lograr con el mecanismo.
2. Se construyó un servomotor rotacional de alto par utilizando un motor de CD, un encoder absoluto de 10 bits y un microcontrolador arduino nano.
3. Se generó una estrategia de utilización de los actuadores del exoesqueleto para que lograra caminar solo.

7.2. Trabajos futuros

Algunos trabajos futuros que se pueden emprender para mejorar el exoesqueleto son los siguientes:

1. Probar el servomotor rotacional de alto par en el exoesqueleto, cuando es utilizado por una persona. Hasta ahora las pruebas fueron sin una persona, debido a que el servomotor rotacional comercial no tiene el par necesario.
2. Continuar con el análisis cinemático del exoesqueleto para determinar las orientaciones de las barras que se conectan con el zapato. Esto será importante para controlar el movimiento del servomotor acoplado al zapato y a una de estas barras.
3. Incluir los dos nuevos servomotores rotacionales de los zapatos en la generación de trayectorias de caminado.
4. Determinar la trayectoria natural de caminado de las personas, posiblemente utilizando un sistema de visión, para que el exoesqueleto las reproduzca.
5. Incluir en los zapatos sensores de fuerza, de manera que el exoesqueleto pueda graduar la ayuda que aporta al caminado de la persona.
6. Incluir una computadora a bordo del exoesqueleto, podría ser una computadora de tarjeta Raspberry PI o una computadora portátil, que procese comandos hablados para la reconfiguración del exoesqueleto y el movimiento.
7. Estudiar la posibilidad de incluir un sensor de Electroencefalograma (EEG, por sus siglas en inglés .^Electroencephalography”) para controlar el exoesqueleto al detectar los pensamientos de la persona.

Estos trabajos se orientan a lograr que el exoesqueleto sea capaz de reproducir el caminar humano y adecuarse a las diferentes necesidades que requiera el proceso de rehabilitación.

Apéndice A

Programa de caminado

Enseguida se presenta el código del microcontrolador Arduino Due utilizado para generar la secuencia de caminado.

```
// Archivo:exoesqueleto_v0.5.ino

#include <Servo.h>

// SERVOS
#define SIzqLinUp    7
#define SIzqLinDown  6
#define SIzqRot      5
#define SDerLinUp    4
#define SDerLinDown  3
#define SDerRot      2

// Potenciómetros
// A0    Controla que tanto sube la pierna al caminar
// A1    Controla el tiempo de espera para los servomotores rotacionales
// A2    Controla la longitud del paso

const int LED_PIN    = 13;    // pin del LED del arduino
const int SW_FB      = 12;    // 0 = Forward, 1 = Backward
const int RedButton  = 11;    // 1 = Sin oprimir, 0 = oprimido
const int BlackButton = 10;   // 1 = Sin oprimir, 0 = oprimido
```

```

const int Relay      = 9;      // Output 1 = Cerrado, 0 = Abierto

Servo sIzqLinUp, sIzqLinDown, sIzqRot, sDerLinUp, sDerLinDown, sDerRot ;

#define E_Wait_Cmd  1

                                // AMBAS PIERNAS
#define LINEAR_MAX 120  // 130 maximo, servo estirado, pierna arriba
#define LINEAR_UP 70
#define LINEAR_WORK 63  // 61 rozando
#define LINEAR_MIN 58   // 55 pegando al piso, servo retraído

#define ROT_IZQ_MAX 140  // 150 maximo, servo izquierdo estira pierna adelante
#define ROT_IZQ_VER 90   // Pierna vertical
#define ROT_IZQ_MIN 40   // 30 minimo, servo izquierdo estira pierna atras

#define ROT_DER_MAX 120  // 130 maximo, servo derecha estira pierna atras
#define ROT_DER_VER 70   // Pierna vertical
#define ROT_DER_MIN 20   // 10 minimo, servo derecho estira pierna adelante

#define TIMER 100
#define MS_DELAY 30
#define MS_DELAY_LONG 700

int estado, demo = 0;
int timer = TIMER;

void setup() {
    estado = E_Wait_Cmd;

    pinMode(LED_PIN, OUTPUT);    // Salidas
    pinMode(Relay, OUTPUT);
    pinMode(SW_FB, INPUT);       // Entradas
    pinMode(RedButton, INPUT);
    pinMode(BlackButton, INPUT);
    digitalWrite(LED_PIN, LOW);
    digitalWrite(Relay, LOW);

```

```
sIzqLinUp.attach(SIzqLinUp); // Servos
sIzqLinDown.attach(SIzqLinDown);
sIzqRot.attach(SIzqRot);
sDerLinUp.attach(SDerLinUp);
sDerLinDown.attach(SDerLinDown);
sDerRot.attach(SDerRot);

servo(SIzqLinUp, LINEAR_WORK-2);
servo(SIzqLinDown, LINEAR_WORK-2);
servo(SIzqRot, ROT_IZQ_VER);
servo(SDerLinUp, LINEAR_WORK);
servo(SDerLinDown, LINEAR_WORK);
servo(SDerRot, ROT_DER_VER);

Serial.begin(9600);
Serial.print("Version 0.5");
Serial.println(" Ready");

digitalWrite(Relay, HIGH);
digitalWrite(LED_PIN, HIGH);
}

#define LIM_SUP 50
#define LIM_INF -50

void servo_rot_izq(int valor) { // valor entre [-50, 50] +50=adelante, -50= atras
    if(valor > LIM_SUP) valor = LIM_SUP;
    if(valor < LIM_INF) valor = LIM_INF;
    servo(SIzqRot, ROT_IZQ_VER + valor);
}

void servo_rot_der(int valor) { // valor entre [-50, 50] +50=adelante, -50= atras
    if(valor > LIM_SUP) valor = LIM_SUP;
    if(valor < LIM_INF) valor = LIM_INF;
    servo(SDerRot, ROT_DER_VER - valor);
}
```

```
void sube_pierna_derecha() {
    float val = analogRead(A0) / 1024.0f * 20;
    int pos = val + LINEAR_UP;

    servo(SDerLinUp, pos);
    servo(SDerLinDown, pos);
    delay(MS_DELAY_LONG);
}

void sube_pierna_izquierda() {
    float val = analogRead(A0) / 1024.0f * 20;
    int pos = val + LINEAR_UP;

    servo(SIzqLinUp, pos);
    servo(SIzqLinDown, pos);
    delay(MS_DELAY_LONG);
}

void baja_pierna_derecha() {
    servo(SDerLinUp, LINEAR_WORK);
    servo(SDerLinDown, LINEAR_WORK);
    delay(MS_DELAY_LONG);
}

void baja_pierna_izquierda() {
    servo(SIzqLinUp, LINEAR_WORK);
    servo(SIzqLinDown, LINEAR_WORK);
    delay(MS_DELAY_LONG);
}

void paso_servos(int izq0, int izq1, int der0, int der1) {
    int inc_izq, inc_der;

    inc_izq = izq1 > izq0 ? 1 : -1;
    inc_der = der1 > der0 ? 1 : -1;
    do {
        izq0 += inc_izq;
```

```

        der0 += inc_der;
        servo_rot_izq(izq0);
        servo_rot_der(der0);
        delay(timer);
    } while(izq0 != izq1);
}

int limite = 50;

void dar_paso_adelante() {
    limite = 50 - analogRead(A2)/30;

    sube_pierna_derecha();
    paso_servos(0, -limite, 0, limite);
    baja_pierna_derecha();
    sube_pierna_izquierda();
    paso_servos(-limite, limite, limite, -limite);
    baja_pierna_izquierda();
    sube_pierna_derecha();
    paso_servos(limite, 0, -limite, 0);
    baja_pierna_derecha();
}

void dar_paso_atras() {
    limite = 50 - analogRead(A2)/30; servo(n, val

    sube_pierna_derecha();
    paso_servos(0, limite, 0, -limite);
    baja_pierna_derecha();
    sube_pierna_izquierda();
    paso_servos(limite, -limite, -limite, limite);
    baja_pierna_izquierda();
    sube_pierna_derecha();
    paso_servos(-limite, 0, limite, 0);
    baja_pierna_derecha();
}
servo(n, val

```

```
int c0,c1,c2, n, val, pos;

void loop() {
  switch(estado) {
    case E_Wait_Cmd:
      while(Serial.available() == 0) { // espera recibir una entrada del serial
        timer = analogRead(A1) / 10 + MS_DELAY;
        if(digitalRead(RedButton) == 0) {
          if(digitalRead(SW_FB) == 0)
            dar_paso_adelante();
          else
            dar_paso_atras();
        }
      }

      c0 = Serial.read();
      switch (c0) {
        case 's': // Comando 's' para mover un servomotor
          n = Serial.parseFloat();
          val = Serial.parseFloat();
          Serial.read();
          servo(n,val);
          Serial.println("t");
          break;
      }
      break;
  }
}

void servo(int n, int val) {
  switch(n){
    case SIzqLinUp:
    case SIzqLinDown:
    case SDerLinUp:
    case SDerLinDown: if (val > LINEAR_MAX) val = LINEAR_MAX;
                      if (val < LINEAR_MIN) val = LINEAR_MIN;
                      break;
  }
}
```

```
        case SIzqRot:      if (val > ROT_IZQ_MAX) val = ROT_IZQ_MAX;
                           if (val < ROT_IZQ_MIN) val = ROT_IZQ_MIN;
                           break;

        case SDerRot:      if (val > ROT_DER_MAX) val = ROT_DER_MAX;
                           if (val < ROT_DER_MIN) val = ROT_DER_MIN;
                           break;

    }

    switch(n){

        case SIzqLinUp:    sIzqLinUp.write(val); break;
        case SIzqLinDown:  sIzqLinDown.write(val); break;
        case SIzqRot:      sIzqRot.write(val); break;
        case SDerLinUp:    sDerLinUp.write(val); break;
        case SDerLinDown:  sDerLinDown.write(val); break;
        case SDerRot:      sDerRot.write(val); break;

    }

}
```


Referencias

- [Accoto14] Accoto, B. D., Sergi, F., Tagliamonte, N. L., y Bion, R. E. A non anthropomorphic wearable robot. December 2014.
- [Anam12] Anam, K. y Al-Jumaily, A. Active exoskeleton control systems: State of the art. *Procedia Engineering*, 41, 2012.
URL <http://doi.org/10.1016/j.proeng.2012.07.273>
- [ard17a] *Arduino Due, características*, 2017.
URL <https://store.arduino.cc/usa/arduino-due>
- [Ard17b] *Servomotores y Arduino*, 2017.
URL <https://www.arduino.cc/en/Reference/Servo>
- [Asbeck14] Asbeck, A. T., DeRossi, S. M., Galiana, I., Ding, Y., y C, J. W. Stronger, smarter, softer. *IEEE Robotics and Automation Magaz*<https://store.arduino.cc/arduino-dueine>, 21, 2014.
- [Barrientos07] Barrientos, A., Peñón, L. F., Balaguer, C., y Aracil, R. *Fundamentos de robótica*. McGraw-Hill, 2^a ed^{ón}., 2007.
- [Cao14] Cao, J., Xie, S. Q., Das, R., y Zhu, G. L. Control strategies for effective robot assisted gait rehabilitation: The state of art and future prospects. *Medical Engineering and Physics*, 2014.
URL <http://doi.org/10.1016/j.medengphy.2014.08.005>
- [Casolo08] Casolo, F., Cinquemani, S., y Cocetta, M. On active lower limb exoskeletons actuators. *En Proceeding of the 5th International Sym-*

- posium on Mechatronics and Its Applications, ISMA*. 2008.
URL <http://doi.org/10.1109/ISMA.2008.4648796>
- [Chen16] Chen, B., QIN, H. M. L., Gao, F., Chan, K., Law, S., y Liao, W. Recent developments and challenges of lower extremity exoskeletons. *Journal of Orthopedic Translation*, 2016.
URL <http://doi.org/10.1016/j.jot.2015.09.007>
- [Chiang13] Chiang, M. H. y Chang, F. R. Anthropomorphic design of the human-like walking robot. *Journal of Bionic Engineering*, 2013.
URL [http://doi.org/10.1016/S1672-6529\(13\)60214-0](http://doi.org/10.1016/S1672-6529(13)60214-0)
- [cod17] *Código Gray, Wikipedia, la enciclopedia libre*, 2017.
URL https://es.wikipedia.org/wiki/C%C3%B3digo_Gray
- [Dollar08] Dollar, A. M. y Herr, H. Lower extremity exoskeletons and active orthoses: Challenges and state of the art. *Transactions on Robotics*, 2008.
- [Dudek10] Dudek, G. y Jenkin, M. *Computational Principles of Mobile Robotics*. Cambridge University Press, New York, NY, USA, 2^a ed^{ón}., 2010. ISBN 0521692121, 9780521692120.
- [Fontana14] Fontana, M., Vertechy, R., Marchetti, S., Salsedo, S., y Bergamasco, M. The body extender. *IEEE Robotics and Automation Magazine*, December 2014.
- [Galle13] Galle, S., Malcolm, P., Derave, W., y caja de Cambios de motor eléctrico de Alta potencia Industrial de Gran par a Baja velocidad De Clercq, D. D. . V. . V. . W. G. Adaptation to walking with an exoskeleton that assist ankle extension. *Gait and Posture*, 2013.
URL <http://doi.org/10.1016/j.gaitpost.2013.01.029>

- [Gea17] Gearwurx, *servomotor rotacional Torxis Servo i00600*, 2017.
URL http://gearwurx.com/wp-content/uploads/2017/08/i03943_05.DTA_SHT_TRXS.pdf
- [Godoy17] Godoy, J. C. Estudio biomecánico aplicado a exoesqueletos con arquitectura no antropomórfica, 2017. Investigación doctoral.
- [HDLS-4-30-12v17] HDLS-4-30-12v. *Servocity, motor lineal HDLS-4-30-12v*, 2017.
URL <https://www.servocity.com/hdls-4-30-12v>
- [INEGI15] INEGI. Estadísticas a propósito del día internacional de las personas con discapacidad. 2015.
URL <http://http://www.inegi.org.mx/saladeprensa/aproposito/2015/discapacidad0.pdf>
- [Lo12] Lo, H. S. y Xie, S. Q. Exoskeleton robots for upper-limb rehabilitation: state of the art and future prospects. *Medical Engineering and Physics*, 34, 2012.
URL <http://doi.org/10.1016/j.medengphy.2011.10.004>
- [Lopez14] Lopez, R., Aguilar, H., Lozano, R., y Torres, J. Modelado y control de un exoesqueleto para la rehabilitación de la extremidad inferior con dos grados de libertad. *Revista Iberoamericana de Automática e informática Industrial, RIAI*, 2014.
- [Low06] Low, K. H., Liu, X., y Yu, H. Design and implementation of ntu wearable exoskeleton as an enhancement and assistive. *Hindawi Applied Bionics and Biomechanics*, 2006.
URL <http://doi.org/10.1533/abbi.2006.003>
- [Luquin15] Luquin, F. Maquinas de vapor, mecanismos y p.l chebyshev. *En Un paseo por la geometría-Matematicalia*. 2015.
- [MATLAB10] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

- [motorGW11416517] motor GW114165. *Sumotor, motor de CD*, 2017.
URL <https://es.aliexpress.com/>
- [Núñez-Altamirano16] Núñez-Altamirano, D. A. *Desarrollo de una unidad de propulsión basada en un mecanismo de línea recta exacta aplicada en vehículos multípodos*. Tesis Doctoral, Universidad Michoacana de San Nicolás de Hidalgo, Facultad de Ingeniería Mecánica, Programa del doctorado en ciencias en Ingeniería Mecánica, Agosto 2016.
- [Petric14] Petric, F. y Hrvatini, K. Four tasks of a robot-assisted autism spectrum disorder diagnostic protocol: First clinical test. *En IEEE Global Humanitarian Technology Conference*. IEEE, 2014.
- [Pons08] Pons, J. L. *Wearable Robots: Biomechatronic Exoskeletons*. John Wiley & Sons, 2008.
- [Sabanovic13] Sabanovic, S., Bennet, C. C., Chang, W. L., y Huber, L. Paro robot affects diverse interaction modalities in group sensory therapy for older adults with dementia. *En IEEE International Conference on Rehabilitation Robotics*. IEEE, 2013.
- [ser17a] *Componentes de un servomotor*, 2017.
URL http://1.bp.blogspot.com/-3VJnUUb_y6o/UR_BJmEK1bI/AAAAAAAAAD18/PSSluj0k05w/s1600/servo2.gif
- [ser17b] *Componentes de un servomotor lineal*, 2017.
URL <https://www.luisllamas.es/arduino-actuador-lineal/>
- [Siegwart11] Siegwart, R., Nourbakhsh, I. R., y Scaramuzza, D. *Introduction to Autonomous Mobile Robots*. The MIT Press, 2^a ed^{ón}, 2011. ISBN 0262015358, 9780262015356.
- [Stefanov04] Stefanov, D. y Bien, Z. Z. Human friendly technologies movement assistance and restoration for people with disabilities. *En Advances in Rehabilitation Robotics*. Springer Berlin Heidelberg, 2004.

-
- [TRD-NA1024NW17] TRD-NA1024NW. *Koyo, encoders rotatorios*, 2017.
URL http://www.koyoele.co.jp/english/product/encoder/pdf/TRD_NA.pdf
- [VNH501917] VNH5019. *Pololu, Motor Driver Carrier*, 2017.
URL <https://www.pololu.com/product/1451>
- [Yu14] Yu, S., Han, C., y Cho, I. Design considerations of a lower limb exoskeletons system to assist alking and load-carrying of infantry soldiers. *11*, 2014.
- [ÁlvarezZalapa17] Álvarez Zalapa, S. *Modelado y control de un exoesqueleto de miembros inferiores*. Proyecto Fin de Carrera, Universidad Michoacana de San Nicolás de Hidalgo, Facultad de Ingeniería Eléctrica, Programa de Maestría en Ciencias en Ingeniería Eléctrica, Junio 2017.