



**Universidad Michoacana de  
San Nicolás de Hidalgo**



**Facultad de Ingeniería Mecánica  
División de Estudios de Posgrado**

**Diseño del sistema de control electrónico del robot  
LUNAbot W6 VJ1.0 utilizando lógica difusa**

Tesis

que para obtener el grado de:

**Maestra en Ciencias en Ingeniería Mecánica**

Presenta:

**Artemisa Zaragoza Ibarra**

Asesor de Tesis:

**Doctor en Ingeniería Mecánica Ignacio Juárez Campos**

Coasesor de Tesis:

**Maestro en Ingeniería Mecánica Miguel Ángel Vega González**



Morelia, Michoacán, Febrero del 2015

# Índice

Lista de Figuras .....	4
Lista de Tablas .....	5
Lista de Diagramas.....	6
RESUMEN.....	7
Palabras Clave:.....	7
ABSTRACT .....	8
Keywords: .....	8
Referencia de siglas y abreviaturas .....	1
Glosario de términos y conceptos .....	2
INTRODUCCIÓN.....	4
Capítulo 1. FUNDAMENTOS DE LA INVESTIGACIÓN .....	5
1.1. Planteamiento del problema .....	5
1.1.1. Descripción del problema .....	5
1.1.1.1. Problema de investigación.....	8
1.1.2. Preguntas de investigación.....	8
1.1.2.1. Pregunta general.....	8
1.1.2.2. Preguntas específicas .....	8
1.1.3. Objetivos de la investigación .....	9
1.1.3.1. Objetivo general .....	9
1.1.3.2. Objetivos específicos.....	9
1.1.4. Hipótesis de la investigación.....	9
1.1.4.1. Hipótesis general.....	9
1.1.4.2. Hipótesis específicas .....	10
1.1.5. Justificación .....	10
1.1.5.1. Trascendencia.....	10
1.1.5.2. Importancia.....	11
1.1.6. Horizonte temporal y espacial.....	11
1.1.6.1. Viabilidad de la investigación.....	12
1.1.7. Alcance de la investigación.....	12
1.1.8. Alcances y limitaciones de la investigación.....	12
1.1.9. Alcances .....	12
1.1.10. Limitaciones .....	13
1.2. Identificación de variables.....	13

1.2.1.	Variables dependientes .....	13
1.2.2.	Variables independientes .....	13
Capítulo 2.	MARCO TEÓRICO .....	14
2.1.	El control y su evolución .....	14
2.2.	Lógica difusa .....	23
2.3.	Microcontroladores .....	25
2.4.	Máquinas caminantes .....	27
2.5.	Mecanismo Peaucellier-Lipkin .....	32
Capítulo 3.	DISEÑO DEL CONTROL.....	34
3.1.	Descripción final del robot.....	34
3.2.	Definición de la forma de movimiento del robot .....	35
3.3.	Diseño de las metodologías de movimiento.....	39
3.4.	Diseño de las metodologías finales de movimiento.....	43
3.5.	Elección del tipo de actuadores del robot .....	52
3.5.1.	Actuador de la pata.....	52
3.5.2.	Actuador del fémur y de configuración de giro .....	54
3.5.3.	Elección del tipo de microcontrolador a utilizar .....	55
3.6.	Diseño del control de lógica difusa .....	57
3.6.1.	Análisis del perfil de velocidad del mecanismo Peaucellier-Lipkin .....	57
3.6.2.	Diseño de las funciones de membresía .....	59
3.6.3.	Diseño de las reglas del control difuso.....	63
3.7.	Codificación del control difuso .....	77
3.8.	Diseño del circuito electrónico .....	90
Capítulo 4.	CONCLUSIONES.....	106
Bibliografía.....		108

## Lista de Figuras

Figura 2.1 Robot móvil.....	29
Figura 2.2 Robot androide .....	29
Figura 2.3 Robot poliarticulado .....	29
Figura 2.4 Robot zoomorfo .....	29
Figura 2.5 Robot Híbrido.....	29
Figura 2.6 Robot de servicio .....	30
Figura 2.7 Robot de seguridad y espacio.....	30
Figura 2.8 Robot de entretenimiento.....	30
Figura 2.9 Robot industrial.....	30
Figura 2.10 Mecanismo Peaucellier-Lipkin.....	33
Figura 2.11 Peaucellier-Lipkin con líneas auxiliares.....	34
Figura 3.1 Mecanismo Peaucellier-Lipkin con las barras DE y EF añadidas. ....	35
Figura 3.2 a) Distribución de extremidades en un arácnido. b) Distribución de extremidades en una morfología similar al cangrejo. ....	35
Figura 3.3 a) Movimiento rectilíneo hacia adelante y hacia atrás. b) Movimiento circular con centro de giro dentro del espacio físico del robot. c) Movimiento circular con centro de giro fuera del espacio físico del robot.....	37
Figura 3.4 Líneas imaginarias mostrando el centro de giro del robot con las patas alineadas sobre las líneas.....	37
Figura 3.5 Nomenclatura de las patas .....	38
Figura 3.6 Motores de posicionamiento. Estos motores se encuentran ubicados en el centro del cuerpo del robot. ....	38
Figura 3.7 Cantidad y distribución de los motores del robot. ....	39
Figura 3.8 Primera nomenclatura utilizada para el diseño de metodologías.....	40
Figura 3.9 Nomenclatura de los motores .....	44
Figura 3.10 Forma final del Lunabot W6 VJ1.0 .....	52
Figura 3.11 Motor FF-050SB .....	53
Figura 3.12 Pierna del Lunabot. Se puede observar el tornillo de la pata y la posición del motor .....	53
Figura 3.13 Forma y dimensiones del FF-050SB [52] .....	54
Figura 3.14 Forma y dimensiones del motor DS3115 Metal Digital Servo.....	55
Figura 3.15 Diagrama del microcontrolador ATmega328P [21].....	56
Figura 3.16 Mecanismo Peaucellier-Lipkin en el que se muestran los puntos involucrados en el control .....	57
Figura 3.17 Velocidad del punto B aplicando al punto A una velocidad constante de 9 rad/s .....	58
Figura 3.18 Gráfica de los rangos tomados para el diseño de las funciones de membresía de entrada. ....	61
Figura 3.19 Funciones de membresía de entrada para el control proporcional .....	62
Figura 3.20 Perfil de velocidad deseado para los servomotores de las piernas. ....	62

Figura 3.21 Primer propuesta de funciones de membresía de salida para el control proporcional.....	63
Figura 3.22 Reglas del control P para el LUNAbot W6 VJ1.0.....	64
Figura 3.23 Perfil de velocidad obtenido después de la primera prueba .....	65
Figura 3.24 Funciones de membresía de salida modificadas.....	66
Figura 3.25 Perfil de velocidad de salida obtenido con las nuevas funciones de membresía de salida .....	66
Figura 3.26 Programa de bloques para la simulación del control con lógica difusa del LUNAbot W6 VJ1.0.....	68
Figura 3.27 Perfil de velocidad obtenido después de la simulación utilizando el control Proporcional medida en rad/s. ....	70
Figura 3.28 Reglas del control difuso PD en el módulo Fuzzy de Matlab .....	72
Figura 3.29 Superficie de la salida del control PD.....	73
Figura 3.30 Programación en bloques del control P+PD utilizando Simulink.....	74
Figura 3.31 Salida del control PD en rad/s.....	75
Figura 3.32 Acercamiento a las perturbaciones de la fase estable del control. ....	75
Figura 3.33 Salida del control P+PD en rad/s. ....	76
Figura 3.34 Ejemplo del proceso de fuzzificación y defuzzificación.....	78
Figura 3.35 Esquema del circuito para el control del LUNAbot W6 VJ1.0 .....	93
Figura 3.36 Esquema del circuito para el control del LUNAbot W6 VJ1.0 (parte de los motores). ....	94
Figura 3.37 Esquema del circuito para el control del LUNAbot W6 VJ1.0 (parte del microcontrolador).....	95
Figura 3.38 Diseño del circuito. En azul se muestran las conexiones de la parte inferior de la baqueta. En rojo se muestran las conexiones de la parte superior de la baqueta.....	104
Figura 3.39 Imagen lista para imprimir el circuito del LUNAbot W6 VJ1.0.....	105

## Listado de Tablas

Tabla 3.1 Tabla del movimiento rectilíneo.....	41
Tabla 3.2 Esta tabla representa las instrucciones que deberán darse a cada salida del microcontrolador .....	41
Tabla 3.3 Funcionamiento del sensor de posición relativa entre DE y EF.....	42
Tabla 3.4 Secuencia del microcontrolador para el movimiento rectilíneo. ....	42
Tabla 3.5 Secuencia del microcontrolador para el movimiento circular relativa entre DE y EF.....	43
Tabla 3.6 Definición de los grupos de motores. ....	44
Tabla 3.7 Metodología de movimiento rectilíneo hacia adelante utilizando los grupos de motores.....	45
Tabla 3.8 Metodología de movimiento rectilíneo hacia atrás utilizando los grupos de motores.....	45
Tabla 3.9 Metodología de movimiento para la orientación de las piernas del robot para realizar un giro en cualquier sentido .....	46

Tabla 3.10 Metodología para detener el movimiento del robot.....	46
Tabla 3.11 Especificaciones del motor FF-050SB [52].....	54
Tabla 3.12 Especificaciones del motor DS3115 Metal Digital Servo [53].....	55
Tabla 3.13 Especificaciones generales del Atmel ATmega328P [21].....	56
Tabla 3.14 Tabla de las reglas para el control.....	64
Tabla 3.15 Matriz de reglas para el control difuso PD.....	72
Tabla 3.16 Tabla de elementos para el circuito electrónico.....	90

## **Lista de Diagramas**

Diagrama 3.1 Diagrama para la secuencia de movimiento hacia adelante del robot.....	48
Diagrama 3.2 Diagrama para la secuencia de movimiento hacia atrás del robot. ....	49
Diagrama 3.3 Diagrama para la secuencia de paro del robot.....	50
Diagrama 3.4 Diagrama para el movimiento en cualquier dirección del robot. ....	51

# RESUMEN

La presente investigación busca ser un aporte para resolver el problema de la falta de accesibilidad universal, para las personas con discapacidades motrices en México, mediante el diseño de un sistema de control electrónico para un robot hexápodo a escala, que sea la base de la construcción de un robot caminante de asistencia médica, capaz de desplazar a una persona que tuviera alguna discapacidad motriz.

Actualmente la población que sufre de discapacidades motrices tiene muchos obstáculos para integrarse a la sociedad de forma activa. Uno de los principales es la falta de infraestructura en las zonas urbanas y rurales para su desplazamiento, por eso es importante desarrollar mecanismos de locomoción fáciles de utilizar y que sean capaces de satisfacer las necesidades del usuario sin importar el lugar en donde realice sus actividades.

## **Palabras Clave:**

1. Hexápodo
2. Control
3. Lógica Difusa
4. Discapacidad motriz
5. Lunabot W6 VJ 1.0

# ABSTRACT

This study intends to be a contribution to solving the problem of the lack of universal accessibility for people with motor disabilities in Mexico, through the design of an electronic control system for a hexapod robot to scale, that is the basis of building of a walking robot healthcare, capable of moving a person who had any physical disability.

Currently the population suffering from motor disabilities have many obstacles to integrate into society actively. One of the main ones is the lack of infrastructure in urban and rural areas for movement, so it is important to develop mechanisms for locomotion easy to use and san able to meet the user's needs regardless of the place where he carries on business.

## **Keywords:**

1. Hexapod
2. Control
3. Fuzzy Logic
4. Motor impairment
5. Lunabot W6 VJ 1.0

## **Referencia de siglas y abreviaturas**

**CNDH.** Comisión Nacional de los Derechos Humanos

**DDC:** Direct digital control [1].

**DIF.** Sistema para el Desarrollo Integral de la Familia.

**IAT.** Institut Avtomatiki i Telemekhaniki. Instituto de Automatización y Equipos de Control Remoto

**KTA.** Kommissiya Telemekhaniki i Avtomatiki Comisión de Automatización y Robots

**LQG:** Linear-quadratic-Gaussian [2].

**MIT.** Massachusetts Institute of Technology

**OMS.** Organización Mundial de la Salud

**PD.** Proporcional Derivativo.

**PI.** Proporcional Integral

**PID.** Proporcional Integral Derivativo.

**UMSNH.** Universidad Michoacana de San Nicolás de Hidalgo

**Vcd.** Voltaje de Corriente Directa.

## Glosario de términos y conceptos

**Accesibilidad Universal:** Las medidas pertinentes para asegurar el acceso de las personas con discapacidad, en igualdad de condiciones con las demás, al entorno físico, el transporte, la información y las comunicaciones [3].

**Accesibilidad:** Combinación de elementos constructivos y operativos que permiten a cualquier persona con discapacidad, entrar, desplazarse, salir, orientarse y comunicarse con el uso seguro, autónomo y cómodo en los espacios construidos, el mobiliario y equipo, el transporte, la información y las comunicaciones. [4]

**Asistencia:** Acción de prestar socorro, favor o ayuda. [5]

**Ayudas Técnicas:** Dispositivos tecnológicos y materiales que permiten habilitar, rehabilitar o compensar una o más limitaciones funcionales, motrices, sensoriales o intelectuales de las personas con discapacidad. [6]

**Bipedestador:** dispositivo formado por una estructura metálica de tubo que mantiene erecto el cuerpo del paciente, usualmente aplicando sobre él varios tipos de apoyo, uno sobre el calcáneo, otro anterior en el tercio superior de la tibia, un tercero posterior-superior sobre la zona de la pelvis, y una sujeción anterior sobre el tórax. [7]

**Control Bang-Bang:** Control cuyo funcionamiento sólo responde a los estados de encendido y apagado.

**Discapacidad:** Deficiencia física, mental, intelectual o sensorial que al interactuar con distintos ambientes del entorno social pueden impedir su participación plena y efectiva en igualdad de condiciones a las demás. [8]

**Diseño:** Es el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso, o sistema, con los suficientes detalles como para permitir su realización física. [9]

**Ortesis:** Dispositivos aplicados externamente utilizados para modificar las características estructurales y funcionales del sistema neuromuscular y esquelético. [10]

# INTRODUCCIÓN

En la actualidad, existen algunos medios de locomoción para personas con discapacidades motrices como son las sillas de ruedas, andaderas, bastones, y en algunos países la ortesis. Sin embargo en México la mayoría de las personas que tienen una discapacidad motriz no puede utilizar ninguno de estos medios ya sea por su precio, por la falta de infraestructura en los espacios públicos para poder utilizarlos, o porque residen en zonas rurales donde el terreno es irregular y no tienen las condiciones apropiadas para desplazarse. Son muchas las personas que en el afán de conseguir un poco de comodidad, modifican sus aparatos ya sea para ajustar su tamaño o para realizar reparaciones de daños causados debido al uso. La falta de opciones en los medios de desplazamiento para las personas con discapacidades motrices ha ocasionado que éstas tengan demasiadas dificultades para integrarse a la sociedad de forma activa y las obliga a depender siempre de otra persona.

Actualmente, se ha propuesto un nuevo robot de desplazamiento que permita llevar a una persona con discapacidad motriz sobre cualquier terreno con el mínimo esfuerzo. Este robot es el LUNAbot W6, un hexápodo diseñado con el mecanismo Peaucellier-Lipkin. El uso de este mecanismo permite reducir las oscilaciones que experimenta la persona que se encuentra utilizando el robot al caminar, ya que nos permite obtener un movimiento en línea recta partiendo de uno circular. Para que todo esto sea posible es necesario tener un control electrónico que el usuario pueda manejar de forma sencilla. En este trabajo se trata el diseño de dicho control electrónico que implementado al LUNAbot W6 VJ1.0, una versión a escala del robot de asistencia médica, logrará su movimiento en forma suave y dejará un precedente para el diseño y construcción del robot.

# Capítulo 1. FUNDAMENTOS DE LA INVESTIGACIÓN

En esta investigación se realizaron cuatro análisis y cálculos que permitieron diseñar el sistema de control del hexápodo caminante LUNAbot W6 VJ1.0 utilizando la lógica difusa como método de diseño del control. En esta primera parte se abordaron los fundamentos de este trabajo de tesis.

## 1.1. Planteamiento del problema

El problema surgió debido a que no existe un sistema electrónico capaz de mover el robot LUNAbot W6 VJ1.0, el cual se planteó como prototipo para el diseño, optimización y construcción de un robot de asistencia médica en apoyo a la solución del problema de accesibilidad universal para las personas con discapacidades motrices.

### 1.1.1. Descripción del problema

En México, las personas con discapacidades motrices no pueden desplazarse fácilmente en terrenos irregulares en la ciudad o en comunidades rurales. Las consecuencias que este problema puede causar son bastas y entre ellas se encuentran las consecuencias físicas como ulceraciones en el cuerpo por movilidad escasa o nula, atrofia muscular, depresión, deformaciones óseas por los aparatos ortopédicos utilizados para desplazarse, dolor muscular, etc. Sin embargo también existen consecuencias sociales dentro de las cuales las principales son la discriminación y la inestabilidad económica [11].

De acuerdo al diagnóstico nacional de accesibilidad realizado por la CNDH en México [12], en el país, el diagnóstico de accesibilidad física en inmuebles de la administración pública federal y área urbana cercana es *limitada*, o en los mejores casos *reducida o fragmentaria*. Basados en información publicada por la OMS

[13], de los 65 millones de personas en todo el mundo que requieren una silla de ruedas, por lo menos 20 millones de ellas no tienen acceso a una y quienes cuentan con silla de ruedas propia son minoría. “Rara vez se cuenta con financiamiento público para el suministro de una silla de ruedas y los usuarios, en su mayoría quedan en la incapacidad de pagar por una por cuenta propia”.

La OMS afirma que “como muchos usuarios son pobres, viven en casas pequeñas o en chozas situadas en lugares inaccesibles. También viven donde los sistemas viales son malos, los pavimentos escasean y el clima y el terreno son a menudo extremos. En muchos casos, el acceso a los edificios públicos y privados es difícil para una silla de ruedas. Dichas barreras físicas agregan exigencias a la resistencia y durabilidad de las sillas de ruedas. También exigen que los usuarios, si han de movilizarse, desplieguen un alto grado de destreza”.

En numerosos países en desarrollo sólo el 3% de las personas con discapacidades que requieren rehabilitación tienen accesos a esos servicios [14]. A lo largo del tiempo varias alternativas han surgido al uso de sillas de ruedas. Dos de ellas, los bipedestadores y la ortesis, son los más comunes alrededor del mundo, sin embargo de acuerdo a la investigación, hasta mayo del 2014 en México, los programas de gobierno del DIF no los consideran para su distribución, por lo que exclusivamente se distribuyen sillas de ruedas bastones, andaderas y muletas para mejorar la calidad de vida de personas con discapacidades, motrices. Además, a los usuarios no se les da la oportunidad de elegir la silla de ruedas más apropiada sino que quedan sujetos al modelo que se encuentra disponible en su lugar de residencia o al que los programas de gobierno les otorgan en algunos casos. Tampoco existe una variedad en cuanto a tamaños, lo que en algunos casos ocasiona incluso que la silla de ruedas sea modificada para adecuarse a las necesidades físicas del usuario, su estilo de vida o su entorno hogareño y laboral.

Existen causas que han impedido hasta la fecha el desarrollo de medios de transporte diferentes que faciliten la movilidad de las personas con discapacidades motrices. Janowsky [15] expresa en su trabajo “Políticas de la Salud y Sistemas de

Desarrollo” realizado para la OMS, algunos factores determinantes para inversión en investigación de nueva tecnología son la relación costo eficacia de la inversión y preferencias de la comunidad, por lo que el gobierno considera no orientar recursos a este tipo de investigación. La participación de las universidades para el desarrollo de transportes novedosos se enfoca más en proporcionar los espacios físicos y las herramientas básicas necesarias para el diseño y construcción de dichos transportes, pero no a su financiamiento. Sin embargo en el programa de acción específico para gestión de equipo médico diseñado para el ejercicio 2007-2012 [16], se invita a las empresas públicas, privadas y universidades a trabajar conjuntamente en investigación y desarrollo, por lo que puede pensarse así en convenios que permitan el desarrollo de proyectos para el diseño y construcción de este tipo de transportes dentro de las universidades públicas con financiamiento de empresas privadas.

Con las condiciones anteriormente expuestas se propone el desarrollo de un robot hexápodo de asistencia médica utilizando el mecanismo Peaucellier-Lipkin como sistema de locomoción desarrollado en la UMSNH y con financiamiento de la empresa Lunabotics.org. El objetivo de este hexápodo es permitir el desplazamiento de las personas con discapacidades motrices de una manera cómoda, reduciendo las oscilaciones en el traslado y proporcionando un sistema de control electrónico con el que la destreza física requerida para mover el robot sea mínima.

Con la finalidad de llevar a cabo la construcción de este robot es necesario contar con un diseño mecánico del mismo así como el diseño del sistema de control, pero principalmente con un medio de financiamiento para llevar a cabo dicho fin. Por lo tanto se propone desarrollar un robot hexápodo prototipo similar al de asistencia médica, a escala y con fines de entretenimiento para el cual se requiera una inversión mucho menor y que pueda construirse en menos tiempo. Se entiende que a este robot prototipo se le diseñará un control electrónico que servirá de base para el control del robot de asistencia médica.

El desarrollo de este prototipo implica el diseño mecánico, un sistema de control, estudios de mercado entre otras cosas. El alcance de esta investigación está acotado al diseño del sistema de control del prototipo de entretenimiento con el fin de realizar un aporte a la solución del problema principal.

#### **1.1.1.1. Problema de investigación**

El problema surgió al hecho de que no existe un sistema electrónico capaz de mover el robot LUNAbot W6 VJ1.0, para ser utilizado como prototipo para el diseño, optimización y construcción de un robot de asistencia médica que apoye a la solución del problema de accesibilidad universal para las personas con discapacidades motrices.

#### **1.1.2. Preguntas de investigación**

##### **1.1.2.1. Pregunta general**

¿Es posible diseñar un sistema de control electrónico para el LUNAbot W6 VJ1.0 utilizando lógica difusa?

##### **1.1.2.2. Preguntas específicas**

1. ¿Qué movimientos se requiere que realice el LUNAbot W6 VJ1.0? (Página 35)
2. ¿Qué mecanismo impulsa las piernas del LUNAbot W6 VJ1.0? (Página 34)
3. ¿Qué mecanismo impulsa las patas del LUNAbot W6 VJ1.0? (Página 34)
4. ¿Qué perfil de velocidad se desea tener en el LUNAbot W6 VJ1.0? (Página 57)
5. ¿Cómo se diseña el control del LUNAbot W6 VJ1.0? (Página 43)
6. ¿En qué elemento del LUNAbot W6 VJ1.0 se aplica la lógica difusa? (Página 57)

7. ¿Qué se requiere para que el control pueda llevarse a cabo con un microcontrolador? (Página 55)
8. ¿Qué elementos se requieren para construir físicamente el control del LUNAbot W6 VJ1.0? (Página 52)

### **1.1.3. Objetivos de la investigación**

#### **1.1.3.1. Objetivo general**

Diseñar el sistema de control electrónico para un robot caminante de entretenimiento tipo LUNAbot W6 VJ1.0 de forma que pueda realizar movimientos hacia adelante, atrás, izquierda, derecha y dar giros.

#### **1.1.3.2. Objetivos específicos**

1. Analizar el mecanismo del LUNAbot W6 VJ1.0.
2. Analizar el perfil de velocidad deseado en el LUNAbot W6 VJ1.0 para obtener las menores oscilaciones.
3. Diseñar las metodologías de movimiento del LUNAbot W6 VJ1.0 para determinar las secuencias de movimiento.
4. Diseñar el sistema de control del robot utilizando lógica difusa para que responda al perfil de velocidad deseado.
5. Programar las metodologías y el sistema de control de lógica difusa para poder ser utilizado en un microcontrolador.
6. Diseñar el circuito electrónico necesario para operar el robot utilizando el programa realizado.

### **1.1.4. Hipótesis de la investigación**

#### **1.1.4.1. Hipótesis general**

Utilizando el método de lógica difusa es posible diseñar un sistema de control electrónico para el robot LUNAbot W6 VJ1.0.

#### **1.1.4.2. Hipótesis específicas**

1. El mecanismo del robot se puede analizar y es controlable.
2. Se puede implementar un sistema de control al mecanismo para que se mueva con un perfil de velocidad deseado.
3. Se pueden crear metodologías de movimiento para el robot.
4. Se puede diseñar el sistema de control del robot utilizando lógica difusa para que responda el perfil de velocidad deseado.
5. Se pueden programar las metodologías y el sistema de control de lógica difusa para poder ser utilizado en un microcontrolador.
6. Se puede diseñar el circuito electrónico necesario para operar el robot utilizando el programa realizado.

#### **1.1.5. Justificación**

##### **1.1.5.1. Trascendencia**

*“La dominación de las personas con discapacidad es producida a través de juegos de reconocimiento, desconocimiento y desprecio. El dolor social inmoviliza y es la contracara cómplice y necesaria de la mercantilización del mundo donde los cuerpos pierden sus energías. Encarnar un cuerpo discapacitado, en tanto cuerpo superfluo, implica vivir en el mundo del no: no poder viajar en transporte público, no poder acceder a los beneficios sociales estatalmente reconocidos, no poder acceder al mercado matrimonial, en fin, no poder vivir” [17]*

De acuerdo a la “Ley general para la inclusión de las personas con discapacidad” emitida por el Congreso de la Unión, las personas con discapacidad tienen derecho a la accesibilidad. Se contempla que ésta sea de carácter universal, obligatoria y adaptada para todas las personas y que incluya el uso de tecnologías, ayudas técnicas y otros apoyos [18].

La plena integración de las personas con discapacidad al bienestar y al desarrollo depende de la generación pública de capacidades y oportunidades de superar todo tipo de barreras. [19]

La creación de un transporte de asistencia médica que permita a las personas con discapacidades motrices tener un desplazamiento eficaz con un mínimo esfuerzo incluso en terrenos irregulares, es la contribución a la sociedad que se planea lograr con el producto final del conjunto de trabajos del cual forma parte esta investigación. Para poder lograr construir el transporte de asistencia médica es necesario pasar por un proceso de diseño. En este proceso se contempla la construcción de un modelo a escala del transporte final que permita estudiar su comportamiento antes de realizar una construcción que requiera una inversión mayor. El diseño del control electrónico que permita el movimiento de este modelo a escala es el que se trata en esta investigación y fijará un precedente para el control del transporte de asistencia médica de tamaño real.

#### **1.1.5.2. Importancia**

El valor de esta investigación radica en que, al realizar el control del LUNAbot W6 VJ1.0, se establecerán bases para el diseño del control del LUNAbot de asistencia médica, el cual tras su construcción permitirá tener una opción más para la accesibilidad de las personas con discapacidades motrices. También aportará información en el uso de lógica difusa para el diseño de sistemas de control en el área médica y de robots hexápodos, logrando así sustentar cada vez más el uso de este método.

#### **1.1.6. Horizonte temporal y espacial**

El horizonte temporal de esta investigación se ubicó hace dos años, en el momento en el que surge la necesidad de movilidad en el robot LUNAbot W6 VJ1.0. Es en ese momento en que se comenzó a concebir la idea de un control electrónico. El horizonte espacial comprende el territorio nacional ya que el desarrollo del robot se llevará a cabo en este país, además de que fue pensado para resolver un problema de accesibilidad universal para personas con

discapacidades motrices en las condiciones que se encuentran presentes en México.

#### **1.1.6.1. Viabilidad de la investigación**

Los recursos materiales son suficientes, y se cuenta con el software, licencias, equipo y espacios necesarios para llevar a cabo el proyecto. De igual forma los recursos humanos se encuentran constantemente disponibles por medio del alumno, asesor, coasesor y sinodales.

#### **1.1.7. Alcance de la investigación**

Esta investigación se acotó en función de un alcance:

1. Exploratorio
2. Descriptivo

De tipo exploratorio porque los proyectos desarrollados actualmente no contemplan el control del mecanismo Peaucellier-Lipkin como locomotor, o para que deba seguir un perfil de velocidad trapezoidal utilizando lógica difusa aplicada directamente al mecanismo.

De tipo descriptivo por la exploración en el comportamiento del perfil de velocidad del mecanismo Peaucellier-Lipkin utilizado como sistema locomotor. En el mismo sentido describe las variables necesarias para su control y los elementos requeridos para el diseño del circuito electrónico.

#### **1.1.8. Alcances y limitaciones de la investigación**

##### **1.1.9. Alcances**

En esta investigación se tomará en cuenta el análisis del perfil de velocidad del mecanismo Peaucellier-Lipkin. El diseño del sistema de control del hexápodo

incluirá la selección de los motores, el microcontrolador y la fuente de alimentación que se utilizarán como referencia para determinar las variables a control. Se contempla también la codificación del sistema de control y el diseño del circuito electrónico del control incluyendo un esquema en el que se muestren los componentes necesarios y las conexiones que existen entre ellos. De igual forma se realizará el bosquejo de la ubicación de los elementos y las pistas de conexión (board).

#### **1.1.10. Limitaciones**

La limitante que se presentó fue el tiempo disponible para realizar la investigación, por eso se decidió no incorporar en la investigación los análisis estructurales, de esfuerzo o el diseño mecánico del robot, tampoco los estudios económicos o de mercado para determinar el costo del robot o de sus elementos.

### **1.2. Identificación de variables**

#### **1.2.1. Variables dependientes**

La variable dependiente de acuerdo a la hipótesis planteada será el sistema de control por medio de lógica difusa que involucra el análisis de movimientos, el mecanismo de la pierna, el mecanismo de la pata, el perfil de velocidad de salida, el método de diseño de control, la codificación del control, y los elementos electrónicos del control.

#### **1.2.2. Variables independientes**

De acuerdo a lo establecido en la hipótesis y objetivos de la investigación se desprenden las siguientes variables independientes:

1. Movimientos que realiza el LUNAbot W6 VJ1.0.
2. Mecanismo de la pierna del LUNAbot W6 VJ1.0.
3. Mecanismo de la pata del LUNAbot W6 VJ1.0.

4. Perfil de velocidad deseado del LUNAbot W6 VJ1.0.
5. Método de diseño de control del LUNAbot W6 VJ1.0.
6. Codificación del control del LUNAbot W6 VJ1.0.
7. Elementos físicos del control del LUNAbot W6 VJ1.0.

## Capítulo 2. MARCO TEÓRICO

### 2.1. El control y su evolución

Desde hace más de 2000 años el control automático de sistemas retroalimentados ha estado presente. Desde las civilizaciones antiguas como los griegos y los árabes, e investigadores más modernos destacando los de nacionalidad francesa e inglesa, han desarrollado sistemas de control automático más sofisticados. Sin embargo fue a mediados del siglo XIX que el diseño de control exigió describir los controladores mecánicos a partir de ecuaciones matemáticas.

En 1868 James Clerk Maxwell, publica su artículo titulado “*On Governors*”, en el que describe como derivar las ecuaciones diferenciales para varios controladores mecánicos [20]. Analiza la dinámica del controlador, impulsado por un experimento eléctrico en el cual la velocidad de rotación de una bobina debía permanecer constante. Maxwell obtiene un modelo lineal de tercer orden y las condiciones correctas para estabilidad en términos de coeficientes de las ecuaciones características. Incapaz de obtener una solución a modelos de mayor orden, expresa su esperanza a que esa pregunta atraiga la atención de los matemáticos [21].

En ese momento los matemáticos y físicos sabían que la estabilidad de un sistema dinámico estaba determinada por la localización de las raíces de las ecuaciones características y que un sistema era inestable cuando la parte real de

una raíz compleja era positiva. El problema era cómo determinar la localización de las partes reales de una raíz compleja sin encontrar las raíces de la ecuación. Maxwell demuestra para sistemas de segundo, tercer y cuarto orden que, examinando los coeficientes de las ecuaciones diferenciales, la estabilidad de los sistemas podía ser determinada. En 1875 Edward James Routh [22] publica “*The criterion of dynamical stability*”, en el que plantea una solución para los sistemas de quinto orden. En 1877 produce un extenso tratado sobre “La estabilidad del movimiento” en el que, basado en el trabajo de Augustin-Louis Cauchy y Charles Sturm, expone lo que se conoce hasta el día de hoy como el criterio de estabilidad de Routh-Hurwitz [23].

De forma paralela en el continente Europeo un trabajo relacionado se llevaba a cabo por I.A. Vyshnegradskii (traducido como Wischnegradki) en San Petersburgo, el cual apareció en la “*Comptes Rendus de l’Academie des Sciences*” en Francia en el año de 1876. Vyshnegradskii transformó un modelo de ecuaciones diferenciales de tercer orden de un motor de vapor en una forma estándar:

$$\varphi^3 + x\varphi^2 + y\varphi + 1 = 0 \qquad \text{Ecuación 1}$$

En dónde  $x$  y  $y$  son conocidos como los parámetros de Vyshnegradskii. Demostró que el punto en el plano  $xy$  define la naturaleza de la respuesta transitoria del sistema. En 1911 Enrico Bompiani (1889-1975) demuestra que tanto el criterio de Vyshnegradskii como el de Routh-Hurwitz son idénticos [24].

Las aplicaciones de los sistemas de control se hicieron aparentes conforme crecía el conocimiento de la electricidad y sus aplicaciones. Por ejemplo, las lámparas de arco requerían que la separación entre electrodos se mantuviera constante y generalmente era útil para todos los usuarios si tanto el voltaje como la corriente del dispositivo eléctrico se mantenían constantes. La electricidad también proporcionó herramientas adicionales para medición, transmisión, manipulación de señales y actuadores. El relé eléctrico, que proveía amplificación

de potencia de alta ganancia, y el solenoide de resorte, que proveía una acción proporcional de control, fueron dispositivos significativos [25].

En el siglo XX se aplicó y esparció rápidamente los controles de retroalimentación de regulación de voltaje, corriente, y frecuencia; el control de caldera para generación de voltaje; el control de velocidad de un motor eléctrico; y control de temperatura, presión y flujo en procesos industriales.

Durante las primeras décadas del siglo XX se incrementó el uso de los giroscopios para la estabilización de naves y pilotos automáticos. Elmer Sperry (1860-1930) fue pionero de la estabilización activa, el girocompás, y el giroscopio de piloto automático, haciendo sus patentes entre 1907 y 1914. El piloto automático de Sperry tenía un ciclo interno controlando a un motor eléctrico el cual operaba a un motor maestro, mientras que el ciclo externo utilizaba un girocompás para sensar el calentamiento. Sperry también diseñó un anticipador para replicar la forma en que un experimentado timonel manejaba el timón para evitar sobreviraje. Por el mismo periodo Sperry y su hijo Lawrence (1892-1923) diseñaron autoestabilizadores aéreos adicionando la complejidad del control tridimensional [26]. Sperry incorpora el control PID (Proporcional Integral Derivativo) y el ajuste automático de ganancia para compensar las perturbaciones causadas cuando las condiciones del mar cambiaban. Otro elemento que ocupaba a las compañías de suministro eléctrico era lograr la operación económica de las calderas de generación de vapor. El control de calderas es un problema multivariable ya que se requiere el control del nivel del agua y de la presión del vapor, y para tener una eficiente combustión la corriente de aire a la caldera tenía que ser también controlada [27].

Conforme los dispositivos y sistemas comenzaron a ser utilizados en diferentes áreas de la ingeniería, dos grandes problemas aparecieron: el primero fue la falta de entendimiento teórico sin un lenguaje común para discutir los problemas, y el segundo que no había métodos de análisis y diseño sencillos que pudieran aplicarse de forma simple.

Durante la Segunda Guerra Mundial el desarrollo de los sistemas de control se enfocó a las armas de guerra y el control de disparo, ya que para realizar estas actividades se requerían al menos catorce operadores, por lo que la automatización era fundamental. Sin embargo, debido a los conflictos bélicos, las publicaciones de las investigaciones permanecían ocultas. Fueron tres países los que lograron avances significativos: la Unión Soviética, Alemania y Estados Unidos, siendo este último el país con resultados más notables [28]. En Inglaterra en 1942 se crea el cuerpo de Servo-Panel como resultado de la iniciativa de Solomon, líder del grupo especial de radar en Malvern. Este cuerpo actuó como sociedad de aprendizaje con reuniones mensuales desde mayo de 1942 hasta agosto de 1945. También Alemania realizó aportaciones importantes en el área de control. En 1939 la “Verein Deutscher Ingenieure” una de las dos más grandes asociaciones de ingenieros, formaron un comité de especialistas en control y en 1940, el presidente de dicho comité Herman Schmidt, dio una conferencia tratando la relación de la ingeniería de control con la economía, las ciencias sociales y aspectos culturales. En Rusia se crearon dos instituciones auspiciadas por la sociedad soviética de ciencias, la KTA en 1934 y la IAT fundada en 1939. La primera era responsable de manufacturar equipo de control y traducir artículos.

En EUA tres laboratorios de investigación fueron destinados a la investigación del control. El primero de ellos era el laboratorio de servomecanismos del departamento de Ingeniería Eléctrica del Instituto Tecnológico de Massachusetts (MIT). En él se reunieron las mentes de Brown, Hall, Forrester y Hazen, quienes desarrollaron métodos de diseño de control de lazo para servos de alto rendimiento. Ellos utilizaron métodos basados en el dominio del tiempo, comenzaron el desarrollo del uso de diagramas de bloques, y emplearon un analizador diferencial para simular sistemas de control.

El segundo grupo importante fue Laboratorios Bell, en el que colaboraba gente como Bode, Shannon y Weaver en colaboración con Wiener y Bigelow quienes trataban temas como la teoría del suavizado y la predicción para apuntar las

armas. Norbert Wiener propuso predecir la posición futura de una aeronave. Su propuesta estuvo basada en un trabajo previo de análisis armónico generalizado.

El tercer grupo fue el laboratorio de radiación también del MIT en donde se realizaban trabajos sobre radares y navegación a larga distancia. Estaba integrado por James, Nichols, Ziegler y Phillips, quienes trabajaban en desarrollar técnicas de radares autodetectores para armas antiaéreas. En 1942, Ziegler y Nichols publicaron artículos en donde se describía cómo encontrar los parámetros óptimos para controles PI y PID, llamadas reglas de sintonía de Ziengler y Nichols.

En 1940 Gordon S. Brown analizó a detalle la respuesta transitoria de un sistema de lazo cerrado introduciendo el operador de sistema  $1/(1 + \text{lazo abierto})$  como funciones del operador diferencial  $p$  de Heaviside [29]. En 1943 Albert C. Hall demostró que tratando los bloques de control como funciones de transferencia y derivando la frecuencia de respuesta de una unidad retroalimentada como  $KG(i\omega)/[1 + KG(i\omega)]$ , el lugar geométrico de los sistemas de transferencia podía ser dibujado y, por lo tanto, el criterio de estabilidad de Nyquist podría ser utilizado, el margen de ganancia y fase podría ser determinado e introdujo el uso de los círculos M y N que permitían realizar estimaciones del comportamiento que tendría en dominio del tiempo un sistema de lazo cerrado usando los decibeles como unidad de medida.

Para el final de la guerra, las técnicas de control, a excepción del método de localización de raíces de Evans, se habían establecido. Las metodologías de diseño eran para sistemas lineales de una sola entrada. Las técnicas de respuesta en frecuencia basadas en el uso de Nyquist, Bode, y Nichols, permitían establecer el comportamiento del sistema en términos de ancho de banda, resonancia y ganancia, y proveían una vista gráfica y pictórica. La aproximación alternativa basada en la solución de ecuaciones diferenciales utilizando la transformada de Laplace expresaban el comportamiento en el dominio del tiempo, porcentaje de rebasamiento, error en el estado estable y amortiguamiento [30].

Las bases del control de muestreo de datos fue desarrollado independientemente durante la Segunda Guerra Mundial en muchos países. La transformada  $z$  fue, a excepción del nombre, completamente descrita en un capítulo por *Hurewicz*. Tustin en el Reino Unido, desarrolló la transformación bilineal para modelos en dominio del tiempo, mientras que Oldenbourg y Sartorius utilizaron diferentes ecuaciones para modelar los mismos sistemas.

Ziegler y Nichols mostraron cómo elegir los parámetros de un controlador dado para obtener un desempeño óptimo de una estructura de control dada (PI, PID). De igual manera, las formas estándar de Whiteley permitían a los diseñadores elegir un comportamiento particular para un rango en el sistema.

Finalmente J.P. LaSalle en un artículo publicado en 1960, generalizó todos los resultados obtenidos con los diferentes modelos de sistemas y mostró que el único control óptimo que puede existir es el control bang-bang.

El problema más complicado fue cómo elegir una estructura de control que permitiera el mejor comportamiento y cómo definir cuál era el mejor comportamiento, por lo que un modelo de la planta era necesario.

La conferencia de “Control Automático” celebrada en Julio de 1951 en Cranfield Inglaterra, y el “Simposio de Respuesta en Frecuencia” realizado en diciembre de 1953 en Nueva York, marcaron los inicios de la transición a las teorías de control moderno y al final de la década el elemento final en el control clásico fue añadido, la técnica de localización de raíces de Evans, la cual permitía agregar puntos que cambiaran la posición de los polos como una función de lazo de ganancia para que fueran fácilmente dibujados [31].

La teoría de control moderno o teoría del espacio de estado se derivó del trabajo de Poincaré y Lyapunov realizado a finales del siglo XIX y fue posible hasta que el periodo de la postguerra terminó y los diarios Soviéticos introdujeron las traducciones. Los ingenieros que laboraban en las industrias aeroespaciales, debido a que los sistemas incrementaban su orden conforme el problema se volvía más complejo, siguieron estos trabajos y pasaron de plantear ecuaciones

diferenciales generales en términos de un conjunto de ecuaciones diferenciales de primer orden en términos de variables de estado. Las variables de estado permitieron una representación más sofisticada y dinámica que los modelos de sistemas de una sola salida, una sola entrada, y eran adecuadas para problemas multivariantes. En general la matriz utilizada es:

$$\mathbf{x} = \mathbf{Ax} + \mathbf{Bu} \quad \text{Ecuación 1}$$

$$\mathbf{y} = \mathbf{Cx} \quad \text{Ecuación 2}$$

Donde  $x$  son variables de estado,  $u$  las entradas y  $y$  las salidas [32].

Las operaciones esenciales para diseñar un sistema de control fueron cada vez más complejas y la necesidad de desarrollar técnicas de diseño sencillas limitaba la evolución de los sistemas. La introducción de tecnologías digitales a finales de los cincuenta provocó grandes cambios en el control automático.

La refinería de Texaco en Texas y la planta Monsanto en 1959 y 1960 respectivamente, fueron las primeras en aplicar los computadores digitales al proceso de control. Si bien a principio sólo monitoreaban y optimizaban el proceso, dejando el control a sistemas eléctricos, neumáticos o hidráulicos, a partir la segunda mitad de 1960, computadoras especializadas en el proceso de control comenzaron a realizar tanto DDC (control digital directo) como control de supervisión.

Pero el microprocesador influyó en muchas áreas. Los controladores digitales basados en microcontroladores fueron adoptados pronto ya que eran compactos, confiables e incluían una amplia selección de algoritmos de control, tenían una buena comunicación con las computadoras supervisoras y era más sencillo utilizar herramientas de programación y diagnóstico mediante una efectiva interface de operador.

Finalmente entre 1960 y 1970 técnicas más avanzadas de control fueron desarrolladas. El control adaptativo fue una de ellas, este control modifica el

algoritmo de acuerdo a las circunstancias y ofrece grandes posibilidades en la identificación de parámetros relevantes del sistema, toma de decisiones acerca de las modificaciones requeridas del algoritmo de control e implementando cambios. Técnicas de control óptimo y robusto fueron desarrolladas también, las más importantes fueron el LQG (diseño cuadrático lineal Gaussiano) y las aproximaciones  $H_\infty$  de 1960.

Un acercamiento muy distinto posible gracias a las computadoras modernas fue alejarse de los modelos puramente matemáticos del comportamiento del sistema y los algoritmos de control. La lógica difusa, por ejemplo, la acción de control está basada en un conjunto de reglas expresadas en términos de variables difusas [33].

Para esta investigación, los microcontroladores y la lógica difusa fueron fundamentales por lo que se describirán más a detalle a continuación.

## **2.2. Lógica difusa**

La lógica difusa es un tipo de lógica probabilística que trata con razonamiento aproximado en lugar de razonamiento exacto. Contrario a la lógica tradicional que trabaja con valores totalmente verdaderos o totalmente falsos, la lógica difusa contempla valores parciales de verdad, tomando un rango de valores entre cero y uno de acuerdo a la veracidad de la variable. Aunque la lógica probabilística también trata con variables que toman valores en un rango entre cero y uno, la diferencia yace en el concepto, ya que la lógica probabilística interpreta estos valores como el grado de probabilidad que tiene la variable en ocurrir mientras que la lógica difusa interpreta los valores como el grado de verdad que tiene la variable.

La lógica difusa nace en Estados Unidos en 1965 propuesta por Lotfali Asker Zadeh (1921-, Azerbaiyán). El trabajo de Lotfali propone el uso de funciones de membresía operando en el dominio de la probabilidad [34].

Este tipo de lógica utiliza variables lingüísticas que toman a su vez valores lingüísticos, lo que facilita expresar las reglas y los hechos. Por ejemplo si una variable es edad, ésta puede tomar valores como viejo o joven. Con estas variables lingüísticas se forman los predicados, que son las reglas que se seguirán. Siguiendo con el ejemplo anterior podemos establecer reglas como:

SI edad ES viejo ENTONCES bajar volumen de la música

SI edad ES joven ENTONCES subir volumen de la música

Existen también los operadores “Y”, “O”, y “NO”, pero se encuentran definidos como el máximo, mínimo y complemento. También existen operadores lingüísticos que cambian el significado de todo el predicado como son “muy”, “poco”, etc.

La lógica difusa puede ofrecer ciertas ventajas al momento de diseñar un control, ya que permite trabajar sin que exista previamente un modelo matemático del sistema que se desea controlar, sin embargo sí se requiere que la experiencia del diseñador con el sistema sea la suficiente como para conocer a la perfección su comportamiento. Un sistema de control con lógica difusa ofrece salidas de forma veloz y precisa, gracias a que el diseñador sabe exactamente cómo reaccionará el control respecto a salidas determinadas.

La primera aplicación de un sistema de control utilizando lógica difusa fue en Dinamarca en 1975, aplicado en un horno de una empresa cementera, mientras que en Estados Unidos estos controladores eran ignorados debido a su asociación con la inteligencia artificial, que en ese tiempo no tenía credibilidad por ser un concepto en exceso comercial [35].

Fue hasta 1985 que los japoneses Seiji Yasunobu y Soji Miyamoto [36] de la empresa Hitachi realizaron la primera simulación del sistema de control de aceleración y frenado del tren de Sendai, demostrando su superioridad al utilizar la lógica difusa. En 1987 se aplicaron sus ideas y el tren fue construido. También en 1987 en Tokio, Takeshi Yamakawa [37] aplica la lógica difusa en un sistema de control de un péndulo invertido montado sobre un carro que lo equilibraba moviéndose hacia adelante o hacia atrás. Fue gracias a estas demostraciones que los japoneses comienzan a aplicar ampliamente los sistemas de control difusos. En 1988 fundan un laboratorio encargado de realizar investigación en el área de la lógica difusa llamado LIFE, siglas que provienen de su nombre en inglés “Laboratory for International Fuzzy Engineering”. En este laboratorio colaboraban 48 compañías incluida la Volkswagen, que era la única empresa foránea e integró a su equipo de investigación durante tres años.

Son varias empresas ya las que trabajan con sistemas de lógica difusa, entre ellas Hitachi, Mitsubishi y Canon. También son diversas las aplicaciones en las que se maneja como son aspiradoras, cámaras de video, lavadoras, aire acondicionado, reconocimiento de escritura, sistemas ópticos, robots, helicópteros controlados por voz, manufactura de película, elevadores, y más [38, 39, 40, 41, 42].

### **2.3. Microcontroladores**

Un microcontrolador es un circuito integrado programable compuesto por un conjunto de bloques funcionales que cumplen tareas específicas, capaz de ejecutar órdenes grabadas previamente en su memoria. Dentro de un microcontrolador existen tres módulos principales que son idénticos a los que se pueden encontrar en una computadora: la unidad central de procesamiento, la memoria y los periféricos de entrada/salida.

Un microcontrolador no puede funcionar por sí solo, es necesario complementarlo con un circuito básico cuyo elemento más importante es un cristal

de cuarzo el cual funciona como un reloj. Este cristal determinará la velocidad a la que el microcontrolador ejecutará cada línea de instrucción dada. Las líneas de instrucción se deben programar desde un inicio por el usuario ya que en la fabricación el microcontrolador viene totalmente en blanco, es decir que no contiene ninguna instrucción en su memoria. Otro requerimiento indispensable para que el microcontrolador pueda funcionar es que cuente con un sistema de alimentación que ordinariamente es de 3 Vcd a 6 Vcd siendo 5 Vcd la alimentación generalmente utilizada.

Todas las instrucciones del microcontrolador son introducidas por el usuario mediante el uso de software especializado que permite la comunicación con él por medio de una computadora, sin embargo aunque estas instrucciones pueden ser programadas en una diversidad de lenguajes como son el lenguaje C y el ensamblador, es necesario codificar esa serie de instrucciones utilizando un sistema numérico hexadecimal [43].

Los microcontroladores son bastante utilizados en el área de diseño de control electrónico cuando los requerimientos del sistema a controlar no sobrepasan las características de fábrica de los microcontroladores comunes principalmente la velocidad para ejecutar instrucciones. Son el cerebro de nuestro control. Tiene la función de adquirir las señales provenientes del exterior o bien del usuario, procesar la información y brindar una señal de respuesta. Para sustituir su funcionamiento sería necesario introducir una cantidad de circuitos inmensa que elevaría el costo de producción y no lograría responder a la misma velocidad que lo haría un microcontrolador.

Algunas de las aplicaciones más actuales son por ejemplo la adquisición a distancia de datos de medidores analógicos, transmitiendo la señal adquirida mediante un microcontrolador por medio de radiofrecuencia desarrollado en Baréin, Asia. También en Cardiff, Inglaterra se ha trabajado desde el 2010 en un sistema de monitoreo basado en microcontroladores para sistemas de automatización de alta velocidad, en este trabajo se ha demostrado el potencial de

la solución de problemas basada en microcontroladores es accesible y muy poderosa.

Las aplicaciones de los microcontroladores en la vida diaria las podemos encontrar en abundancia si miramos hacia el campo automotriz. Actualmente el sistema electrónico de un automóvil tiene una infinidad de sensores que adquieren información de la temperatura, cantidad de aceite, gasolina, agua refrigerante, velocidad, y prácticamente cualquier cosa que pase con él. Quien se encarga de procesar toda esta información y hacer que el auto responda a las peticiones del usuario es precisamente un microcontrolador.

El uso de los microcontroladores en sistemas de control para máquinas caminantes se ha ido incrementando gracias a su fácil programación y variedad de modelos, es por ello que en esta investigación se utilizan como parte primordial del sistema de control.

## **2.4. Máquinas caminantes**

Un robot es un dispositivo mecánico o virtual que guiado por un programa de computadora o por circuitos electrónicos realiza tareas deseadas por el humano. [44]

La idea de los autómatas se crea en la mitología de diferentes culturas antiguas como la china, griega o egipcia, quienes intentaron crear seres auto-operables que se asemejaran a los animales o a los humanos. Uno de los primeros inventores que empieza a desarrollar máquinas autónomas es el griego Ctesibio, quien en el 270 d.C. crea relojes con figuras móviles. Posteriormente Arquitas de Tarento, también griego, diseña “El pichón”, una máquina impulsada por vapor. Es Herón de Alejandría quien describe autómatas más complejos impulsados por vapor, agua y aire. Más tarde Leonardo Da Vinci diseña diversos mecanismos con el fin de construir una máquina capaz de desplazarse sin

necesidad de asistirlo por un humano, y es entonces cuando bosqueja el robot humanoide llamado “El caballero” en 1495.

No solamente en Europa se desarrollan autómatas, también en Japón entre los siglos XVII y XVIII se construyen animales y humanoides autómatas llamados karakuri, utilizados como adornos, juguetes y en ceremonias religiosas.

Dentro de los constructores más destacados se encuentra Jacques de Vaucanson, quien en el siglo XVIII en Francia, construye tres famosos autómatas muy complejos, “El tamborilero”, “El flautista”, y su obra más compleja y famosa “El pato”. Este último contenía más de trescientas piezas en su ala y un mecanismo hidráulico. El pato podía graznar, comer, beber, digerir y defecar. [45]

Sin embargo, ninguno de estos diseños destaca en su época ya que estos artefactos eran considerados meramente juguetes.

En la actualidad, los robots se pueden clasificar de acuerdo a su arquitectura, la cantidad de extremidades, aplicación y evolución.

De acuerdo a su arquitectura se pueden dividir de la siguiente manera [46]:

- **Móviles:** son aquellos que pueden desplazarse. Su sistema locomotor es de tipo rodante y generalmente están basados en un carro o una plataforma. Se dirigen por medio de un control remoto o se les asigna un programa predeterminado para realizar su recorrido. Figura 2.1
- **Androides:** buscan emular el comportamiento y la forma humana. En estos robots lo más complejo ha sido el desarrollo de la locomoción bípeda. Figura 2.2
- **Poliarticulados:** son los robots manipuladores e industriales. Su característica principal es que se encuentran fijos, y sus articulaciones sólo pueden moverse dentro de un espacio específico. Tienen un grado de libertad limitado por el número de articulaciones. Figura 2.3
- **Zoomorfos:** imitan el comportamiento de los animales. Dentro de esta categoría se encuentran los no caminantes, basados en segmentos

cilíndricos realizan un movimiento relativo de rotación, y los caminantes, artrópodos capaces de moverse en diversos tipos de superficies. Figura 2.4

- **Híbridos:** son aquellos que integran a dos o más de los anteriores, como por ejemplo un robot poliarticulado montado sobre la plataforma de un robot móvil. Figura 2.5



Figura 2.1 Robot móvil



Figura 2.2 Robot androide



Figura 2.3 Robot poliarticulado



Figura 2.4 Robot zoomorfo



Figura 2.5 Robot Híbrido

De acuerdo a su aplicación tenemos los robots [46]:

- **De servicio:** realizan actividades que generan un beneficio a los humanos ya sea en el área del hogar, de la salud o realizando servicios profesionales en ambientes peligrosos que comprometen la integridad de la vida humana. Figura 2.6
- **De seguridad y espacio:** en donde se encuentran todos los robots de tierra, mar y aire, utilizados para garantizar la seguridad civil o militar o

para cumplir misiones determinadas incluyendo así también las misiones espaciales. Figura 2.7

- **Industriales:** son aquellos diseñados para efectuar tareas repetitivas de manera automática dentro del proceso de fabricación o manipulación. Figura 2.9
- **De entretenimiento:** su aplicación es únicamente para satisfacer la parte recreativa del ser humano. Figura 2.8



Figura 2.6 Robot de servicio



Figura 2.7 Robot de seguridad y espacio



Figura 2.9 Robot industrial



Figura 2.8 Robot de entretenimiento

En base a su evolución tenemos a los:

- **De primera generación:** Su sistema de control está basado en “paradas fijas” mecánicamente.
- **De segunda generación:** estos robots contienen sus instrucciones en un sistema de almacenamiento por lo que sus movimientos se encuentran preestablecidos por una secuencia numérica.

- **De tercera generación:** se pueden controlar por medio de una computadora utilizando un lenguaje de programación. Comienzan a percibir su entorno por medio de sensores.
- **De cuarta generación:** Son robots inteligentes con mejor percepción de su entorno. Utilizan modelos de conducta.
- **De quinta generación:** no sólo emplean modelos de conducta sino que aprenden del comportamiento humano.

En la actualidad existen dos tipos de clasificación de robots de acuerdo a su aplicación y son de autómatas de propósito general y robots de propósito específico. Los robots ahora se pueden diseñar para realizar una tarea específica o un grupo de tareas deseadas, pero pueden ser reprogramados para realizar otra tarea deseada, sin embargo están limitados por su forma física. Generalmente los robots de propósito general son capaces de navegar en su propio espacio y satisfacer sus necesidades de recarga y realizar algunas tareas simples sin la intervención de un usuario. Pueden ser complementados con software o diversos elementos físicos que optimizan su uso.

Existen robots que se utilizan en las fábricas como en la producción de automóviles o componentes electrónicos. También son utilizados como empacadores, cargadores, para pruebas espaciales, recolección de la cosecha de frutas o tareas domésticas. En tareas militares son utilizados para realizar tareas que pondrían en riesgo la vida humana de ser realizadas por soldados como son por ejemplo la desactivación de bombas, el control de aviones, el disparo de armas peligrosas o el reconocimiento de terreno. También son utilizados robots en la industria minera, para transporte de material o para taladrar las paredes y extraer material en lugares que comprometen las vidas de los trabajadores. En el área educativa los robots también han sido útiles al grado de llegar a ser tutores que ayudan a los niños a hacer su tarea o como ayudantes de profesores. En el área médica han sido muy útiles para el transporte de personas que presentan alguna discapacidad motriz, ya que pueden utilizarse como medio de transporte o

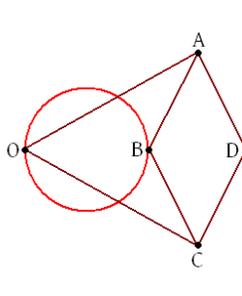
incluso como una articulación ya sea una mano, una pierna etc. También son usados en farmacias como surtidores de recetas para los farmacéuticos.

En nuestros días, los robots se han modificado de acuerdo a los materiales que existen en el mercado. Por ejemplo, se están desarrollando robots utilizando materiales como nanotubos de silicón llamados nanorobots, otros investigadores buscan crear robots que se puedan reconfigurar en su forma física para realizar tareas diferentes sin que se encuentren limitados por su estructura. Otra investigación busca crear robots con actuadores flexibles hechos de silicón con el fin de ampliar su movimiento. También se ha ampliado el campo de inspiración de los diseñadores y ahora se están creando robots que asemejan las colonias de hormigas, gusanos o abejas. Por último, se ha desarrollado la tecnología háptica aplicándola a los robots y así creando universos de realidad virtual más reales.

## **2.5. Mecanismo Peaucellier-Lipkin**

Este mecanismo surge en el año de 1864 diseñado por el oficial francés Charles-Nicolas Peaucellier (1832-1913) [47], y el lituano Yom Tov Lipman Lipkin (1846-1875) [48]. Fue el primer mecanismo plano capaz de obtener un movimiento recto a partir de uno circular y viceversa.

Consta de 6 barras en total, en el dibujo señaladas en color marrón. La longitud OC es igual a la longitud OA y las longitudes AB, BC, CD y DA son idénticas entre sí y forman un rombo. El punto O rota en su propio eje provocando que el punto B se mueva alrededor de un círculo imaginario descrito en rojo en la figura el punto D esta forzado a moverse en línea recta, mientras que si el punto B se mueve en línea recta el punto D se moverá en la trayectoria de un círculo. Figura 2.10



**Figura 2.10 Mecanismo Peaucellier-Lipkin**

Los triángulos ABC y ADC son congruentes porque B y D son congruentes entre sí. BA es congruente con BC y AD es congruente con CD por eso los ángulos ABD y CBD son iguales. Los triángulos OBA y OBC son congruentes ya que OA y OC son congruentes y por lo tanto O es colineal con B. BA y BC son congruentes y así los ángulos OBA y OBC son iguales. La suma de los ángulos OBA, ABD, DBC y CBO debe ser  $360^\circ$ . Pero como los ángulos OBA y OBC son iguales entre sí y los ángulos DBA y DBC son también iguales entre sí tenemos entonces que dos veces OBA más dos veces DBA suman  $360^\circ$ . De aquí se deduce que los ángulos OBA y DBA deben sumar  $180^\circ$ . Por lo tanto se puede determinar que O, B y D son colineales.

Si consideramos como P al punto formado por la intersección de las líneas AC y BD, P se encuentra a la mitad de ambos segmentos ya que ABCD es un rombo. Por lo tanto, la longitud BP es la misma que PD. El triángulo BPA es congruente con el triángulo DPA porque el lado BP es congruente con el lado DP, el lado AP es colineal con CD y el lado AB es congruente con el lado AD. Por lo tanto el ángulo BPA es igual al ángulo DPA, y como los ángulos BPA y DPA suman  $180^\circ$ , entonces dos veces el ángulo BPA suman  $180^\circ$ , y así BPA es igual a  $90^\circ$  y DPA también. Figura 2.11

Después de haber analizado lo anterior tenemos las herramientas necesarias para comenzar el desarrollo de la investigación. [49, 50, 51]

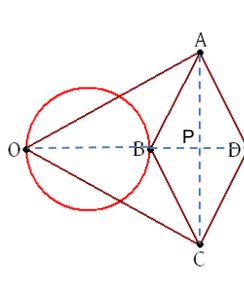


Figura 2.11 Peaucellier-Lipkin con líneas auxiliares.

## Capítulo 3. DISEÑO DEL CONTROL

### 3.1. Descripción final del robot

Como ya se ha mencionado anteriormente, el mecanismo sobre el que se basa el diseño del LUNAbot W6 VJ1.0 es el mecanismo Peaucellier-Lipkin. Cada pata del robot cuenta con un mecanismo similar al que se le han añadido dos barras (DE, EF) que permiten modificar la relación de tamaño que existen entre ellas (Figura 3.1).

De acuerdo a esta relación es posible que el robot en su totalidad realice un movimiento circular o uno rectilíneo. El diseño final de la estructura del robot nos permite apreciar que asemeja la morfología de un cangrejo, ya que sus patas se encuentran originalmente horizontales a diferencia de los arácnidos cuyas patas se ubican alrededor de su cuerpo (Figura 3.2). Partiendo de este diseño comenzamos el desarrollo del control. Para desarrollar este control se precisó analizar diversos factores como fueron el económico y el de factibilidad como los principales. Teniendo en cuenta estos factores se logró acotar el problema como se describe a continuación.

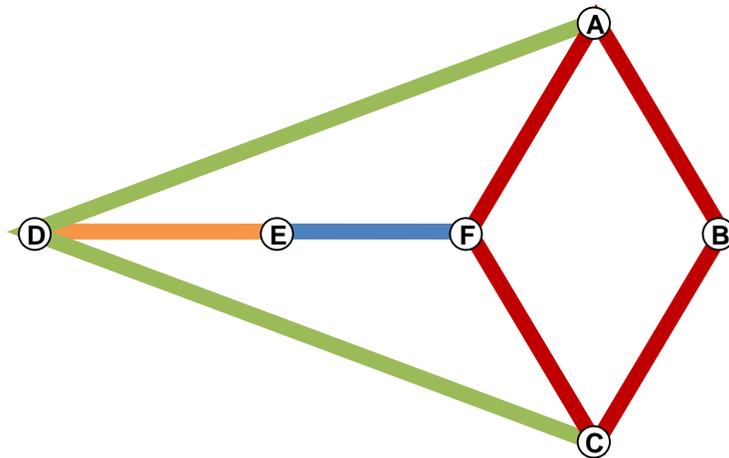


Figura 3.1 Mecanismo Peaucellier-Lipkin con las barras DE y EF añadidas.

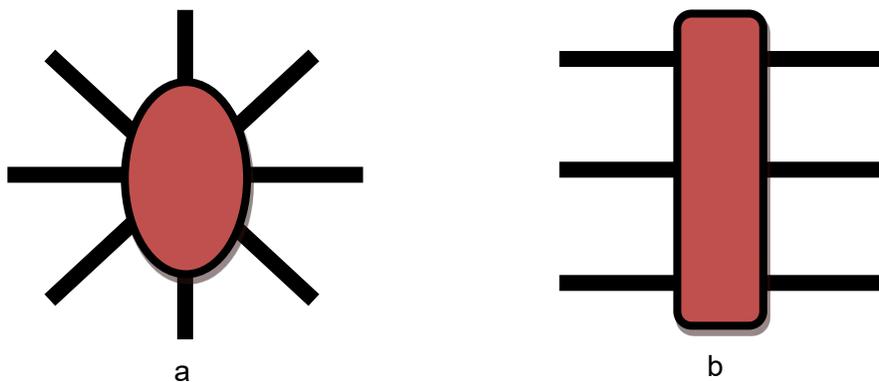


Figura 3.2 a) Distribución de extremidades en un arácnido. b) Distribución de extremidades en una morfología similar al cangrejo.

### 3.2. Definición de la forma de movimiento del robot

El LUNAbot W6 VJ1.0 es un robot que requiere de una metodología específica de movimiento ya que aunque asemeja una forma usada con anterioridad en la robótica como es la del cangrejo, el mecanismo de locomoción con el cual ha sido diseñado no se aplica actualmente a ningún otro robot. Por lo anterior fue necesario efectuar diversos análisis para conocer movimientos que el robot podría

realizar de acuerdo a la configuración y secuencia de movimiento de sus patas, llegando así a enlistar los siguientes:

- Movimiento rectilíneo hacia adelante y hacia atrás. (Figura 3.3 a)
- Movimiento circular con eje variable dentro (Figura 3.3 b) y fuera del robot en sentido de las manecillas del reloj y en sentido contrario. (Figura 3.3 c)

Para realizar el movimiento rectilíneo es necesario que las patas del robot estén alineadas horizontalmente y la relación de tamaño entre las barras DE Y EF sea de 1. Con esto podremos obtener un movimiento recto por medio de uno circular que es el objetivo original del mecanismo.

Para realizar el movimiento circular independientemente de la ubicación del eje de giro, es necesario que todas las patas del robot estén apuntando al eje seleccionado por medio de líneas imaginarias (Figura 3.4). También es necesario que la relación DE y FE sea menor a 1 para que cuando apliquemos el mismo movimiento de entrada al mecanismo Peaucellier-Lipkin lo que obtengamos sea un movimiento circular en torno al eje de giro elegido. Sin embargo al analizar los requerimientos necesarios de un robot de entretenimiento se definió que los movimientos circulares fueran limitados únicamente a un eje ubicado justo en el centro del robot. Esta elección se debe a que si el giro se realiza con un eje en cualquier otra posición, el espacio utilizado para realizar este giro será mayor, y en dado caso de encontrarse en un espacio reducido no sería sencillo girar el LUNAbot. Por lo tanto los movimientos finales elegidos para el LUNAbot W6 VJ1.0 son:

- Movimiento rectilíneo hacia adelante y hacia atrás
- Movimiento circular con eje de giro en el centro del robot en sentido horario y antihorario.

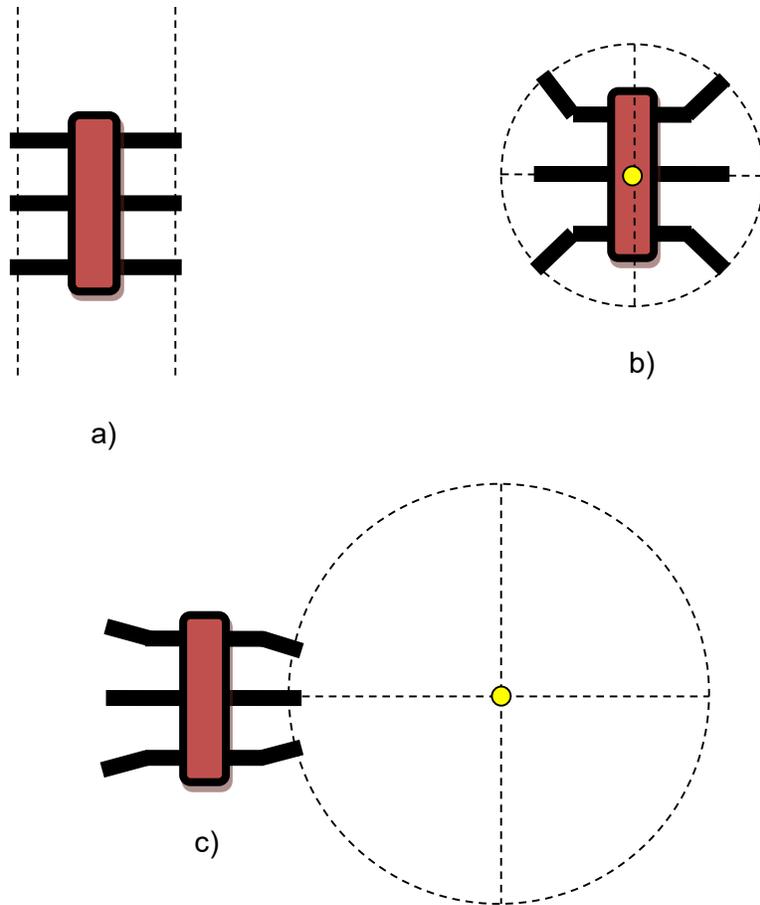


Figura 3.3 a) Movimiento rectilíneo hacia adelante y hacia atrás. b) Movimiento circular con centro de giro dentro del espacio físico del robot. c) Movimiento circular con centro de giro fuera del espacio físico del robot

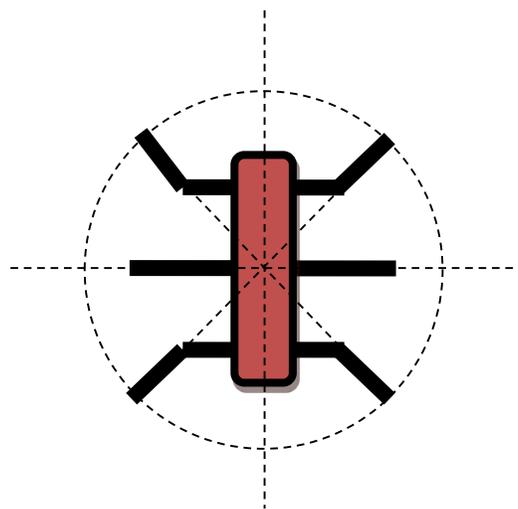
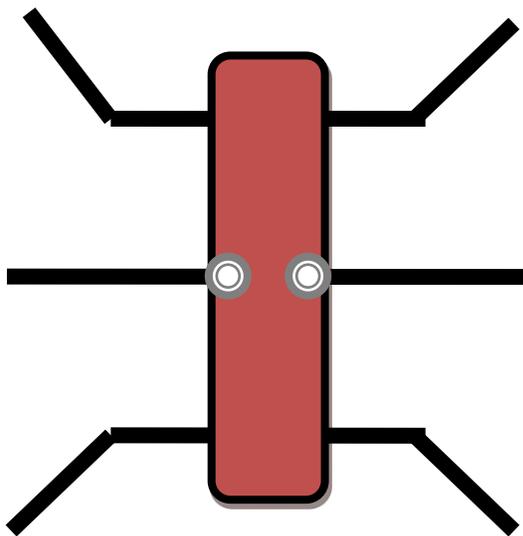
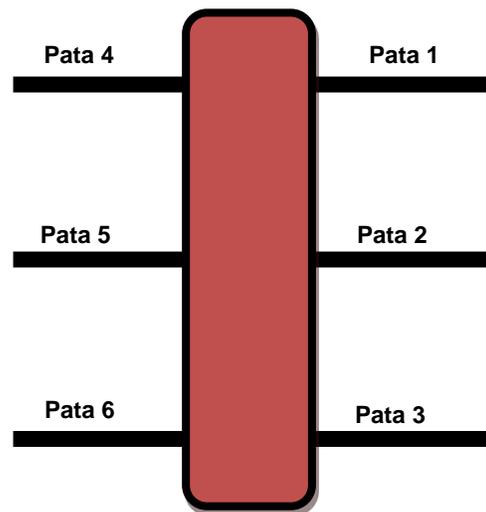


Figura 3.4 Líneas imaginarias mostrando el centro de giro del robot con las patas alineadas sobre las líneas.

Otra necesidad que debíamos recordar es que el robot debía mantener el precio más económico posible, por lo que fue necesario crear una solución para reducir la cantidad de motores en el mecanismo para ajustar la relación de tamaño entre las barras DE y EF. Inicialmente era considerado que se requerirían dos motores por cada una de las patas para asegurar que se mantendría la proporción siendo necesarios doce motores en un robot completo. Sin embargo ya que los giros del robot se encuentran limitados a tener un eje en su propio centro, la relación de tamaño entre DE y EF sólo variaría a un intervalo específico con un valor menor a 1 y no en un rango de 0 a 1. También gracias a la delimitación del movimiento se logró definir que ambas patas centrales no requerían ajustar su posición angular con respecto a la pelvis del robot, lo que permitió diseñar un mecanismo capaz de realizar el ajuste angular de las patas 1, 3, 4 y 6 (Figura 3.5) utilizando sólo dos motores para posicionar las piernas en el robot completo cuando inicialmente se consideraban seis (Figura 3.6).



**Figura 3.6 Motores de posicionamiento.** Estos motores se encuentran ubicados en el centro del cuerpo del robot.



**Figura 3.5 Nomenclatura de las patas**

Con todo lo anterior puede determinarse que serán requeridos dos motores por pata haciendo un total de doce motores en el robot para las piernas y patas. A estos doce motores se le sumarán dos motores indispensables para la configuración de la relación entre los segmentos DE y EF para realizar los dos tipos de movimientos. (Figura 3.7)

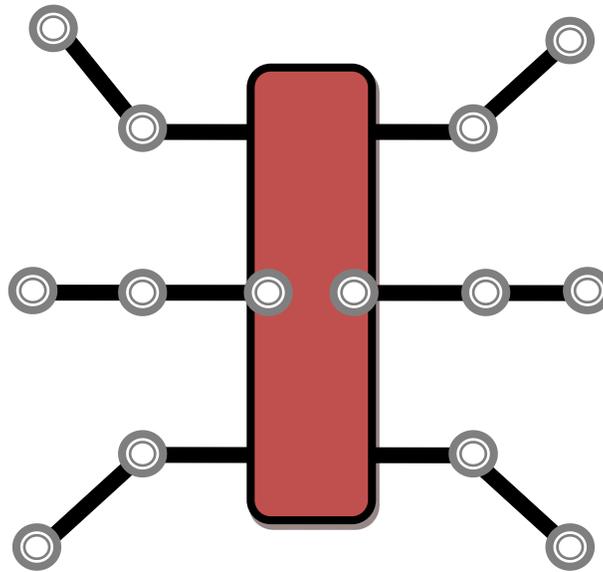


Figura 3.7 Cantidad y distribución de los motores del robot.

### 3.3. Diseño de las metodologías de movimiento

Al conocer cómo se movería el robot, fue necesario diseñar secuencias de movimiento para lograr el movimiento deseado en él. Para esto, fue forzoso definir una nomenclatura en cada uno de los elementos del robot (Figura 3.8). Teniendo la nomenclatura establecida, posteriormente definimos una serie de movimientos en los cuales lográbamos mantener siempre apoyadas tres patas del robot, mientras que las otras tres eran configuradas para lograr el movimiento rectilíneo o el circular (Tabla 3.1 y Tabla 3.2).

Al analizar las metodologías fue notoria la necesidad de determinar la posición de las patas del robot para definir la posición inicial de la misma, antes de entrar a

alguna de las secuencias anteriores, ya que para que se pueda realizar la configuración de las patas para comenzar el movimiento en línea curva, partiendo del movimiento en línea recta, es necesario que los segmentos DE y EF se encuentren completamente paralelas. De igual forma es necesario que se encuentren en la posición descrita anteriormente cuando se comienza un movimiento en línea recta, partiendo del movimiento en línea curva.

Por lo tanto, se realiza la propuesta de incluir un sensor sencillo que nos permitiera determinar la posición relativa entre DE y EF (Tabla 3.3). Se tabuló el funcionamiento del sensor, definiendo también que era necesario que existiera un sensor por cada pata, dando como resultado la necesidad de 6 sensores para un robot completo.

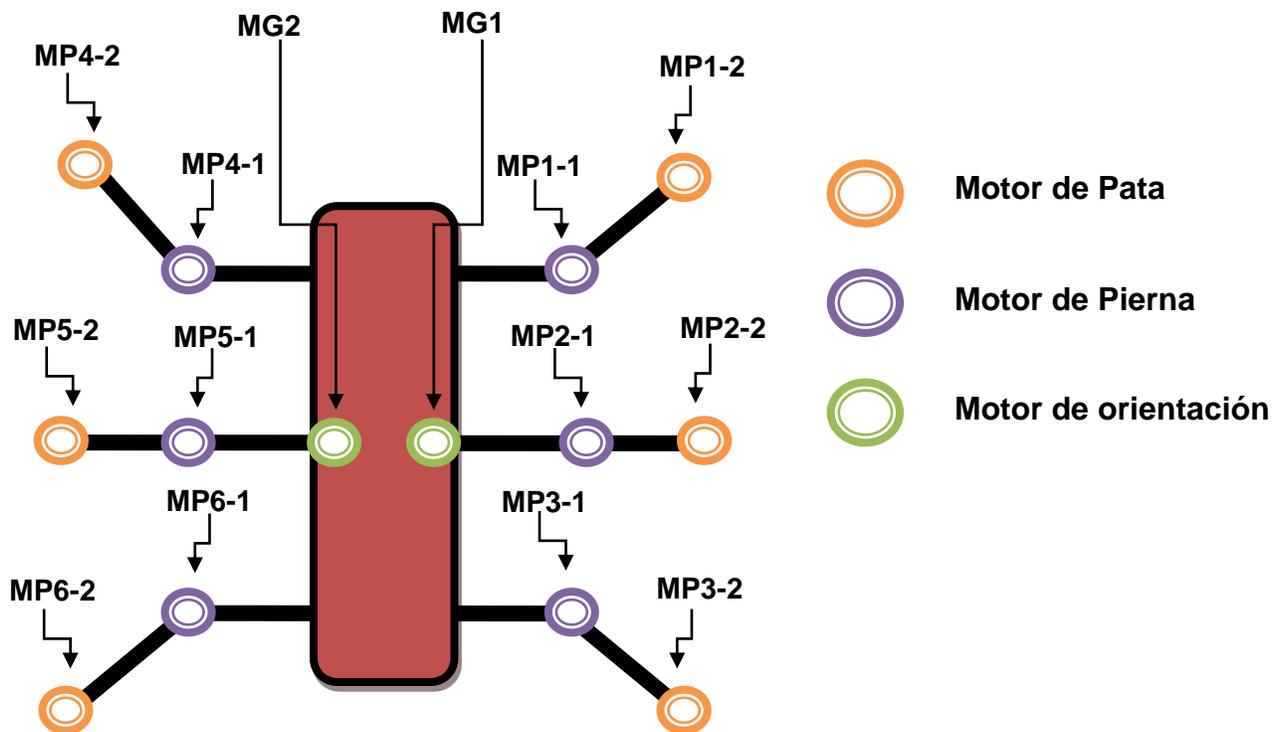


Figura 3.8 Primera nomenclatura utilizada para el diseño de metodologías.



**Tabla 3.3 Funcionamiento del sensor de posición relativa entre DE y EF.**

Sensor		
Pos. Motor	Control R.	Movimiento
0	0	1
0	1	1
1	0	0
1	1	1

Único caso en el que no hay movimiento

Utilizando entonces las metodologías antes propuestas se determinó el uso de un microcontrolador en el cual eran requeridos cuatro puertos de 8 bits cada uno para controlar el robot con todas las variables ya definidas. Para realizar estas secuencias propuestas para el microcontrolador se tomaron las metodologías (Tabla 3.1 y Tabla 3.2) y se agruparon las salidas que controlan a los motores de cada pata en grupos de cuatro. A cada grupo se le asigna un número hexadecimal obtenido a través de la codificación de los números binarios que se forman por cada uno de los grupos de cuatro de acuerdo al movimiento deseado (Tabla 3.4 y Tabla 3.5). Por ejemplo, para las patas 1 y 3 en el movimiento rectilíneo, el primer paso es activar el M2S. Tomando en cuenta que cada puerto del microcontrolador tiene 8 salidas que a la salida del puerto A veríamos un 0010 para la pata 1 y un 0010 para la pata 2, por lo que el valor hexadecimal asignado al puerto A sería 22.

**Tabla 3.4 Secuencia del microcontrolador para el movimiento rectilíneo.**

Puerto A		Puerto B		Puerto C		Puerto D	
Pata 1	Pata 3	Pata 4	Pata 6	Pata 2	Pata 5	Sensores	Alineación
0	0	0	0	0	0	1	0
2	2	0	0	0	2	1	0
8	8	0	0	0	4	1	0
1	1	0	0	0	1	1	0
0	0	2	2	2	0	1	0
4	4	4	4	8	8	1	0
0	0	1	1	1	0	1	0
2	2	0	0	0	2	1	0
8	8	8	8	4	4	1	0
1	1	1	1	1	1	2	1

Tabla 3.5 Secuencia del microcontrolador para el movimiento circular relativa entre DE y EF.

Puerto A		Puerto B		Puerto C		Puerto D	
Pata 1	Pata 3	Pata 4	Pata 6	Pata 2	Pata 5	Sensores	Alineación
0	0	0	0	0	0	1	0
0	0	2	2	0	0	1	0
0	0	0	0	0	0	1	2
0	0	1	1	0	0	1	0
2	2	0	0	0	2	1	0
0	0	0	0	0	0	1	4
8	8	0	0	0	4	1	0
1	1	0	0	0	1	1	0
0	0	2	2	2	0	1	0
4	4	4	4	8	8	1	0
0	0	1	1	1	0	1	0
2	2	0	0	0	2	1	0
8	8	8	8	4	4	1	0
1	1	1	1	1	1	2	1

### 3.4. Diseño de las metodologías finales de movimiento

Una vez definida la cantidad de actuadores necesarios para llevar a cabo el control del robot, se analizaron nuevamente las metodologías de movimiento y se analizó la posibilidad de reducir los puertos utilizados en el microcontrolador ya que se encontró que algunos de los motores realizaban movimientos paralelos. Por lo tanto, se definió que las salidas similares fueran agrupadas en una sola terminal del microcontrolador obteniendo así nuevas secuencias para el microcontrolador. Nuevamente lo primero que se requirió fue definir una nomenclatura para los motores finalmente establecidos en el robot (Figura 3.9).

Posteriormente se realizaron agrupaciones de los motores que realizarían movimientos de forma paralela. (Tabla 3.6)

Tabla 3.6 Definición de los grupos de motores.

GRUPOS				
G1	G2	G3	G4	G5
M1	M4	MP1	MS1	MS4
M5	M2	MP2	MS5	MS2
M3	M6		MS3	MS6

Cabe observar que aunque los motores MP1 y MP2 se encuentran agrupados debido a que sus movimientos son similares, en ningún momento pueden llegar a ser activados por la misma terminal ya que para realizar esta tarea sería necesario levantar las cuatro patas de los extremos del robot provocando así una pérdida de equilibrio.

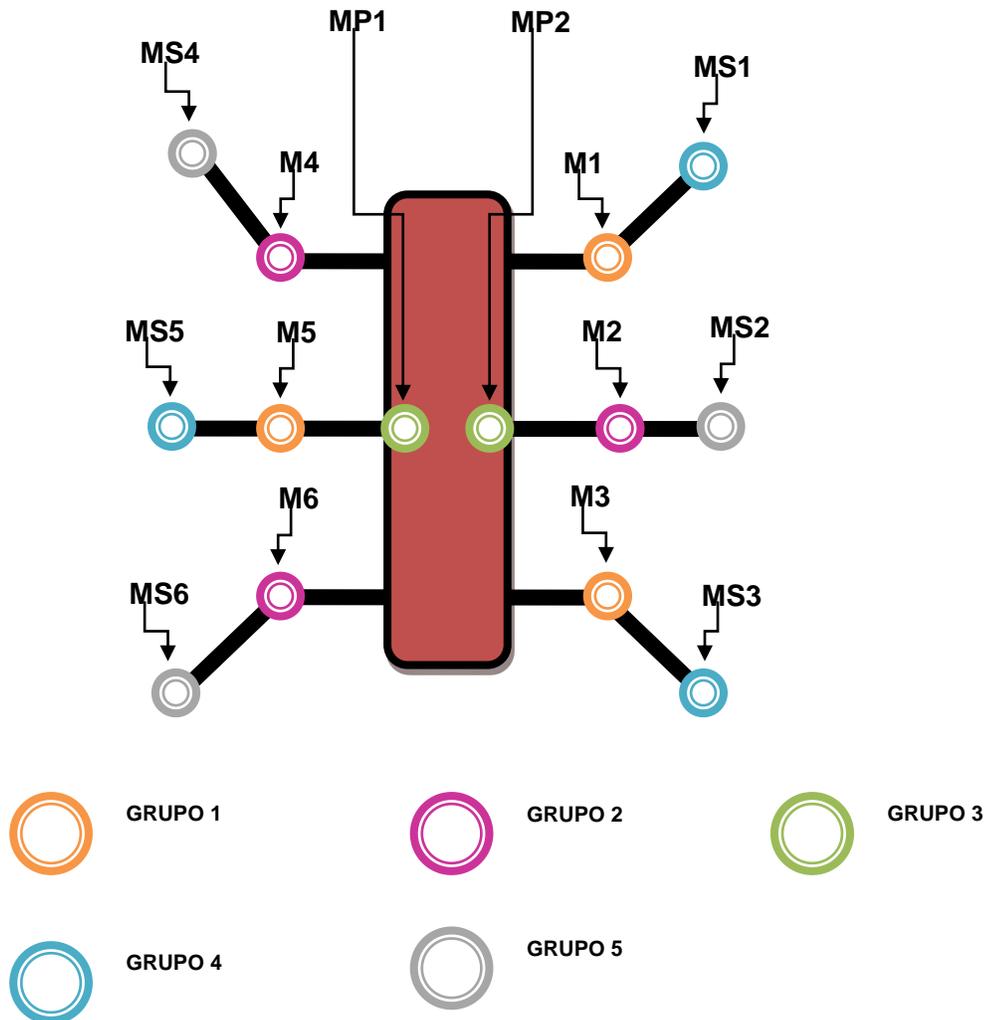


Figura 3.9 Nomenclatura de los motores

Una vez definidos los grupos que se utilizarán procedemos a definir las secuencias lógicas para realizar las metodologías de movimiento. Para el movimiento rectilíneo tenemos solamente dos pasos de inicialización antes de entrar a una secuencia de movimiento repetitivo que se mantendrá durante el tiempo en el que sea presionado el mando en el control. (Tabla 3.7 y Tabla 3.8)

**Tabla 3.7 Metodología de movimiento rectilíneo hacia adelante utilizando los grupos de motores.**  
S=Subir B=Bajar I=Izquierda D=Derecha

METODOLOGÍA LÍNEA RECTA ADELANTE			
G1	G2	G4	G5
		S	
I			
		B	
			S
D	I		
			B
		S	
I	D		

**Tabla 3.8 Metodología de movimiento rectilíneo hacia atrás utilizando los grupos de motores.** S=Subir  
B=Bajar I=Izquierda D=Derecha

METODOLOGÍA LÍNEA RECTA ATRÁS			
G1	G2	G4	G5
		S	
D			
		B	
			S
I	D		
			B
		S	
D	I		

El tercer renglón de las tablas anteriores señala el inicio de la secuencia que se repetirá, es decir, que pasando los dos primeros pasos de inicialización, se continúa hasta llegar al final de la tabla y una vez alcanzado el último paso, el robot repetirá la secuencia partiendo del tercer paso y continuará así, hasta que la instrucción de movimiento rectilíneo deje de ser enviada por el usuario.

También fue necesario crear metodologías ya con los motores agrupados para el movimiento circular en sentido horario y antihorario, logrando así completar las

metodologías de movimiento para el robot. Sin embargo, se puede notar que el giro puede realizarse replicando las metodologías hacia adelante y hacia atrás, solamente cambiando los pasos iniciales en los que se debe orientar las patas del robot para lograr que éstas apunten al centro del mismo. Por lo tanto la metodología de giro sólo implica la configuración de las patas (Tabla 3.9), y una vez terminada esta función, comienza una secuencia repetitiva igual al movimiento hacia adelante o hacia atrás dependiendo del sentido de giro que se desee.

**Tabla 3.9 Metodología de movimiento para la orientación de las piernas del robot para realizar un giro en cualquier sentido. S=Subir B=Bajar I=Izquierda D=Derecha**

METODOLOGÍA PARA GIRAR					
G1	G2	G4	G5	MP1	MP2
			S		
					D
			B		
		S			
				D	

Una vez diseñadas las metodologías de movimiento se observó la necesidad de crear una metodología de paro, en la cual se pudiera dejar al robot en una posición neutral, es decir, listo para poder iniciar cualquier tipo de movimiento sin importar cual haya sido la última instrucción. Esta posición neutral implica que los eslabones DE y EF de cada una de las patas del robot se encuentren a 90° de la pelvis. Al analizar la necesidad de que las 6 patas se encuentren apoyadas en el suelo, se estableció una secuencia que puede ser aplicada a partir de cualquier movimiento, ya sea rectilíneo o de giro (Tabla 3.10), para después solamente realizar pocos movimientos extras dependiendo de la configuración previa de las patas del robot.

**Tabla 3.10 Metodología para detener el movimiento del robot S=Subir B=Bajar I=Izquierda D=Derecha**

METODOLOGÍA DE PARADA				
G1	G2	G3	G4	G5
C				
			B	
				S
	C			
				B

Para poder observar mejor las metodologías del robot se realizó un diagrama de flujo incluyendo ya la lógica de programación que se empleará en el microcontrolador. Ya que contamos con secuencias repetitivas se pudieron crear tres funciones que facilitan la programación del robot, reduciendo líneas de programación y haciendo el programa más sencillo para su comprensión. Las funciones creadas fueron la función ADELANTE, que contiene el ciclo del movimiento rectilíneo de avance (Diagrama 3.1), la función ATRÁS, que contiene el ciclo del movimiento rectilíneo de retroceso (Diagrama 3.2), y la función PARO, que contiene la configuración que permite regresar al robot a una posición neutral con el fin de poder ser configurado para iniciar cualquier movimiento (Diagrama 3.3).

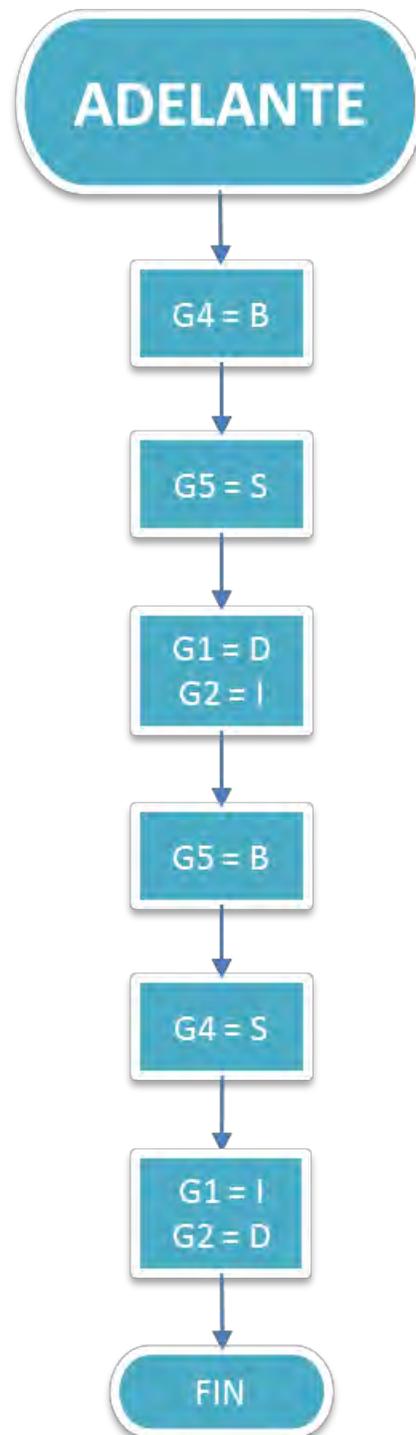


Diagrama 3.1 Diagrama para la secuencia de movimiento hacia adelante del robot.

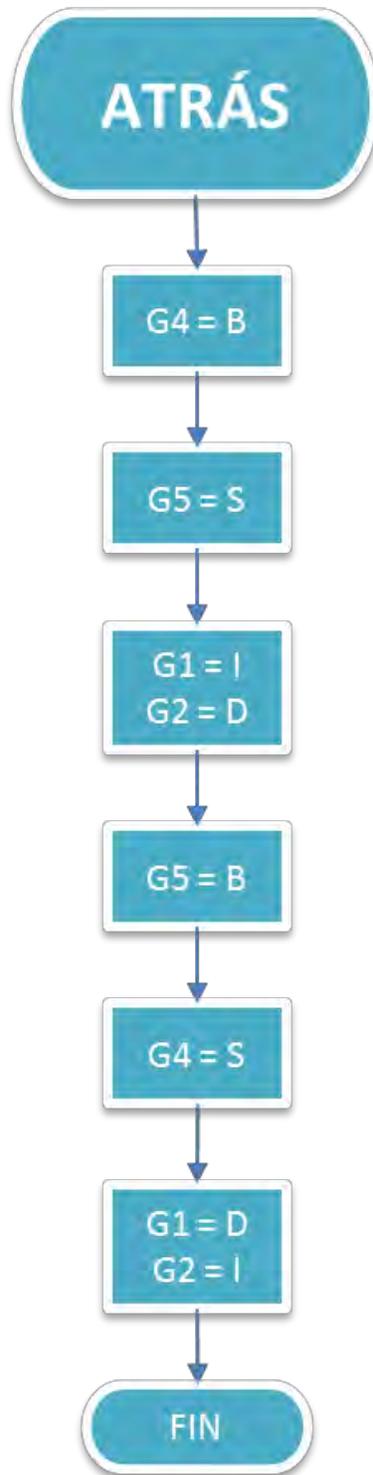
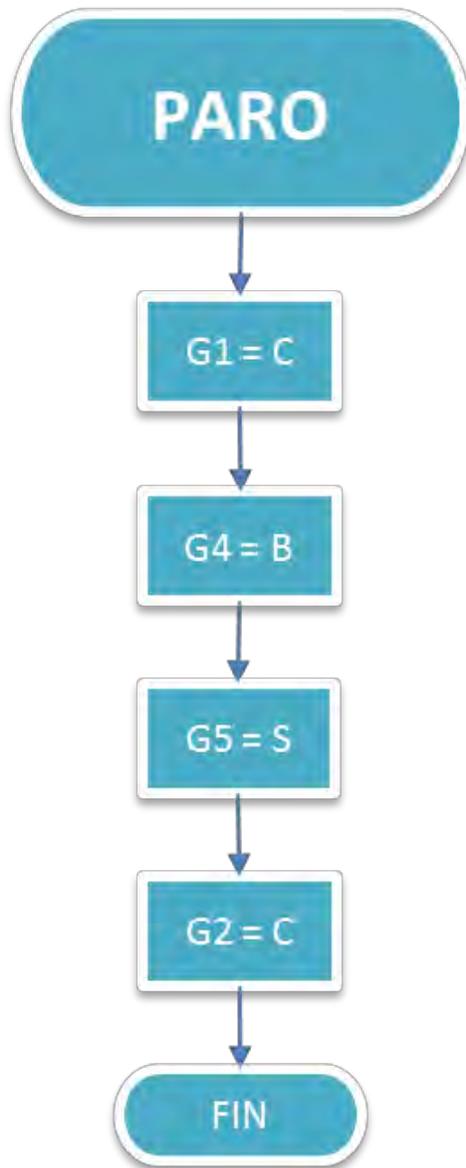


Diagrama 3.2 Diagrama para la secuencia de movimiento hacia atrás del robot.



**Diagrama 3.3 Diagrama para la secuencia de paro del robot.**

Una vez establecidas estas tres secuencias podemos definir finalmente un diagrama de flujo completo. (Diagrama 3.4)

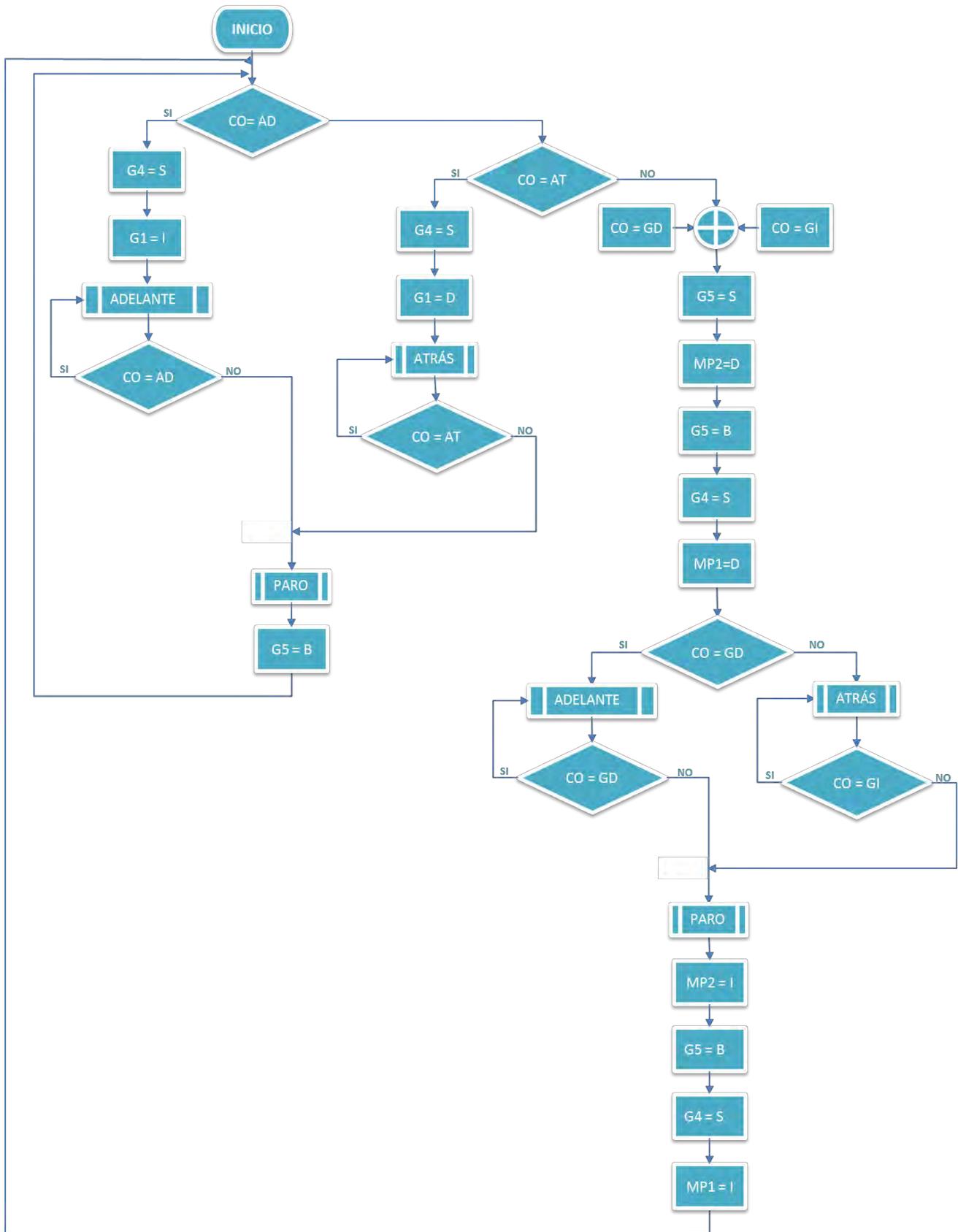


Diagrama 3.4 Diagrama para el movimiento en cualquier dirección del robot.

### 3.5. Elección del tipo de actuadores del robot

De acuerdo a la estructura del robot (Figura 3.10) y a las funciones que se requieren de los motores es necesario establecer el tipo de actuadores que serán utilizados para la locomoción del robot. Para poder realizar la elección de actuadores fue necesario conocer el peso del robot, ya que los actuadores debían ser capaces de moverlo. De acuerdo a las dimensiones ya conocidas del LUNAbot W6 VJ1.0, el peso total del robot es aproximadamente de tres kilogramos. Conforme a las metodologías ya definidas en el capítulo anterior, el robot debe tener siempre apoyadas tres patas para poder conservar el equilibrio y realizar sus desplazamientos, por lo tanto, el peso del robot siempre se encontrará distribuido entre tres patas. Teniendo en cuenta los datos anteriores se realizó la elección de actuadores.

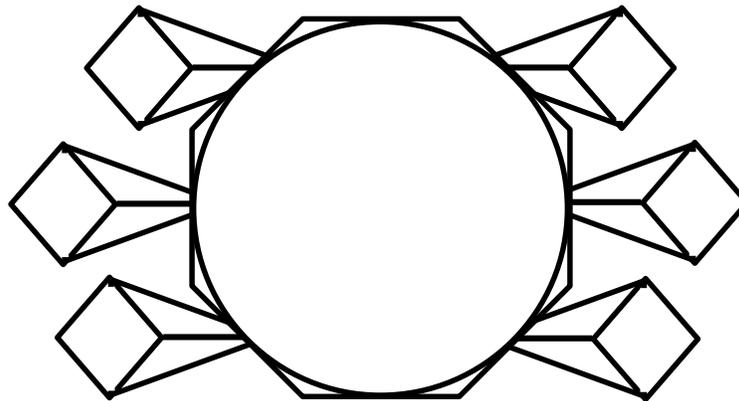
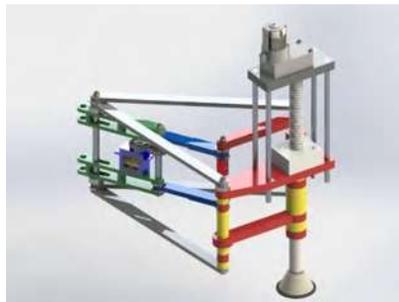


Figura 3.10 Forma final del Lunabot W6 VJ1.0

#### 3.5.1. Actuator de la pata

La pata del robot realiza un movimiento ascendente y descendente, que permite liberar o apoyar al robot en el suelo para lograr su movimiento. Inicialmente el diseño consideraba a esta pata como un mecanismo piñón-cremallera, sin embargo si esta consideración aplica, el robot requiere que todos los motores de las patas, en caso de utilizar motores de CD, estén

permanentemente energizados para lograr mantener al robot en posición de movimiento y evitar que su peso lo empujara hacia el suelo. Por lo tanto se rediseñó la pata sustituyendo la cremallera por un tornillo, cambiando así incluso la posición del motor que inicialmente estaría perpendicular a la dirección de la pata, ahora a una posición colineal, lo cual proporciona mayor comodidad en el cableado (Figura 3.12). Dado que el motor ya no será responsable de mantener la posición de la pata en un lugar fijo sino sólo de trasladarla a este lugar haciéndola girar, es posible utilizar motores de CD controlando la velocidad, tiempo y dirección de giro. Cumpliendo con las necesidades del fabricante de disponibilidad y bajo costo, se utilizará en el diseño el motor de CD FF-050SB11170 (Figura 3.11 y Figura 3.13). De acuerdo a las especificaciones del motor (Tabla 3.13), se consideró adecuado para utilizarlo en el robot, además de que el peso que agrega al robot es muy poco.



**Figura 3.12 Pierna del Lunabot. Se puede observar el tornillo de la pata y la posición del motor**



**Figura 3.11 Motor FF-050SB**

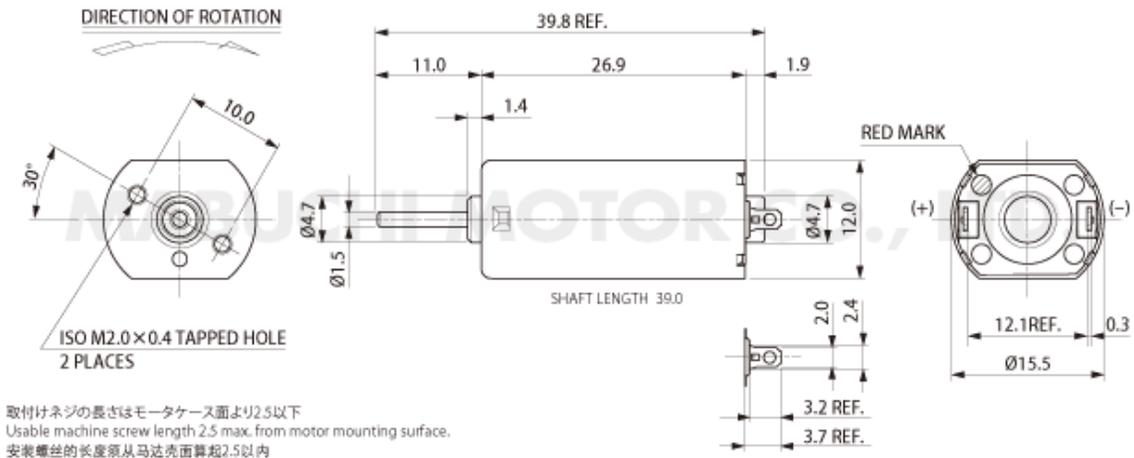


Figura 3.13 Forma y dimensiones del FF-050SB [52]

Tabla 3.11 Especificaciones del motor FF-050SB [52]

Motor FF-050SB
Rango de voltaje de trabajo: 2-8 Vcd
Torque: 45 g*cm
Velocidad: 10400 rpm
Consumo de corriente sin cargas: 46 mA
Consumo de corriente atrancado: 850 mA
Peso: 18 gr

### 3.5.2. Actuador del fémur y de configuración de giro

El LUNAbot W6 VJ1.0 requiere tener un movimiento que permita orientar sus fémures en una línea imaginaria que apunte al centro del robot. (Figura 3.4). También requiere que el movimiento de la pierna tenga tres estados (adelante, atrás, centro) (Figura 3.3) y que pueda determinarse exactamente en qué estado se encuentra el motor. De igual forma se quiere que los motores utilizados sean capaces de desplazar las piernas del robot, por lo que es necesario que cuenten con un torque considerable. Sabemos que el peso del robot será menor de tres kilogramos, y que el peso del robot siempre se encuentra distribuido mínimo en tres patas, por lo que cada una de las patas deberá ser capaz de desplazar un

tercio del peso del robot, es decir, cada motor deberá de tener como mínimo 1 kg\*cm. Tomando en cuenta las anteriores características fueron elegidos servomotores, los cuales permiten controlar su posición utilizando modulación de pulso, y nos brindan un torque suficiente. En ambos casos se utiliza el DS3115 Metal Digital Servo (Figura 3.14) (Tabla 3.12)

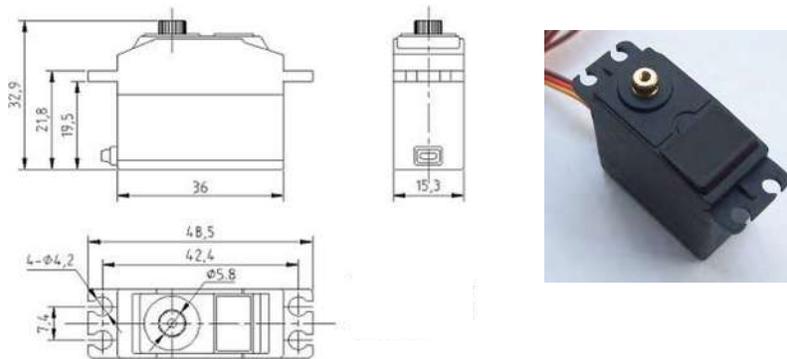


Figura 3.14 Forma y dimensiones del motor DS3115 Metal Digital Servo

Tabla 3.12 Especificaciones del motor DS3115 Metal Digital Servo [53]

DS3115 Metal Digital Servo
Rango de voltaje de trabajo: 4.8-7.2 Vcd
Torque: 13.5 kg*cm
Velocidad: 0.16 seg/60°
Corriente de operación: 150 mA
Peso: 60 gr

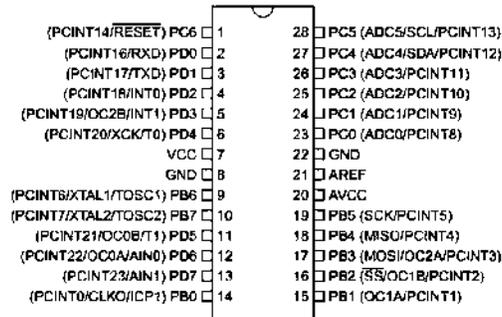
### 3.5.3. Elección del tipo de microcontrolador a utilizar

Conociendo ya las metodologías de movimiento y el tipo y cantidad de motores, se puede elegir el microcontrolador necesario para controlar los elementos ya determinados. Se eligió el microcontrolador ATmega328P de la compañía Atmel (Figura 3.15), ya que de acuerdo a sus especificaciones (Tabla 3.13), se satisface la cantidad de puertos y velocidad necesaria para nuestro control.

También se satisface la necesidad de un microcontrolador pequeño que pueda ser colocado fácilmente en el chasis del robot sin causar mayor problema de espacio, y de montaje sencillo lo que previene problemas de ensamblaje y permite su sustitución.

**Tabla 3.13 Especificaciones generales del Atmel ATmega328P [21]**

Atmel ATmega328P	
Voltaje de alimentación: 1.8 a 5.5 volts	
Frecuencia máxima de operación: 20 MHz	
Pines de entrada y salida: 23	



**Figura 3.15 Diagrama del microcontrolador ATmega328P [21]**

Una ventaja adicional que nos proporciona este microcontrolador es que se utiliza en las placas de arduino, permitiendo realizar la programación en este lenguaje. El lenguaje que utiliza la placa de arduino, Processing/Wiring, permite programar servomotores de forma sencilla utilizando las funciones preprogramadas del lenguaje.

## 3.6. Diseño del control de lógica difusa

### 3.6.1. Análisis del perfil de velocidad del mecanismo Peaucellier-Lipkin

Para el diseño del control utilizando lógica difusa no fue necesario contar con una ecuación matemática que describiera el sistema, en cambio se requirió conocer a profundidad el funcionamiento del sistema y así se utilizó la experiencia del diseñador para realizar el control. Lo primero que se requirió fue saber qué parte del mecanismo es la que se requería controlar y en qué lugar se debía realizar el control. Para el mecanismo aquí utilizado, la parte que se controló fue la pierna (Punto A), ya que ésta es la que realiza el movimiento que da origen al desplazamiento. Este desplazamiento debía ser suave para que se redujeran las perturbaciones en el robot. El extremo del mecanismo en donde se colocó la pata del robot (Punto B), es la que requería tener un desplazamiento suave, sin embargo, el control se realizó en el extremo opuesto del mecanismo en donde se utilizó un servomotor para realizar el movimiento necesario para mover la pierna. (Figura 3.16)

Una vez conocidos estos puntos se realizó un análisis del funcionamiento del mecanismo de forma experimental. Se observó la velocidad del punto B aplicando una velocidad constante de 9 rad/s en el punto A. (Figura 3.17)

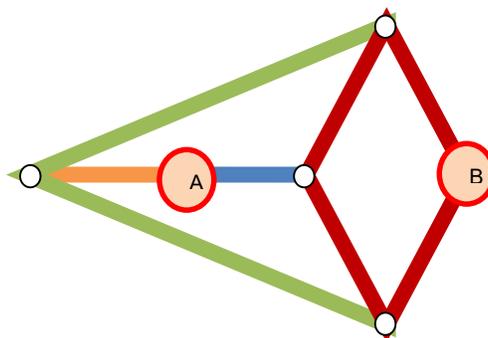
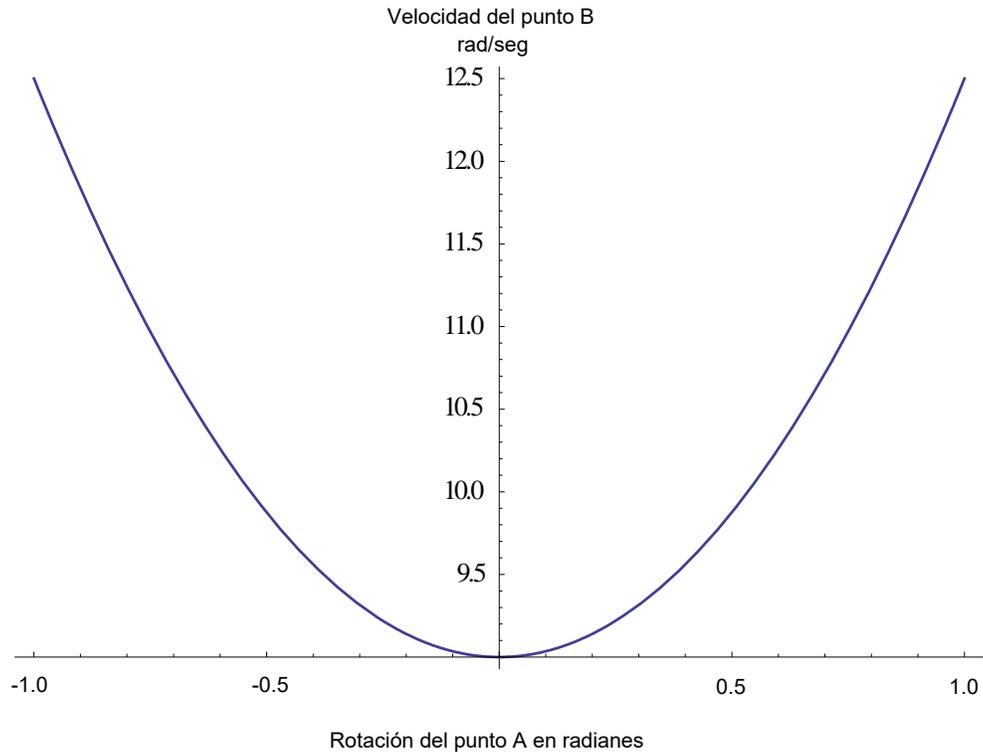


Figura 3.16 Mecanismo Peaucellier-Lipkin en el que se muestran los puntos involucrados en el control.  
A: Punto a controlar. B: Punto controlado.



**Figura 3.17 Velocidad del punto B aplicando al punto A una velocidad constante de 9 rad/s**

Se observó que a pesar de que la velocidad en el punto A es constante, en el punto B no es de esa forma (Figura 3.16). Se realizó un poco de análisis matemático a la gráfica y se determinó que se trata de una parábola que responde a la ecuación  $y = \frac{7}{2}x^2 + 9$ . (**Ecuación 4**)

Conociendo estos detalles se identifica que el mecanismo alcanza su velocidad mínima cuando se encuentra alineado en el centro de la pierna, y que la velocidad aumenta mientras se dirige a los extremos.

### 3.6.2. Diseño de las funciones de membresía

Una vez analizado el comportamiento del mecanismo y teniendo una gráfica del perfil de velocidad, se comenzó con el diseño del control de la pierna utilizando lógica difusa. Es importante mencionar que basta con diseñar el control para una pierna ya que es el mismo mecanismo y el mismo actuador para todas.

El primer paso para diseñar el control fue realizar la fuzzificación, esto es, tomar el rango de valores que presenta el mecanismo y convertirlo a valores difusos que podamos utilizar para nuestro control. Los valores difusos son aquellos que representan el rango de entrada de los valores del mecanismo con valores comprendidos entre el cero y el uno. Como el objetivo del control por medio de lógica difusa es utilizar variables lingüísticas para que el diseño del control sea más natural, el rango de los valores del perfil de velocidad se separó en cinco variables cuyo nombre y valor representa la velocidad. Las variables son:

1. Muy Lento
2. Lento
3. Medio
4. Rápido
5. Muy Rápido

Los rangos en los que se consideraron los intervalos de cambio de velocidad son los siguientes (Figura 3.18):

- a)  $11.24 < \text{Muy rápido} < 12.50$
- b)  $11.24 < \text{Rápido} < 10.26$
- c)  $10.26 < \text{Medio} < 9.56$
- d)  $9.56 < \text{Lento} < 9.14$
- e)  $9.14 < \text{Muy Lento} < 9.00$

Para crear las funciones de membresía se tomó en cuenta que para un mejor funcionamiento del control es necesario trabajar con el error. Por lo tanto se estableció un valor de referencia en 9 rad/s. Ese es el valor que tiene el punto más bajo de la parábola que describe el mecanismo Peaucellier-Lipkin, lo que permite trabajar ahora con un rango de 0 a 3.5 rad/s.

Estableciendo las funciones de membresía de entrada (Figura 3.19) se tiene que:

- a)  $2.24 < \text{Muy rápido} < 3.50$
- b)  $2.24 < \text{Rápido} < 1.26$
- c)  $1.26 < \text{Medio} < 0.56$
- d)  $0.56 < \text{Lento} < 0.14$
- e)  $0.14 < \text{Muy Lento} < 0$

El error se toma directamente del microcontrolador, ya que el servomotor permite conocer su posición actual y determinar su posición futura sin mayor problema debido a que se controla por ancho de pulso.

Teniendo ya las funciones de entrada se crearon las funciones de salida deseadas para el microcontrolador. Se requería obtener un perfil de velocidad trapezoidal por lo que el objetivo fue obtener a la salida un trapecio. (Figura 3.20)

Velocidad del punto B en rad/seg

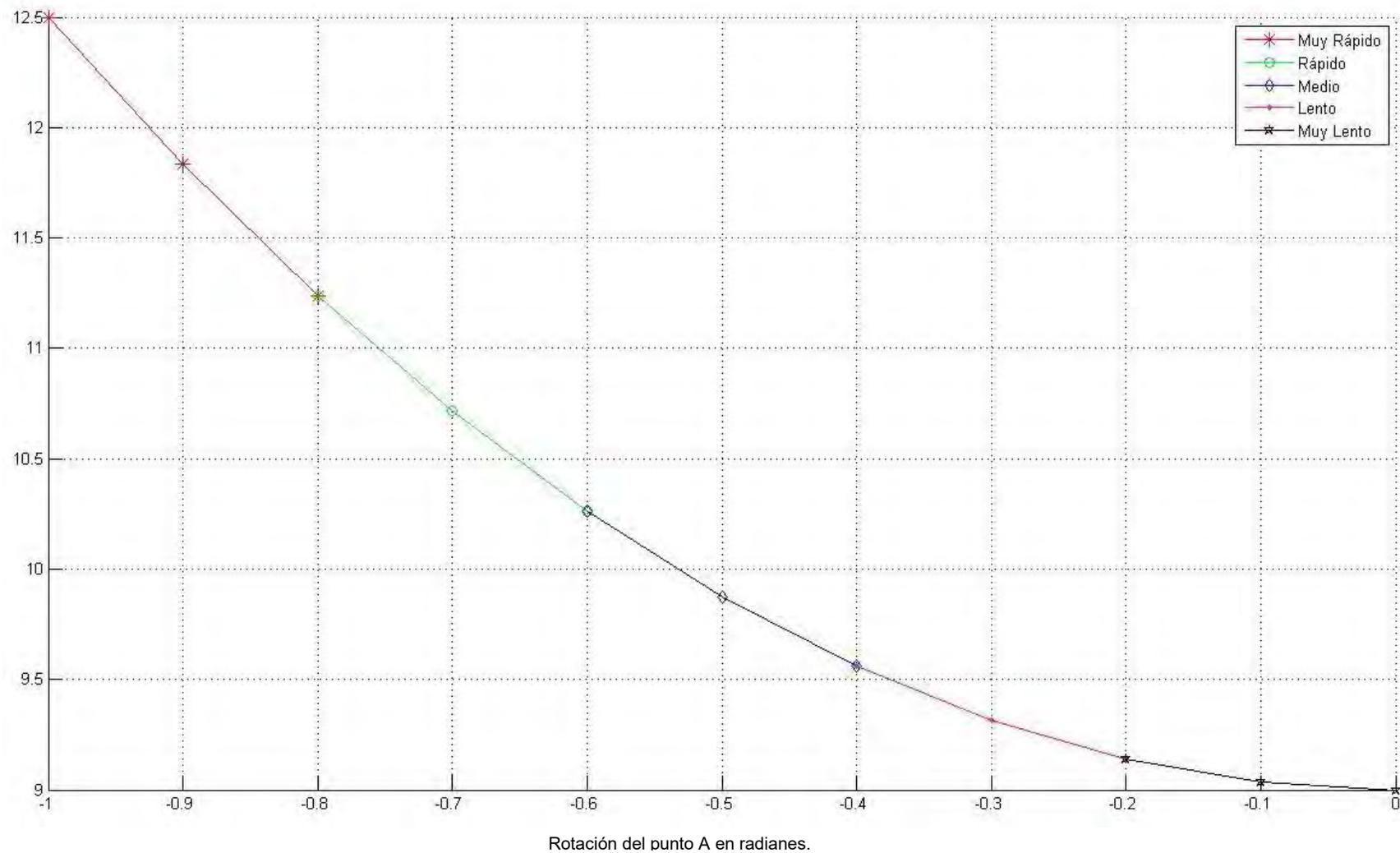


Figura 3.18 Gráfica de los rangos tomados para el diseño de las funciones de membresía de entrada.

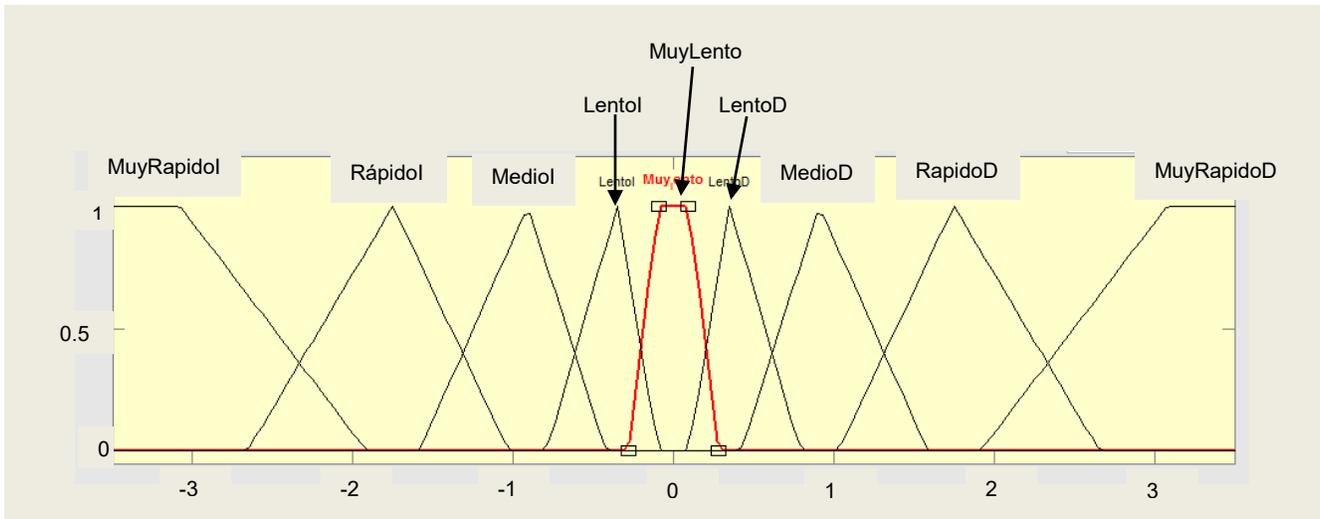


Figura 3.19 Funciones de membresía de entrada para el control proporcional

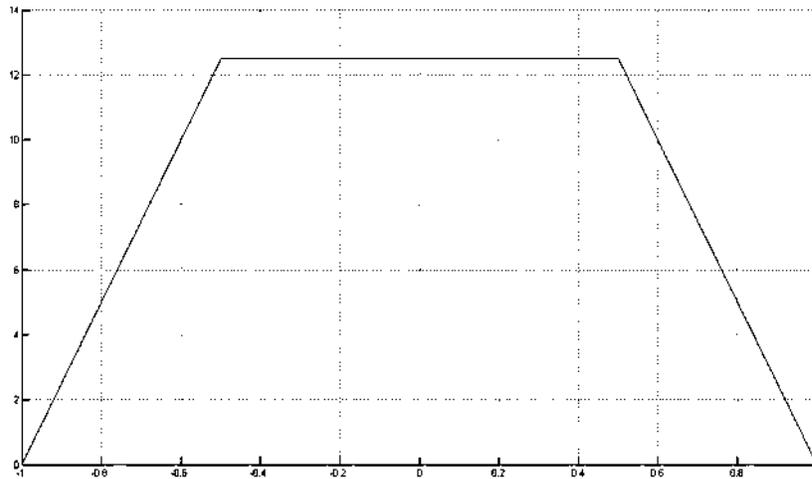


Figura 3.20 Perfil de velocidad deseado para los servomotores de las piernas.

Se requiere diseñar la salida del control de acuerdo a los parámetros de entrada con rango de velocidad de 0 a 12.5 red/s y no con respecto al error.

Para diseñar las funciones de membresía se dividió el perfil en cinco intervalos:

- a)  $0 < \text{Muy Lento} < 1.5$
- b)  $1.5 < \text{Lento} < 4$
- c)  $4 < \text{Medio} < 7$
- d)  $7 < \text{Rápido} < 10.5$
- e)  $10.5 < \text{MuyRápido} < 12.5$

Una vez definidos todos los parámetros y rangos necesarios, se introdujo la información en el software de Matlab para diseño de control de lógica difusa. Siguiendo los estándares de diseño de control de lógica difusa para lograr salidas amortiguadas se eligió utilizar funciones de membresía de forma triangular. También se ajustaron las funciones para que quedaran bien distribuidas en el rango. Figura 3.21

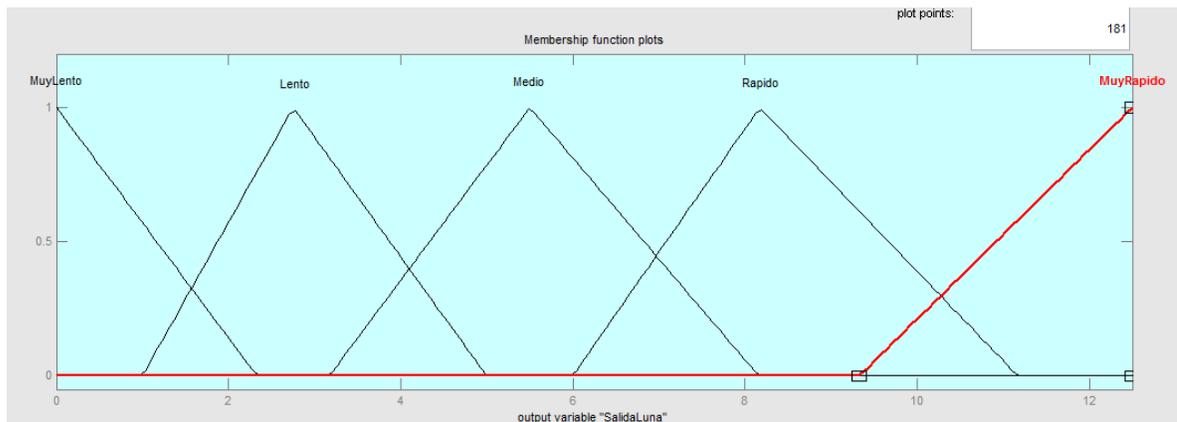


Figura 3.21 Primer propuesta de funciones de membresía de salida para el control proporcional.

### 3.6.3. Diseño de las reglas del control difuso

Las reglas de control difuso se crean utilizando la lógica natural. Partiendo de las características de entrada conocidas se infiere qué debe de suceder con el control, es decir, si se sabe que en la parte inicial del control el mecanismo Peaucellier-Lipkin alcanza su máxima velocidad, es lógico pensar que justo ahí es cuando se debe de tener la menor velocidad en nuestro motor. Conforme el mecanismo se acerca a la parte media se observa que la velocidad del punto B (Figura 3.16) va disminuyendo, por lo que es necesario que el motor se mueva a la mayor velocidad para compensarlo. Las reglas se ponen en una tabla para cuidar el orden y deducir el funcionamiento a la salida del control. Como en este caso

sólo se tiene una sola función de membresía de entrada, la tabla queda de forma directa. (Tabla 3.14)

**Tabla 3.14** Tabla de las reglas para el control.

Entrada	Salida
MuyRápido I	MuyLento
Rapido I	Lento
Medio I	Medio
Lento I	Rapido
MuyLento	MuyRapido
Lento D	Rapido
Medio D	Medio
Rapido D	Lento
Muy Rápido D	MuyLento

Ya teniendo completa la tabla ingresamos los datos en el módulo de lógica difusa de Matlab. (Figura 3.22)

```

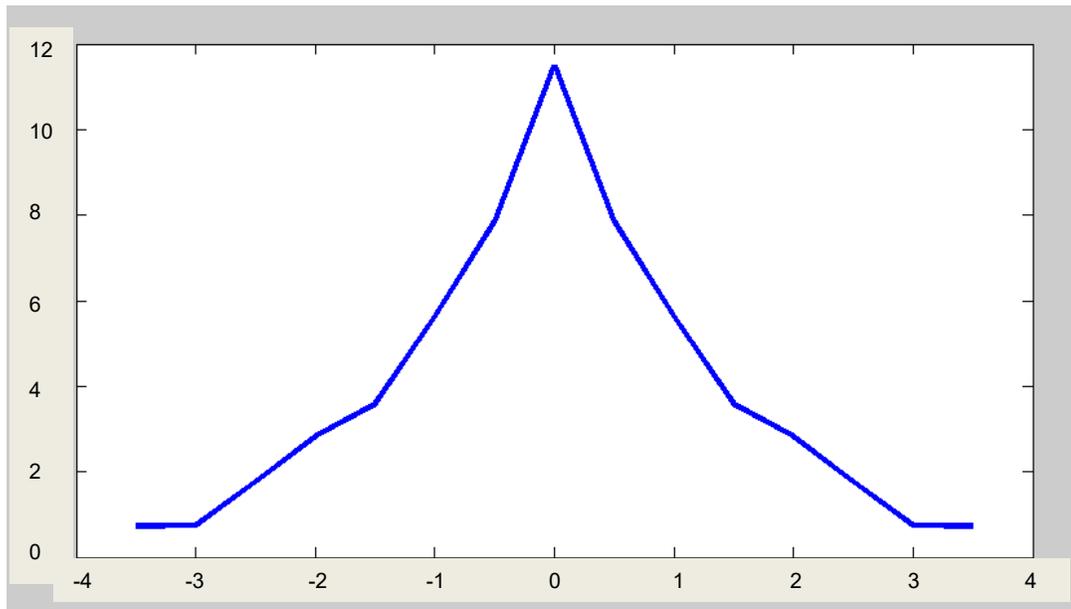
1. If (EntradaLuna is MuyRapidoD) then (SalidaLuna is MuyLento) (1)
2. If (EntradaLuna is MuyRapidoI) then (SalidaLuna is MuyLento) (1)
3. If (EntradaLuna is RapidoD) then (SalidaLuna is Lento) (1)
4. If (EntradaLuna is RapidoI) then (SalidaLuna is Lento) (1)
5. If (EntradaLuna is MedioI) then (SalidaLuna is Medio) (1)
6. If (EntradaLuna is MedioD) then (SalidaLuna is Medio) (1)
7. If (EntradaLuna is LentoI) then (SalidaLuna is Rapido) (1)
8. If (EntradaLuna is LentoD) then (SalidaLuna is Rapido) (1)
9. If (EntradaLuna is Muy_lento) then (SalidaLuna is MuyRapido) (1)

```

**Figura 3.22** Reglas del control P para el LUNAbot W6 VJ1.0

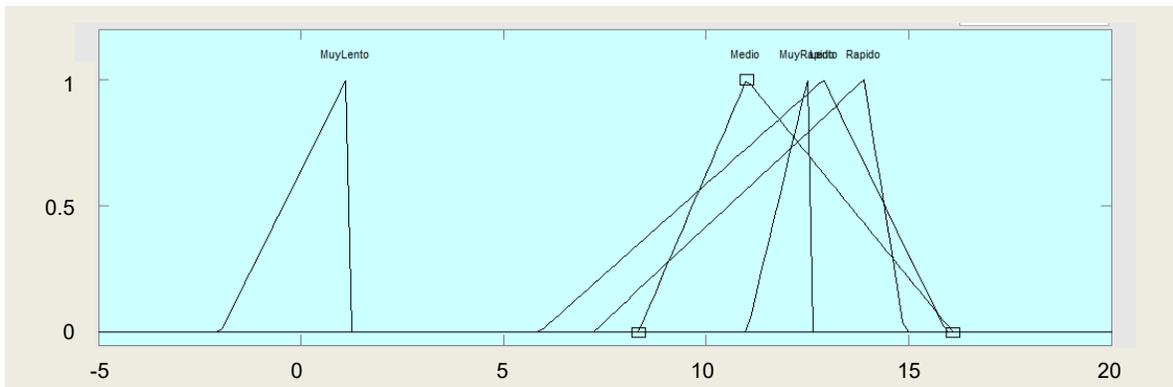
Teniendo ya las reglas se observa la salida que tendrá el control con respecto a la entrada y así se determina si el perfil de velocidad será el deseado. El perfil de velocidad obtenido con las especificaciones dadas tiene una forma cónica (Figura 3.23), por lo tanto no se define como satisfactorio, sin embargo se considera el

hecho de que ya se alcanza la velocidad máxima en el punto medio del trayecto, por lo que las modificaciones siguientes se hacen en base a los resultados de este primer intento.



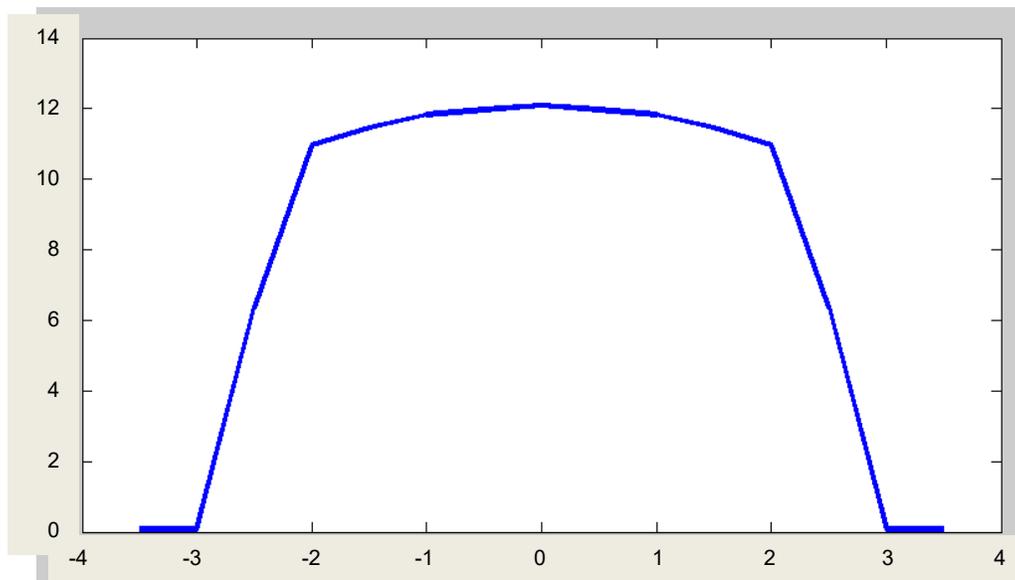
**Figura 3.23 Perfil de velocidad obtenido después de la primera prueba**

A simple vista se puede observar que el perfil está compuesto por secciones de recta. Cada una de las secciones de recta corresponde a la acción que aplica cada una de las funciones de membresía de salida, por lo tanto se realizó un ajuste a cada una de las funciones de membresía (Figura 3.24) para modificar el perfil de salida.



**Figura 3.24 Funciones de membresía de salida modificadas**

Al realizar estas modificaciones el perfil de velocidad obtenido es mucho más cercano a lo deseado. (Figura 3.25)



**Figura 3.25 Perfil de velocidad de salida obtenido con las nuevas funciones de membresía de salida**

Se puede observar un perfil que se acerca más al deseado. Se tiene un aumento gradual de velocidad hasta alcanzar un punto máximo, después esta velocidad se mantiene durante todo el trayecto y vuelve a disminuir gradualmente hasta llegar a cero. Este perfil se considera como satisfactorio ya que cumple con las expectativas, por lo que se definen las nuevas funciones de membresía, quedando los rangos de la siguiente forma:

- Muy Lento:
  - Inicio: -1.991
  - Max: 1.12
  - Fin: 1.18
  
- Lento :
  - Inicio: 5.88
  - Max: 12.9
  - Fin: 15.93
  
- Medio:
  - Inicio: 8.327
  - Max: 11
  - Fin: 16.1
  
- Rápido:
  - Inicio: 7.202
  - Max: 13.9
  - Fin: 14.9
  
- Muy Rápido:
  - Inicio: 11
  - Max: 12.5
  - Fin: 12.5

Una vez diseñado el control se procede a realizar una simulación para observar la señal de salida que se obtiene del control. La simulación se realiza en el programa de Matlab simulink. Se comenzó por crear el control por medio de programación por bloques. (Figura 3.26)

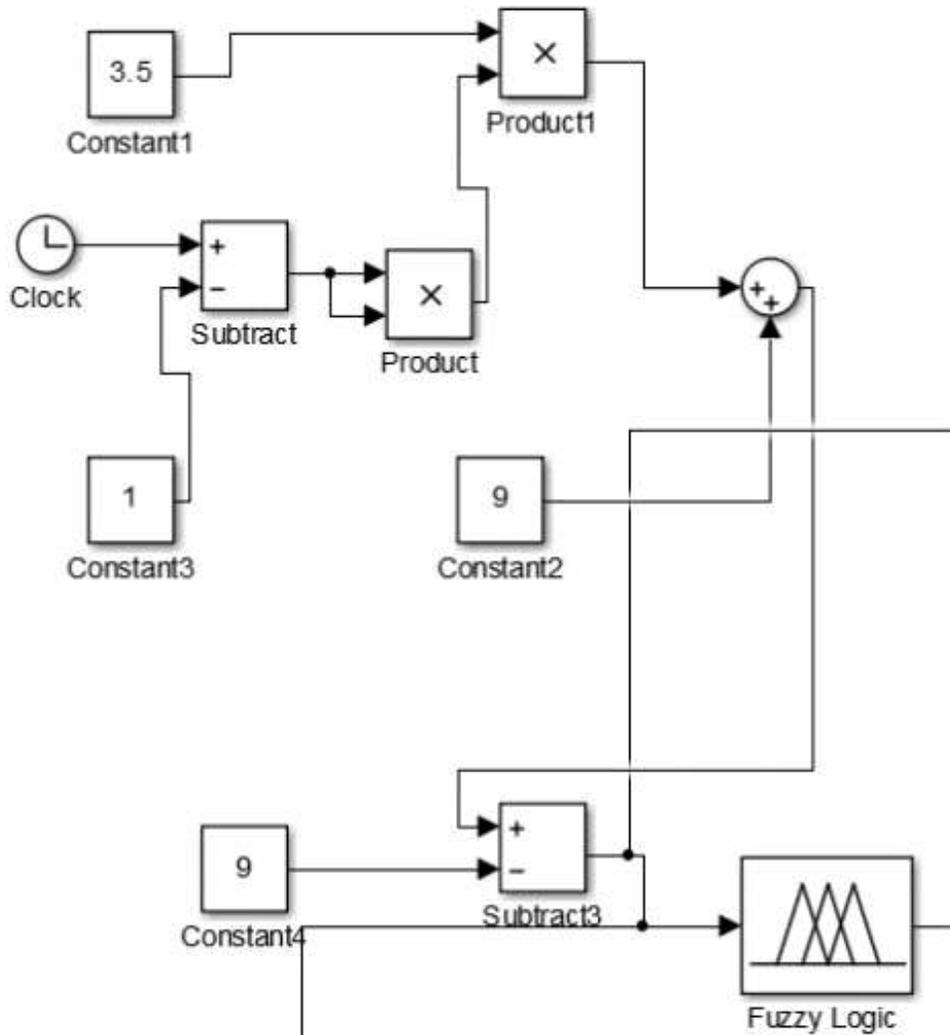


Figura 3.26 Programa de bloques para la simulación del control con lógica difusa del LUNAbot W6 VJ1.0

En la imagen se puede observar el bloque del control definido como “*Fuzzy Logic*”, al cual entra la señal de salida del bloque “*Subtract3*”. Este bloque es el que representa el cálculo del error con el que trabajará el control difuso. En este bloque se puede ver que a una señal de llegada se le resta una constante que es nueve (nuestra referencia). Todos los bloques restantes son sólo bloques necesarios para representar el perfil de entrada parabólica que representa al perfil de velocidad del mecanismo Peaucellier-Lipkin, antes de pasar por el control.

Se observa cómo el resultado de la simulación (Figura 3.27) se asemeja bastante al perfil de velocidad que deseamos obtener como salida de nuestro control, sin embargo, las perturbaciones que se encuentran en la fase estable del control son significativas, por lo que es determinante agregar una etapa de control derivativo al control que tenemos actualmente que es solamente proporcional para reducir las oscilaciones que existen en la parte estable del control.

Se quiere que el control sea estable en 12.5 por lo tanto se puede trabajar con la diferencia que se tenga con esa referencia. Se diseña un control Proporcional Derivativo para solucionar el ruido del control y se proponen las siguientes funciones de membresía:

Funciones de entrada para la parte proporcional:

- Saturación Baja:
  - Inicio: -13
  - Máximo: -0.2
  - Final: -0.16
- Abajo:
  - Inicio: -0.36
  - Máximo: -0.2
  - Final -0.04
- Bueno:
  - Inicio: -0.1589
  - Máximo: 0.001058
  - Final: 0.1611
- Arriba:
  - Inicio: 0.04
  - Máximo: 0.2
  - Final: 0.36
- Saturación Alta:
  - Inicio: -13
  - Máximo: -0.2
  - Final: -0.16

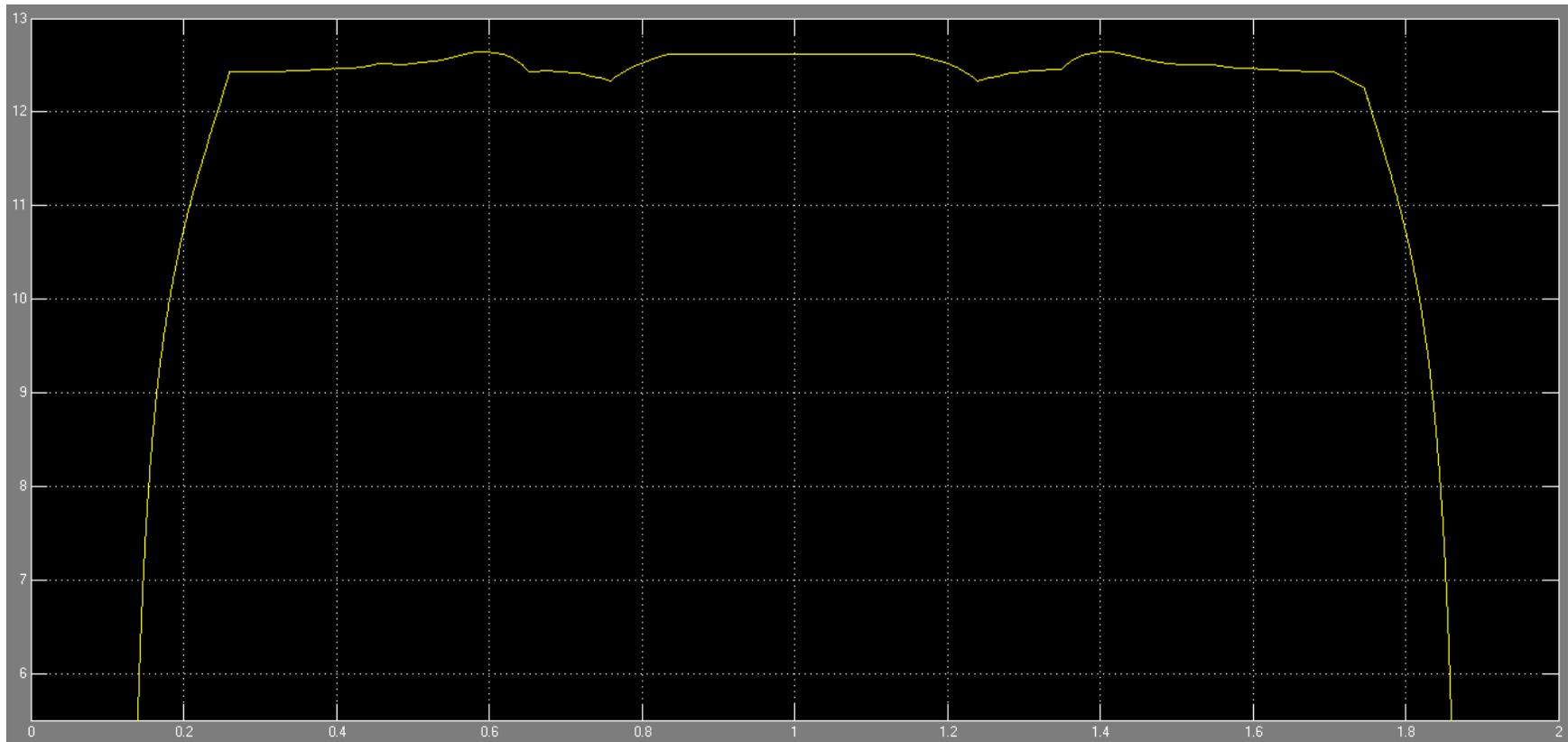


Figura 3.27 Perfil de velocidad obtenido después de la simulación utilizando el control Proporcional medida en rad/s.

Funciones de entrada para la parte derivativa:

- Bajo:
  - Inicio: -1.789
  - Máximo: -0.9895
  - Final -0.1894
- Medio:
  - Inicio: -0.8
  - Máximo: 0
  - Final: 0.8
- Alto:
  - Inicio: 0.2
  - Máximo: 1
  - Final: 1.8

Funciones de membresía de salida:

- Bajar:
  - Inicio: -0.54
  - Máximo: -0.3
  - Final -0.06
- Mantener:
  - Inicio: -0.24
  - Máximo: 0
  - Final: 0.24
- Subir:
  - Inicio: 0.06
  - Máximo: 0.3
  - Final: 0.54

Una vez teniendo las funciones de membresía se redactan las reglas del control haciendo una tabla de reglas. A diferencia del control proporcional, este

control tiene dos entradas y una salida por lo que nuestra matriz de reglas queda ahora como una matriz bidimensional. (Tabla 3.15)

Tabla 3.15 Matriz de reglas para el control difuso PD.

	Saturación Baja	Abajo	Bueno	Arriba	Saturación Alta
Bajo	Mantener	Subir	Mantener	Bajar	Mantener
Medio	Mantener	Subir	Mantener	Bajar	Mantener
Alto	Mantener	Mantener	Mantener	Mantener	Mantener

Una vez creada la matriz de reglas es posible introducirlas al módulo Fuzzy de Matlab. (Figura 3.28)

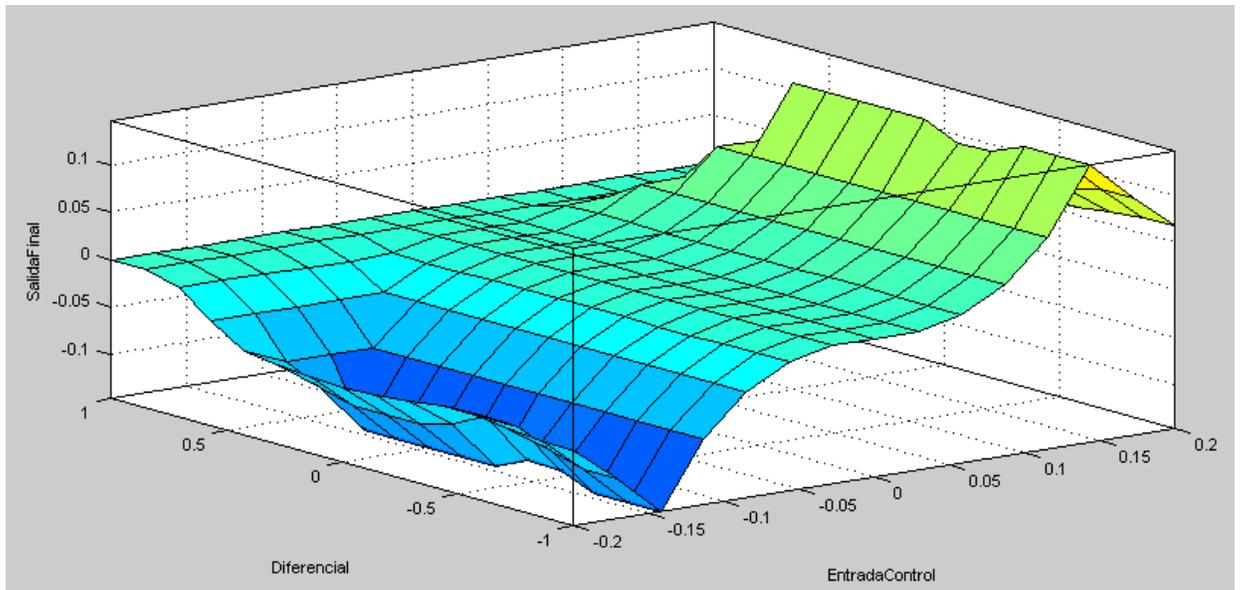
```

1. If (EntradaControl is SatBaja) and (Diferencial is Bajo) then (SalidaFinal is Mantener) (1)
2. If (EntradaControl is SatBaja) and (Diferencial is Medio) then (SalidaFinal is Mantener) (1)
3. If (EntradaControl is SatBaja) and (Diferencial is Alto) then (SalidaFinal is Mantener) (1)
4. If (EntradaControl is SatAlta) and (Diferencial is Bajo) then (SalidaFinal is Mantener) (1)
5. If (EntradaControl is SatAlta) and (Diferencial is Medio) then (SalidaFinal is Mantener) (1)
6. If (EntradaControl is SatAlta) and (Diferencial is Alto) then (SalidaFinal is Mantener) (1)
7. If (EntradaControl is Bueno) and (Diferencial is Medio) then (SalidaFinal is Mantener) (1)
8. If (EntradaControl is Bueno) and (Diferencial is Bajo) then (SalidaFinal is Mantener) (1)
9. If (EntradaControl is Bueno) and (Diferencial is Alto) then (SalidaFinal is Mantener) (1)
10. If (EntradaControl is Abajo) and (Diferencial is Medio) then (SalidaFinal is Subir) (1)
11. If (EntradaControl is Abajo) and (Diferencial is Alto) then (SalidaFinal is Mantener) (1)
12. If (EntradaControl is Abajo) and (Diferencial is Bajo) then (SalidaFinal is Subir) (1)
13. If (EntradaControl is Arriba) and (Diferencial is Medio) then (SalidaFinal is Bajar) (1)
14. If (EntradaControl is Arriba) and (Diferencial is Alto) then (SalidaFinal is Mantener) (1)
15. If (EntradaControl is Arriba) and (Diferencial is Bajo) then (SalidaFinal is Bajar) (1)

```

Figura 3.28 Reglas del control difuso PD en el módulo Fuzzy de Matlab

Una vez escritas las reglas se observa la salida del bloque de control que eliminará las perturbaciones del control proporcional. Se debe considerar que debido a que se tienen dos funciones de membresía de entrada, se crea una superficie tridimensional. (Figura 3.29)



**Figura 3.29 Superficie de la salida del control PD**

La salida de este control se añadirá a la salida del control proporcional, formando finalmente un control P+PD (Proporcional + Proporcional Derivativo). Una vez diseñado el control PD, ampliamos el programa de bloques para realizar la simulación con este control. (Figura 3.30)

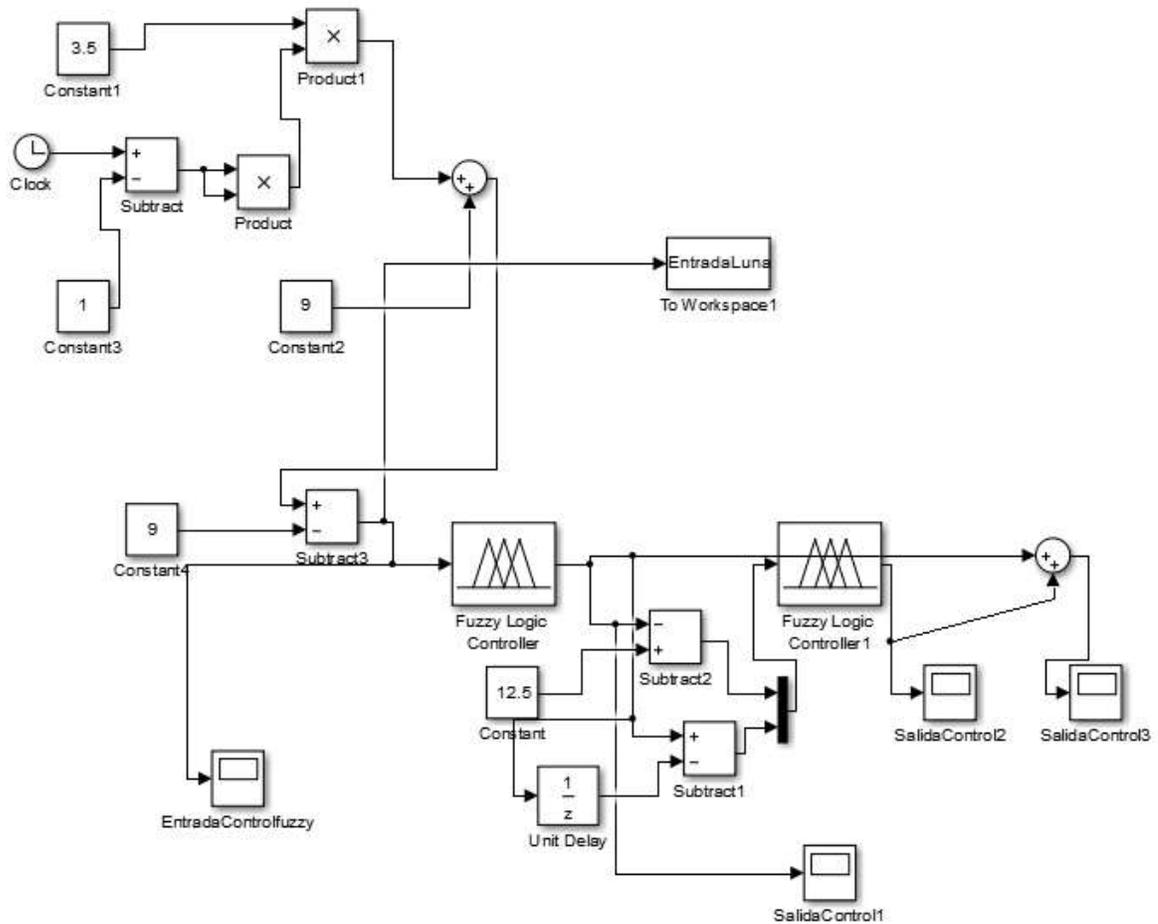
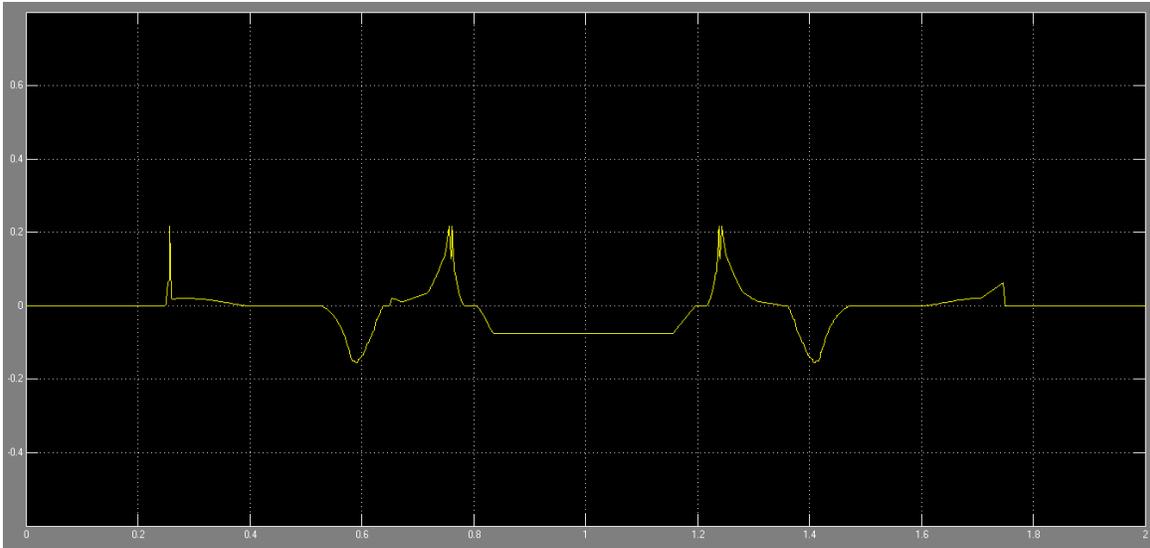


Figura 3.30 Programación en bloques del control P+PD utilizando Simulink.

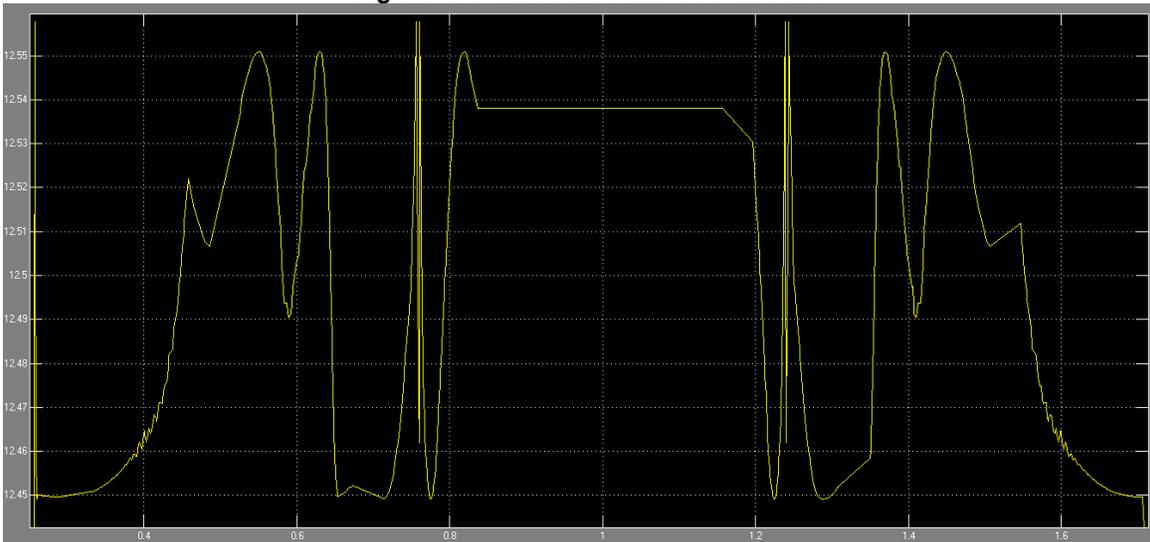
Se observa la adición de un segundo bloque de control llamado “Fuzzy Logic 1”, el cual tiene dos entradas, la primera es la diferencia que existe entre el nivel actual de nuestra salida y el nivel estable que deseamos que es de 12.5. La segunda entrada es la diferencia que existe entre el nivel actual de nuestro control y el nivel en que nos encontrábamos anteriormente.

Realizando la simulación se observa que la salida de este control da los valores que se agregarán al control proporcional para corregir las perturbaciones de la fase estable (Figura 3.31 y Figura 3.32). Esta salida se sumará a la salida del primer bloque de control obteniendo una salida con menor porcentaje de perturbaciones en la fase estable (Figura 3.33). Al observar la gráfica de la salida

final, se ve que las oscilaciones son bastante tenues y que el resultado ha mejorado significativamente. Se muestra que la variación de la señal con respecto a la referencia deseada de 12.5 es de +0.05 y -0.052 rad/s. Esto equivale a un 0.416% del valor de la señal deseada por lo tanto de acuerdo a la teoría de diseño de control [54], que marca una variación del 5% como tolerable, se considera una señal estable.



**Figura 3.31 Salida del control PD en rad/s**



**Figura 3.32 Acercamiento a las perturbaciones de la fase estable del control.**

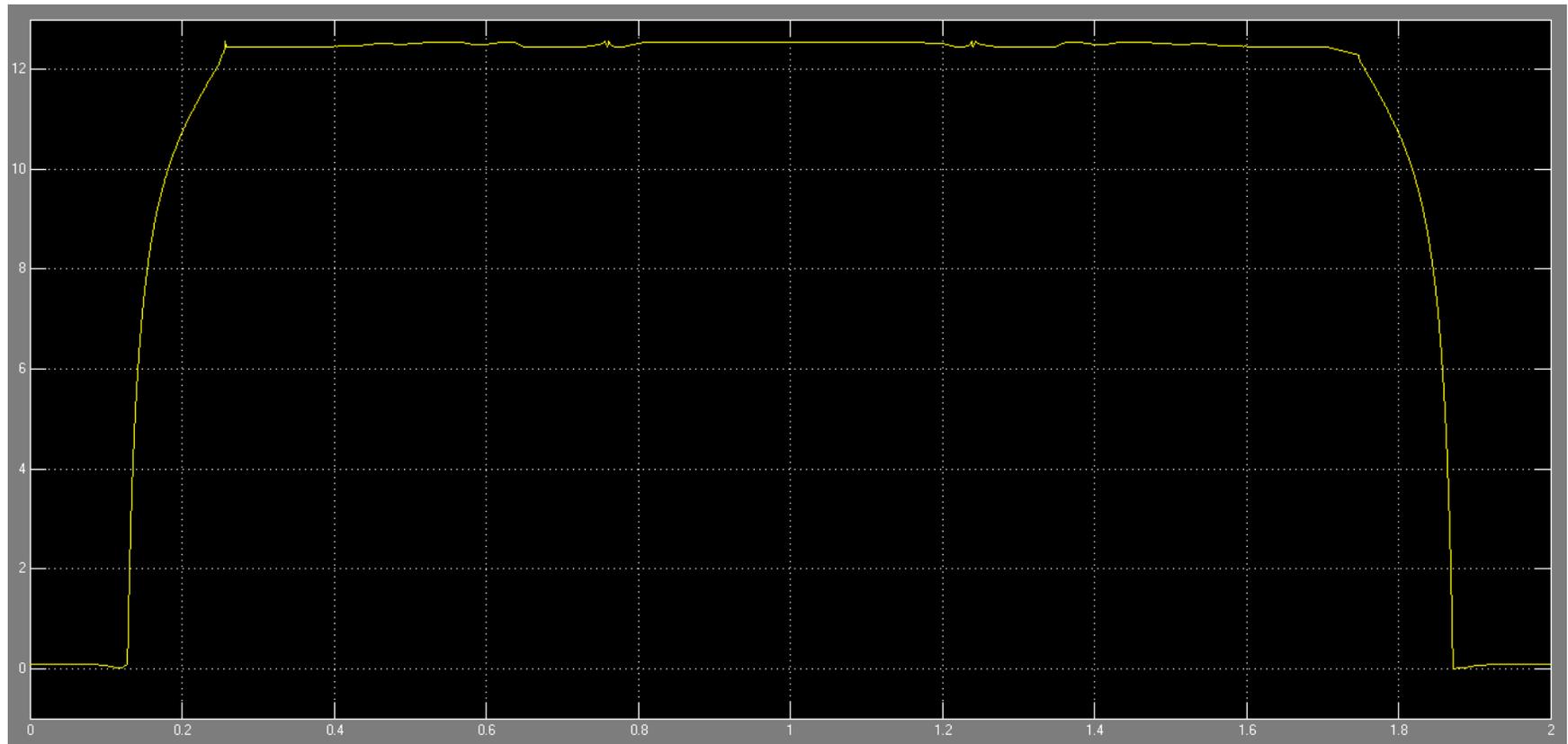


Figura 3.33 Salida del control P+PD en rad/s.

### 3.7. Codificación del control difuso

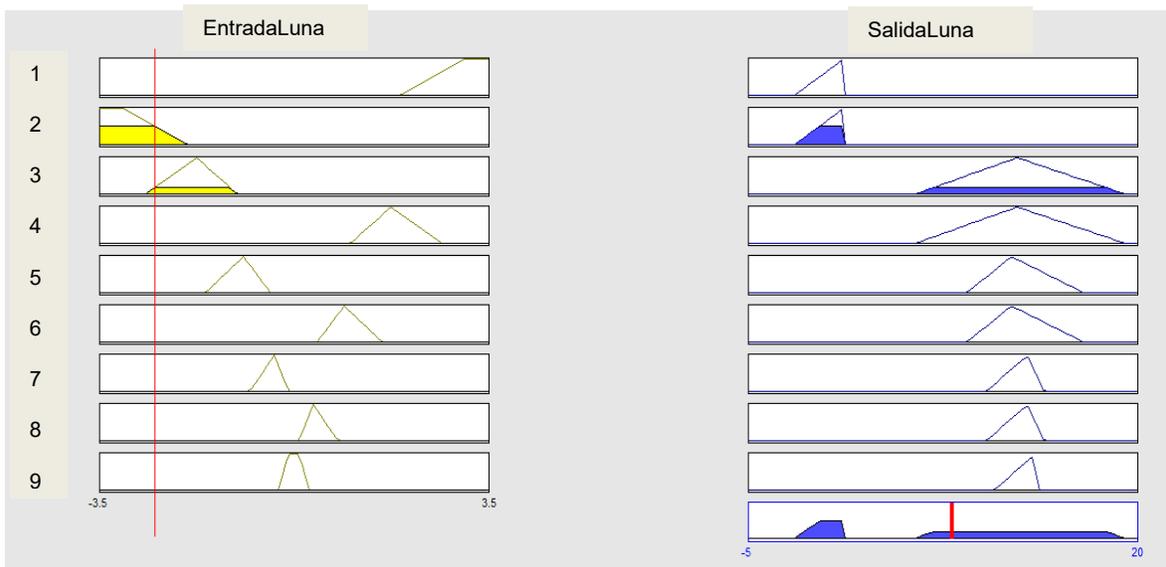
Una vez finalizado el diseño del control difuso se procede a la codificación del control para poder utilizarlo en un microcontrolador. Utilizando el diagrama de flujo previamente elaborado (Diagrama 3.4) y el diseño del control en Matlab se puede realizar la codificación adecuada.

Todos los microcontroladores necesitan ser programados por medio de un archivo hexadecimal, en donde se encuentran todas las instrucciones. Este archivo puede crearse utilizando un programa que permita la traducción de un lenguaje de programación a un lenguaje hexadecimal. Uno de estos programas es el arduino. El arduino aparte de proporcionar esta herramienta, brinda un lenguaje de programación sencillo y compacto (Procesind/Wiring), es decir, que se requieren menos líneas de programación para realizar las acciones deseadas que otro lenguaje como por ejemplo el c++ o el lenguaje ensamblador.

Para realizar la programación del control difuso en arduino fue necesario establecer exactamente como se realiza la fuzzificación y defuzzificación, ya que se debía indicar al microcontrolador cómo realizarlas.

La fuzzificación consiste en evaluar el valor de entrada al control en cada una de nuestras funciones de membresía de entrada, y en cada una de ellas asignarle un valor, que corresponda al peso que tiene éste en las funciones. Cuando el valor de entrada se encuentra fuera de rango de una función de membresía, su peso o valor en esta función será de cero, mientras que será de uno cuando el valor de entrada quede justo en la punta de la función de membresía. En caso de encontrarse entre estos dos valores, el peso o valor de la entrada debe asignarse utilizando la función de recta que describe a esa parte de la función de membresía. Cuando existen dos funciones de membresía de entrada o más, se aplica la regla del máximo, en donde se toma el valor máximo de las dos funciones de membresía de entrada y, con ese valor, se evalúa en la función de membresía de salida que corresponda según la regla.

Una vez realizado este proceso se procede a la defuzzificación, que consiste análogamente en calcular el centroide en el eje x del área formada por las funciones de membresía de salida. Digitalmente, este proceso puede realizarse utilizando los puntos dados como resultado en las funciones de membresía de salida y calculando el área de los rectángulos que forman desde ese punto hasta la esquina inicial de la función. Si bien el resultado que se obtiene no es exacto, la aproximación es bastante buena ya que la diferencia que existe del cálculo entre estos dos métodos se encuentra en el orden de los centésimos y para este control, dicha diferencia no es significativa, por el contrario, al calcular el centroide de esta forma se reduce la cantidad de líneas de instrucción necesarias para programar el control, lo que reduce a su vez el tiempo de operación del control difuso.



**Figura 3.34** Ejemplo del proceso de fuzzificación y defuzzificación para una entrada de -2.5 en el control P. En amarillo podemos observar las funciones de membresía de entrada. La línea roja que la cruza marca el valor de entrada. De azul observamos las funciones de membresía de salida. En la parte inferior podemos ver en azul el polígono formado por la suma de áreas de las funciones de membresía de salida. Y en color rojo una línea que marca el centroide en el eje X del área.

La codificación se muestra a continuación.

```
//INCLUYENDO LIBRERIAS
#include <Servo.h>

//DECLARACION DE VARIABLES

//VARIABLES PARA LOS SERVOS
Servo G1;
Servo G2;
Servo MD1;
Servo MD2;

//VARIABLES PARA LOS MOTORES DE CD
int G4S=4;
int G4B=5;
int G5S=6;
int G5B=7;

//VARIABLES PARA EL CONTROL DEL USUARIO
int arriba=8;
int abajo=9;
int derecha=10;
int izquierda=11;

//VARIABLES PARA EL CONTROL DIFUSO
//VARIABLE DE ENTRADA
int posmotor;
float error;
float diferencial;

/*-----
      FUNCIONES DE TRANSFERENCIA DE ENTRADA Y SALIDA DEL CONTROL
PROPORCIONAL
      -----

ARREGLO PARA LAS FUNCIONES DE MEMBRESIA DE ENTRADA Y SALIDA

ENTRADA
MuyLento Trapezoidal -0.28 -0.0933 0.0933 0.28
Lento Triangular 0.0933 0.35 0.35 0.7933
Medio Triangular 0.42 0.91 0.91 1.587
Rapido Triangular 1.027 1.75 1.75 2.66
MuyRapido Trapezoidal 1.913 3.08 3.4 3.5

SALIDA
MuyLento Triangular -1.982 1.117 1.117 1.172
Lento Triangular 5.795 12.29 12.29 19.19
Medio Triangular 9.115 11.91 11.91 16.5
Rapido Triangular 10.34 12.94 12.94 14.04
MuyRapido Triangular 10.85 13.45 13.45 13.55
*/
```

```

float EntfunP[]={-0.28,-0.0933, 0.0933, 0.28,0.0933,0.35,0.35,0.7933, 0.42,0.91,0.91,1.587,
1.027, 1.75,1.75,2.66,1.913,3.08,3.4,3.5};
float SalfunP[]={-1.982,1.117,1.117,1.172,5.795,12.29,12.29,19.19, 9.115,11.91,11.91,16.5,
10.34,12.94,12.94,14.04,10.85,13.45,13.45,13.55};

```

```

//ARREGLO PARA LA EVALUACION DE FUNCIONES
float ResFunEntP[5];
float ResFunSalP[5];

```

```

/*-----
FUNCIONES DE TRANSFERENCIA DE ENTRADA Y SALIDA DEL CONTROL
PORPORCIONAL DERIVATIVO
-----

```

ARREGLO PARA LAS FUNCIONES DE MEMBRESIA DE ENTRADA Y SALIDA

ENTRADA

EntradaControl

SatBaja	Trapezoidal	-13	-13	-0.2	-0.16
Arriba	Triangular	-0.036	-0.2	-0.2	-0.04
Bueno	Triangular	-0.16	0	0	0.16
Abajo	Triangular	0.04	0.2	0.2	0.36
SatAlta	Trapezoidal	0.16	0.2	13	13

Diferencial

Bajo	Triangular	-1.789	-0.9895	-0.9895	-0.1894
Medio	Triangular	-0.8	0	0	0.8
Alto	Triangular	0.2	1	1	1.8

SALIDA

SalidaFinal

Bajar	Triangular	-0.54	-0.3	-0.3	-0.06
Mantener	Triangular	-0.24	0	0	0.24
Subir	Triangular	0.06	0.3	0.3	0.54

\*/

```

float Entfun1PD[]={-13,-13,-0.2,-0.16,-0.036,-0.2,-0.2,-0.04,-0.16,0,0,0.16, 0.04,0.2, 0.2,0.36,
0.16, 0.2,13,13};
float Entfun2PD[]={-1.789,-0.9895,-0.9895,-0.1894,-0.8,0,0,0.8,0.2,1,1,1.8};
float SalfunPD[]={-1.789,-0.9598,-0.9895,-0.1894,-0.8,0,0,0.8,0.2,1,1,1.8};

```

```

//ARREGLO PARA LA EVALUACION DE FUNCIONES

```

```

float ResFunEnt1PD[5];
float ResFunEnt2PD[3];
float ResFunSalPD[15];

```

```

float Minimo[15];
float Maximo[3];

```

```

int salidaP;
int salidaPD;
int marca[]={0,0,0};

```

```
/*-----  
                               INICIO DEL PROGRAMA  
-----*/
```

```
void setup(){  
  G1.attach(0);  
  G2.attach(1);  
  MD1.attach(2);  
  MD2.attach(3);  
  pinMode(G4S,OUTPUT);  
  pinMode(G4B,OUTPUT);  
  pinMode(G5S,OUTPUT);  
  pinMode(G5B,OUTPUT);  
  pinMode(arriba,INPUT);  
  pinMode(abajo,INPUT);  
  pinMode(derecha,INPUT);  
  pinMode(izquierda,INPUT);  
}
```

```
void loop(){  
  
  posmotor=90;  
  G1.write(90);  
  G2.write(90);  
  
  if (arriba=1){  
    while(arriba=1){  
      adelante();  
    }  
  }  
  if (abajo=1){  
    while (abajo=1){  
      atras();  
    }  
  }  
  if (izquierda=1){  
    preparagiro();  
    while (izquierda=1){  
      adelante();  
    }  
    terminagiro();  
  }  
  if (derecha=1){  
    preparagiro();  
    while (derecha=1){  
      atras();  
    }  
    terminagiro();  
  }  
}
```

```
void adelante(void){  
  G4B=1;  
  G5S=1;  
  
  //G1=30; Hacer que el motor llegue a 30 con el control
```

```

posmotor=120;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G1.write(salidaPD);

//G2=-30; Hacer que el motor llegue a -30 con el control
posmotor=60;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G2.write(salidaPD);

G5B=1;
G4S=1;
//G2=-30; Hacer que el motor llegue a -30 con el control
posmotor=60;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G2.write(salidaPD);

//G1=30; Hacer que el motor llegue a 30 con el control
posmotor=120;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G1.write(salidaPD);
}

void atras(void){
G4B=1;

```

```

G5S=1;
//G1=-30; Hacer que el motor llegue a -30 con el control
posmotor=60;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G1.write(salidaPD);

//G2=30; Hacer que el motor llegue a 30 con el control
posmotor=120;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G2.write(salidaPD);

G5B=1;
G4S=1;

//G2=30; Hacer que el motor llegue a 30 con el control
posmotor=120;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G2.write(salidaPD);

//G1=-30; Hacer que el motor llegue a -30 con el control
posmotor=60;
EvalEntP();
EvalRegP();
DefusP();
EvalEnt1PD();
EvalEnt2PD();
MinPD();
EvalRegPD();
CalcMax();
DefusPD();
G1.write(salidaPD);
}

```

```

void preparagiros(void){
    G5S=1;
    MD2.write(120);
    G5B=1;
    G4S=1;
    MD1.write(120);
}

void terminagiros(void){
    MD2.write(0);
    G5B=1;
    G4S=1;
    MD1.write(0);
}

/*-----
                                CONTROL PROPORCIONAL
-----*/
//FUNCION PARA EVALUAR EL VALOR DE ENTRADA EN CADA UNA DE LAS
FUNCIONES
int EvalEntP(void){
    float posicion;
    int i=0;
    int a=0;
    posicion=(posmotor-64.44)/15.873;
    for (i=0;i<19;i=i+4){
        if(posicion<=EntfunP[i] || posicion>=EntfunP[i+3]){
            ResFunEntP[a]=0;
        }
        else if (posicion>=EntfunP[i+1] && posicion<=EntfunP[i+2]){
            ResFunEntP[a]=1;
        }
        else if (posicion>EntfunP[i] && posicion<EntfunP[i+1]){
            ResFunEntP[a]=(posicion-EntfunP[i])*(1/(EntfunP[i+1]-EntfunP[i]));
        }
        else{
            ResFunEntP[a]=(EntfunP[i+3]-posicion)*(1/(EntfunP[i+3]-EntfunP[i+2]));
        }
        a=a+1;
    }
}

//FUNCION PARA LAS REGLAS DEL CONTROL PROPORCIONAL
float EvalRegP(void){
    int i=19;
    int a=0;
    for (i=15;i>0;i=i-4){
        if(ResFunEntP[a]<=SalfunP[i] || ResFunEntP[a]>=SalfunP[i+3]){
            ResFunSalP[a]=0;
        }
        else if (ResFunEntP[a]>=SalfunP[i+1] && ResFunEntP[a]<=SalfunP[i+2]){
            ResFunSalP[a]=1;
        }
        else if (ResFunEntP[a]>SalfunP[i] && ResFunEntP[a]<SalfunP[i+1]){
            ResFunSalP[a]=(ResFunEntP[a])/(1/(SalfunP[i+1]-SalfunP[i]));
        }
    }
}

```

```

    }
    else{
        ResFunSalP[a]=(0-ResFunEntP[a])/(1/(SalfunP[i+3]-SalfunP[i+2]));
    }
    a=a+1;
}
}

//DEFUSIFICACION PARA EL CONTROL PROPORCIONAL
float DefusP(void){
    float areacent=0;
    float area=0;
    int a=0;
    int i=0;
    for (i=0;i<5;i=i+1){
        areacent=areacent+((ResFunEntP[i]*(ResFunSalP[i]-SalfunP[a]))*((ResFunEntP[i]*
(ResFunSalP[i]-SalfunP[a]))/2));
        area=area+(ResFunSalP[i]*(ResFunEntP[i]-SalfunP[a]));
        a=a+4;
    }
    salidaP=areacent/area;
    salidaP=(15.873*salidaP)+64.44;
}

/*-----
CONTROL PROPORCIONAL DERIVATIVO
-----*/

//FUNCION PARA EVALUAR EL VALOR DE ENTRADA EN CADA UNA DE LAS
FUNCIONES
float EvalEnt1PD(void){
    int i=0;
    int a=0;
    error=posmotor-salidaP;
    error=4.8*error;
    for (i=0;i<20;i=i+4){
        if(error<=Entfun1PD[i] || error>=Entfun1PD[i+3]){
            ResFunEnt1PD[a]=0;
        }
        else if (error>=Entfun1PD[i+1] && error<=Entfun1PD[i+2]){
            ResFunEnt1PD[a]=1;
        }
        else if (error>Entfun1PD[i] && error<Entfun1PD[i+1]){
            ResFunEnt1PD[a]=(error-Entfun1PD[i])*(1/(Entfun1PD[i+1]-Entfun1PD[i]));
        }
        else{
            ResFunEnt1PD[a]=(Entfun1PD[i+3]-error)*(1/(Entfun1PD[i+3]-Entfun1PD[i+2]));
        }
        a=a+1;
    }
}

float EvalEnt2PD(void){
    diferencial=0;
    int dif=0;
    dif=dif-diferencial;
    int a=0;

```

```

int i=0;
for (i=0;i<11;i=i+4){
  if(dif<=Entfun2PD[i] || dif>=Entfun2PD[i+3]){
    ResFunEnt2PD[a]=0;
  }
  else if (dif>=Entfun2PD[i+1] && dif<=Entfun2PD[i+2]){
    ResFunEnt2PD[a]=1;
  }
  else if (dif>Entfun2PD[i] && dif<Entfun2PD[i+1]){
    ResFunEnt2PD[a]=(dif-Entfun2PD[i])*(1/(Entfun2PD[i+1]-Entfun2PD[i]));
  }
  else{
    ResFunEnt2PD[a]=(Entfun2PD[i+3]-dif)*(1/(Entfun2PD[i+3]-Entfun2PD[i+2]));
  }
  a=a+1;
}
diferencial=error;
}

```

//REGLAS: SE OBTIENE EL MINIMO VALOR ENTRE LAS DOS FUNCIONES DE ENTRADA

```

float MinPD(void){
  int i=0;
  int j=0;
  int a=0;
  float num1=0;
  float num2=0;
  for (i=0;i<5;i=i+1){
    for (j=0;j<3;j=j+1){
      num1=ResFunEnt1PD[a];
      num2=ResFunEnt2PD[a];
      Minimo[a]=min(num1,num2);
      a=a+1;
    }
  }
}

```

//SE EVALUA CADA MINIMO EN LA FUNCION QUE CORRESPONDE  
//FUNCION PARA LAS REGLAS DEL CONTROL PROPORCIONAL DERIVATIVO

```

float EvalRegPD(void){
  int i=15;
  int a=0;
  int j=4;
  for (i=0;i<9;i=i+1){
    if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
      ResFunSalPD[a]=0;
    }
    else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
      ResFunSalPD[a]=1;
    }
    else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
      ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
    }
    else{
      ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
    }
  }
}

```

```

    }
    a=a+1;
}

i=9;
j=8;
if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
    ResFunSalPD[a]=0;
}
else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
    ResFunSalPD[a]=1;
}
else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
    ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
}
else{
    ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
}
a=a+1;

i=10;
j=4;
if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
    ResFunSalPD[a]=0;
}
else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
    ResFunSalPD[a]=1;
}
else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
    ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
}
else{
    ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
}
a=a+1;

i=11;
j=8;
if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
    ResFunSalPD[a]=0;
}
else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
    ResFunSalPD[a]=1;
}
else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
    ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
}
else{
    ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
}
a=a+1;

i=12;
j=0;
if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
    ResFunSalPD[a]=0;
}

```

```

}
else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
    ResFunSalPD[a]=1;
}
else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
    ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
}
else{
    ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
}
a=a+1;

i=13;
j=4;
if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
    ResFunSalPD[a]=0;
}
else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
    ResFunSalPD[a]=1;
}
else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
    ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
}
else{
    ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
}
a=a+1;

i=14;
j=0;
if(Minimo[i]<=SalfunPD[j] || Minimo[i]>=SalfunP[j+3]){
    ResFunSalPD[a]=0;
}
else if (Minimo[i]>=SalfunPD[j+1] && Minimo[i]<=SalfunPD[j+2]){
    ResFunSalPD[a]=1;
}
else if (Minimo[i]>SalfunP[i] && Minimo[i]<SalfunP[j+1]){
    ResFunSalPD[a]=(Minimo[i])/(1/(SalfunP[j+1]-SalfunP[j]));
}
else{
    ResFunSalP[a]=(0-Minimo[i])/(1/(SalfunP[j+3]-SalfunP[j+2]));
}
a=a+1;
}

//CALCULANDO EL MAXIMO DE CADA FUNCION DE SALIDA
float CalcMax(void){
    int i=0;
    for (i=0;i<8;i=i+1){
        Maximo[1]=max(Maximo[1],ResFunSalPD[i]);
        if (ResFunSalPD[i]>Maximo[1]){
            marca[1]=i;
        }
    }
    Maximo[1]=max(Maximo[1],ResFunSalPD[10]);
    if (ResFunSalPD[10]>Maximo[1]){

```

```

    marca[1]=10;
}
Maximo[1]=max(Maximo[1],ResFunSalPD[13]);
if (ResFunSalPD[10]>Maximo[1]){
    marca[1]=13;
}
Maximo[0]=max(ResFunSalPD[12],ResFunSalPD[14]);
if (ResFunSalPD[12]>ResFunSalPD[14]){
    marca[0]=12;
}
else{
    marca[0]=14;
}
Maximo[2]=max(ResFunSalPD[9],ResFunSalPD[11]);
if (ResFunSalPD[9]>ResFunSalPD[11]){
    marca[0]=9;
}
else{
    marca[0]=11;
}
}

//DEFUZZIFICACION DEL CONTROL PROPORCIONAL DERIVATIVO
float DefusPD(void){
    float areacent=0;
    float area=0;
    int a=0;
    int i=0;
    for (i=0;i<3;i=i+1){
        areacent=areacent+((ResFunEntP[marca[i]]*(Maximo[i]-SalfunPD[a]))*((Maximo[i]-
SalfunPD[a])/2));
        area=area+(ResFunEntP[marca[i]]*(Maximo[i]-SalfunPD[a]));
        a=a+4;
    }
    salidaPD=areacent/area;
}

```

### 3.8. Diseño del circuito electrónico

Una vez codificado el control se realiza el circuito capaz de darle movimiento al robot LUNAbot W6 VJ1.0. Lo primero es realizar la lista de los elementos que integran el control. Los elementos principales para el diseño se muestran a continuación. Sabiendo que el peso del robot será menor a 3kg y que el consumo de corriente de los motores no es mayor a 1 A se eligen los elementos. (Tabla 3.16)

Tabla 3.16 Tabla de elementos para el circuito electrónico.

Parte	Características principales	Cantidad	Utilidad
Motor FF-050SB 	Voltaje de alimentación: 5V  Velocidad: 10400RPM	6	Pata del robot
L298 	Voltaje de alimentación: 2.5 – 46V  I <sub>o</sub> : hasta 2 A	2	Conectado al motor de pata
DS3115 Metal Digital Servo 	Velocidad de Operación: 0.16 sec/60° Torque : 13.5 kg*cm  Voltaje de Operación : 4.8-7.2 Vcd	6	Pierna del Robot

<p style="text-align: center;">Atmega328</p> 	<p>Alimentación: 1.8 a 5.5V</p> <p>Consumo de energía en modo activo 0.2 mA</p>	1	Parte central del circuito
<p style="text-align: center;">Cargador Solar (Solar Charger)</p> 	Salida: 5V 3A	1	Alimentación para el circuito.

Una vez seleccionados los elementos principales diseñamos el circuito utilizando el programa Eagle (Figura 3.35).

En el circuito (Figura 3.36) podemos observar a los L298 controlando el giro de los motores de CD. Cada integrado tiene dos salidas colocándolos en paralelo se puede tener hasta 3.5 A por canal. Se debe tener en cuenta que los motores de las patas 1,3 y 5 se mueven simultáneamente, al igual que los de las patas 2, 4 y 6, es por eso que solo son necesarias cuatro salidas para controlar a los motores de las 6 patas, asignando dos para el primer grupo y dos para el segundo. De igual forma solo se requieren dos puentes H (L298) para realizar el movimiento en ambos sentidos de los 6 motores ya que, a pesar de que cada puente H puede controlar dos motores diferentes, solamente soporta 3 A en cada pin [55]. Cada uno de nuestros motores tiene un consumo máximo de corriente de 850 mA [52] por lo que es necesario conectar en paralelo las dos salidas del puente H y así obtener una corriente de salida suficiente para soportar a los tres motores.

Se observa también que se agregaron diodos a las salidas del microcontrolador con el objeto de protegerlo de corrientes de retorno (Figura 3.37). El elemento etiquetado como Q1 es un oscilador necesario para que el microcontrolador realice su trabajo y junto con el capacitor marcado en el circuito como C6, controlan la frecuencia del microcontrolador. La alimentación se hace conectando una batería solar por medio de miniUSB que nos proporciona una corriente de 3 A.

Los dispositivos marcados como B1 y B2 son puentes de diodos necesarios para controlar la corriente de los motores de CD de las patas y los marcados como R19 y R20 son resistencias necesarias también para regular la corriente.

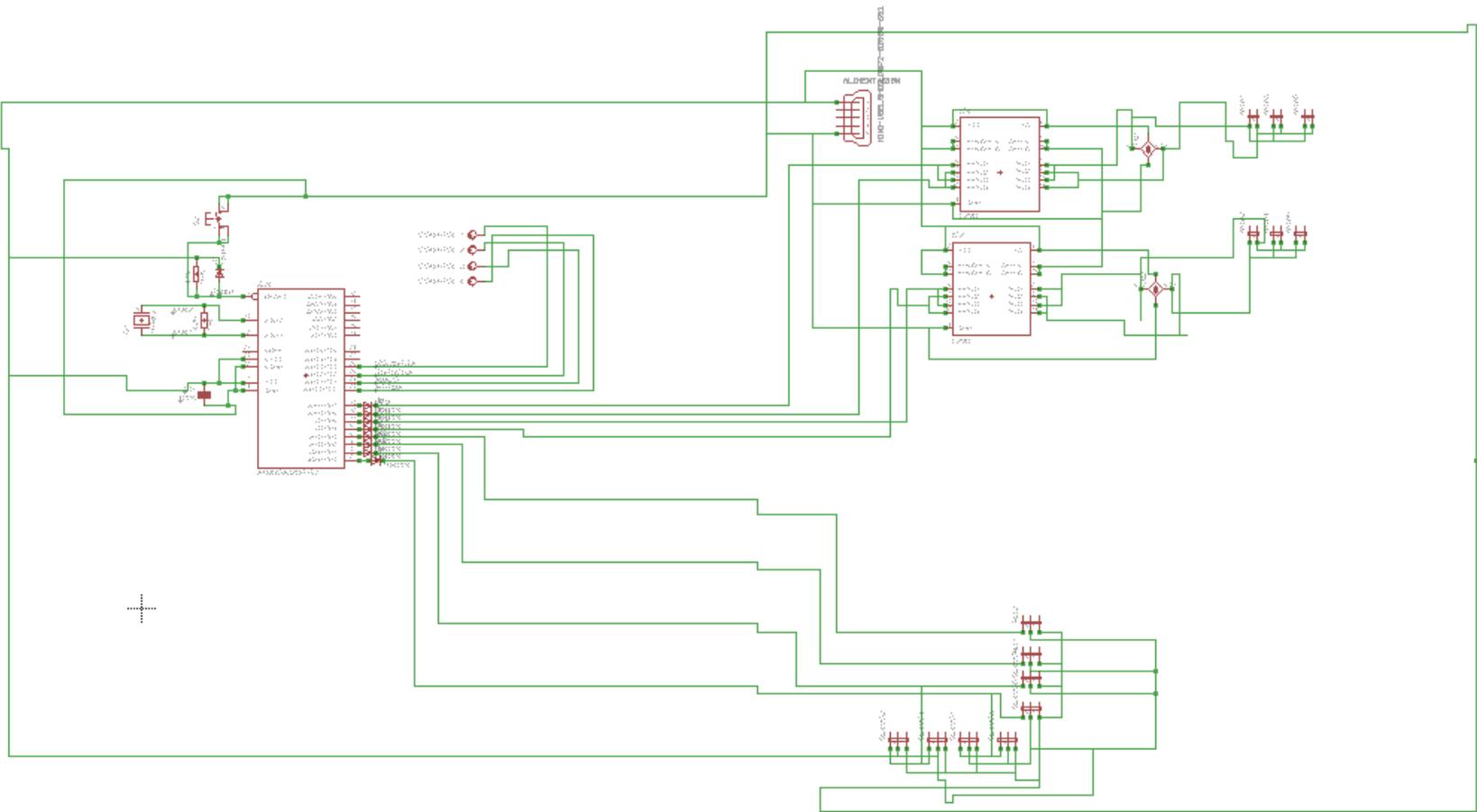


Figura 3.35 Esquema del circuito para el control del LUNAbot W6 VJ1.0

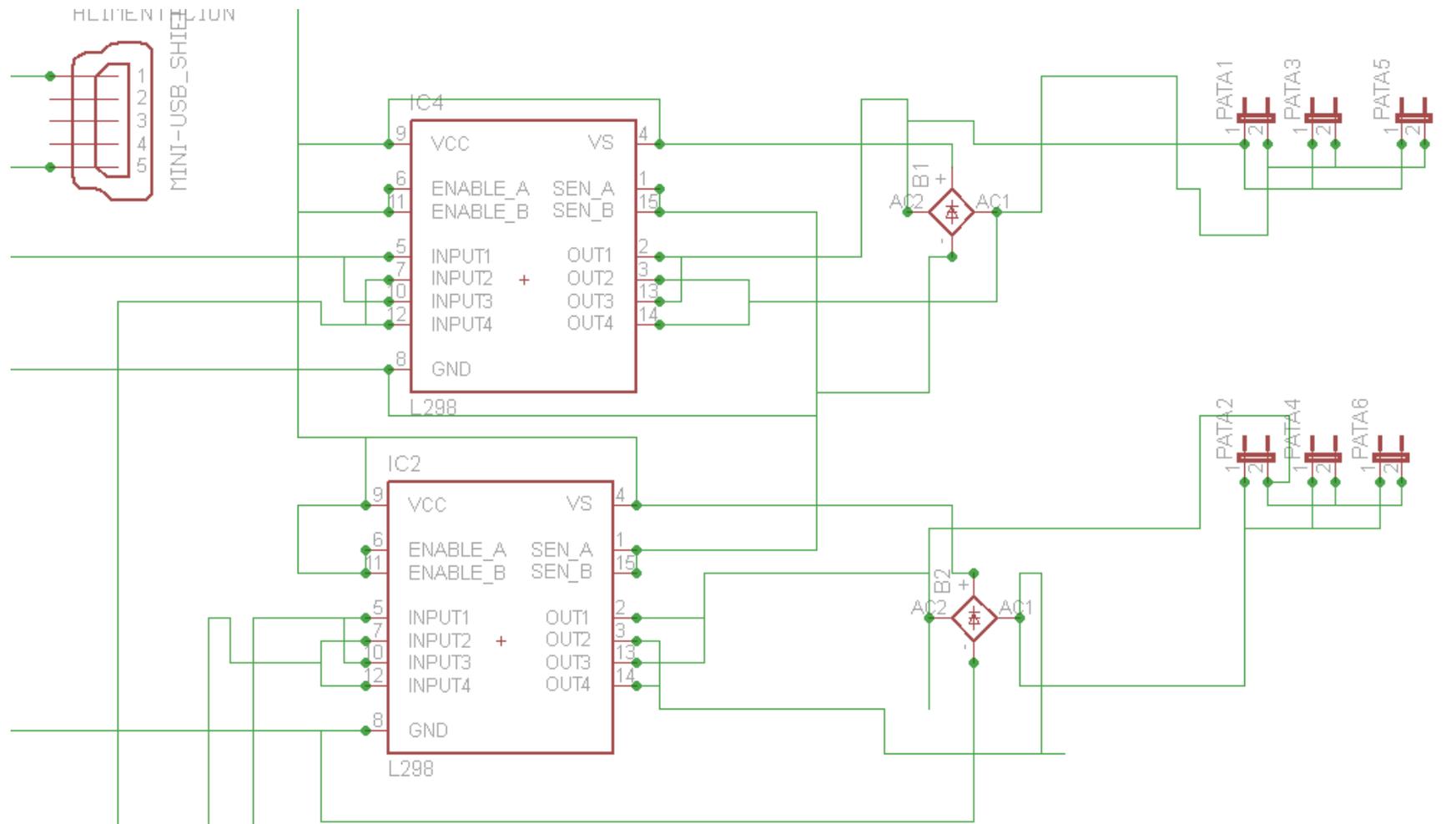


Figura 3.36 Esquema del circuito para el control del LUNAbot W6 VJ1.0 (parte de los motores).

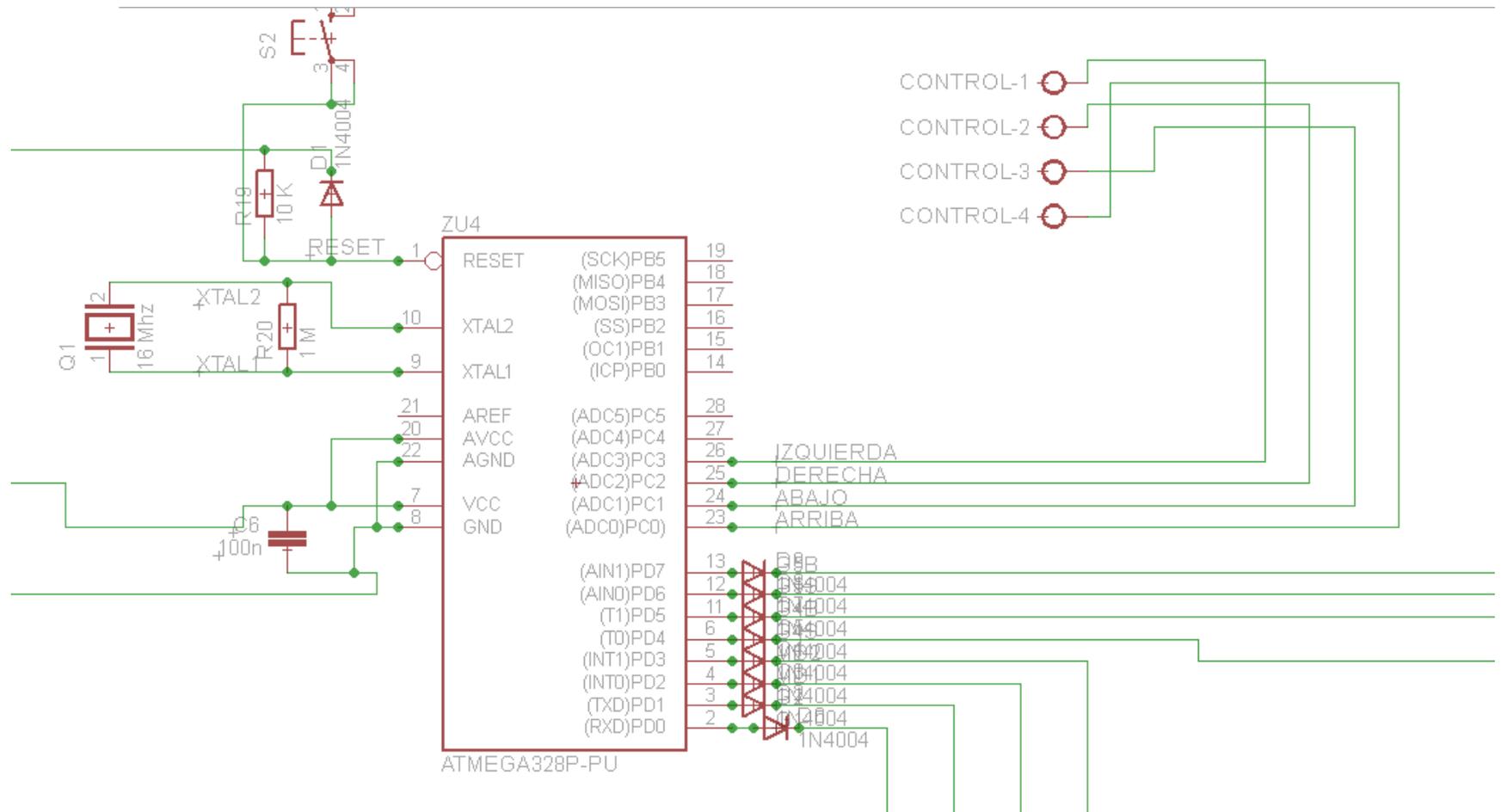


Figura 3.37 Esquema del circuito para el control del LUNAbot W6 VJ1.0 (parte del microcontrolador).

Una vez completo el diseño esquemático se debe realizar el diseño de la placa para imprimir en baquelita y crear así el circuito electrónico. (Figura 3.38)

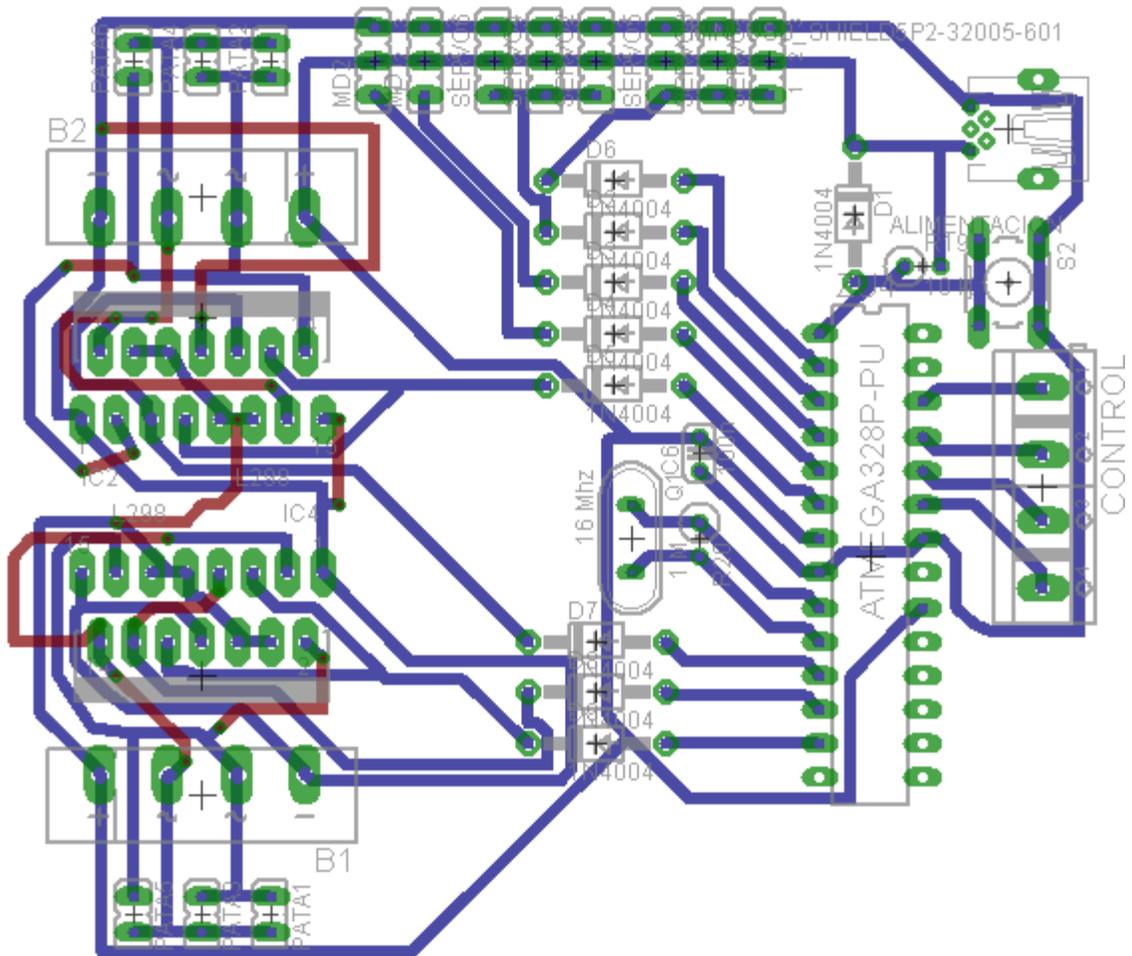


Figura 3.38 Diseño del circuito. En azul se muestran las conexiones de la parte inferior de la baquelita. En rojo se muestran las conexiones de la parte superior de la baquelita.

Para que este circuito pueda ser utilizado directamente para imprimirlo en baquelita y realizar el circuito electrónico es necesario crear un archivo en blanco y negro que muestre las pistas y los conectores. (Figura 3.39)

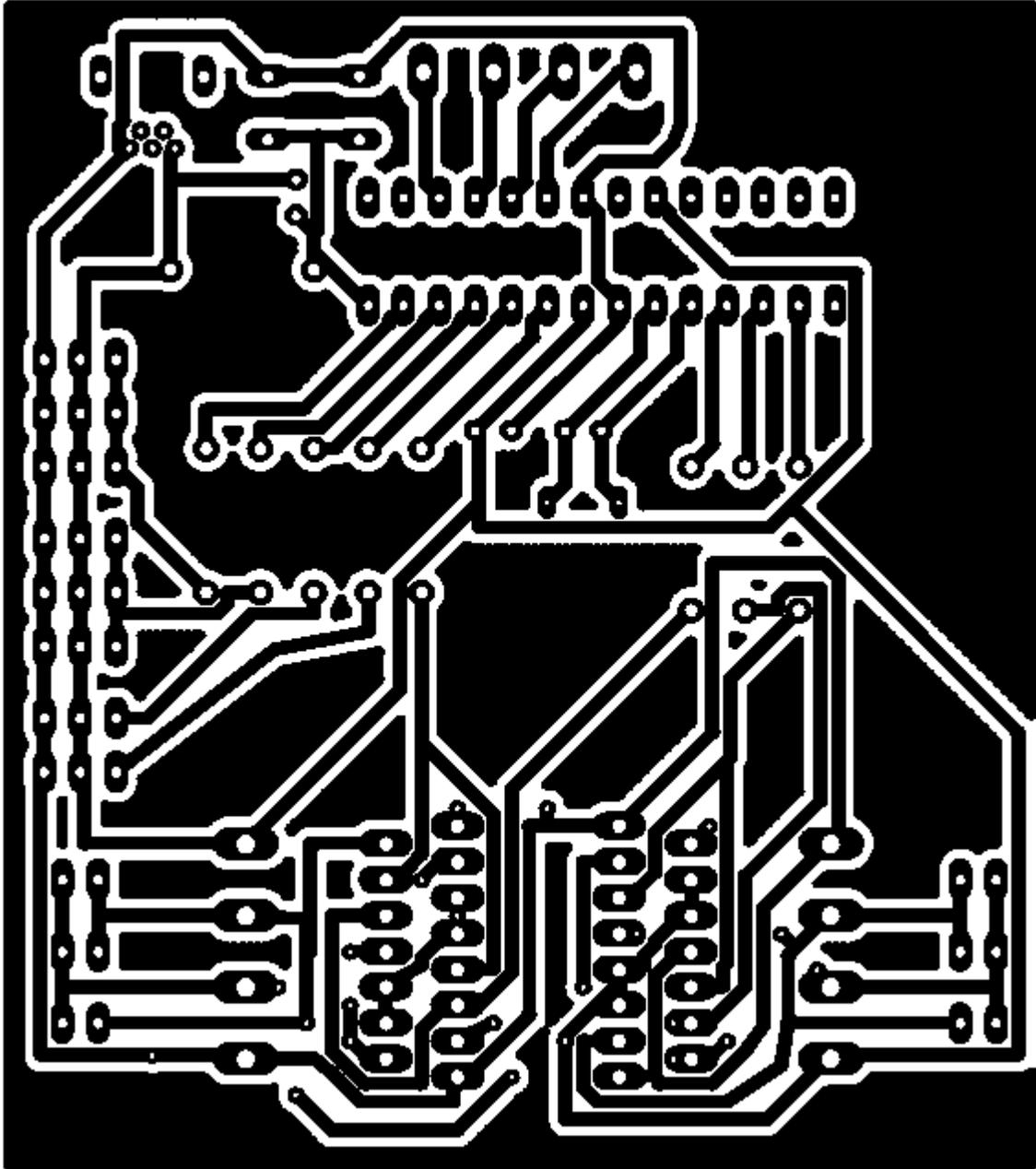


Figura 3.39 Imagen lista para imprimir el circuito del LUNAbot W6 VJ1.0

Con esto se concluye el diseño del control del LUNAbot W6 VJ1.0 cumpliendo así los objetivos planteados en el protocolo de investigación.

## Capítulo 4. CONCLUSIONES

Al analizar el mecanismo Peaucellier-Lipkin se logró obtener una función que describió el perfil de velocidad del punto B. También se observó la relación que existe entre la rotación del punto A con respecto al movimiento del punto B. Gracias a estos dos análisis se determinó que se puede controlar el movimiento del punto B variando la velocidad de rotación aplicada en el punto A.

Realizando un estudio del movimiento del robot se pudieron identificar patrones de movimiento que se repetían incluso cuando el LUNAbot giraba. Un punto importante a considerar antes de plantear las metodologías fue que el robot necesitaba mantener forzosamente tres patas en el suelo para conservar el equilibrio. Una vez analizadas estas condiciones fue posible crear metodologías de movimiento e incluso utilizarlas para crear funciones, y así facilitar la programación.

Se determinó la posibilidad de controlar el punto B variando la velocidad de rotación del punto A, se obtuvo un perfil de velocidad totalmente diferente al perfil parabólico que originalmente tenía el mecanismo Peaucellier-Lipkin al aplicar una velocidad constante de rotación. Aunque inicialmente el perfil obtenido era de forma cónica, se observó que era suficiente con realizar algunos ajustes a las funciones de membresía sin cambiar las reglas del control para obtener un perfil de velocidad trapezoidal.

Durante la investigación se eligió entre los diferentes lenguajes de programación para realizar el programa. Analizamos las ventajas y desventajas de cada lenguaje al igual que las herramientas que en cada uno podríamos utilizar. Este análisis nos llevó a elegir el lenguaje arduino, ya que este lenguaje nos permitió utilizar funciones y librerías propias del lenguaje para generar un código sencillo. El problema principal al realizar la programación fue el de lograr una codificación de los procesos de fuzzificación y defuzzificación, ya que en estos casos había que formar un polígono diferente para cada cambio de posición del

servomotor de la pierna y calcular su centroide. Es por eso que se buscó una alternativa que nos permitió realizar este cálculo partiendo de los puntos que nos brinda cada una de las funciones de membresía, lo cuales nos marcan los vértices del polígono. Con este método el resultado fue lo suficientemente exacto para ser utilizado y nos permitió ahorrar instrucciones para realizar la codificación del control.

Una vez creado el código, realizar el circuito electrónico fue bastante sencillo, ya que el programa es el que realiza todas las funciones de control, lo único pendiente de diseñar fue el circuito para que los actuadores pudieran ser controlados. Al realizar el diseño del circuito logramos que todos los elementos se pudieran conseguir de forma simple dentro del país, que fueran de fácil montaje, con el mejor precio posible, que no añadieran un peso considerable al robot, y que no ocasionaran modificaciones mayores para su inclusión en la estructura del robot.

## Bibliografía

- [1] C. Bissell, «A History of Automatic Control.,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 12 de 1812.
- [2] C. Bissell, «A History of Automatic Control.,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 13 de 1812.
- [3] H. "LVII" Legislatura del Estado de México, «Ley para la protección, integración y desarrollo de las personas con discapacidad del estado de México,» Secretaría General de Gobierno, Toluca, Estado de México, 2012.
- [4] Gobierno del Distrito Federal, «Norma Técnica Complementaria para el Proyecto Arquitectónico,» D.F. México, 2011.
- [5] Real Academia Española, Diccionario de la lengua española, España, 2014.
- [6] «Ley general para la inclusión de las personas con discapacidad,» Diario Oficial de la Federación, México, 2011.
- [7] J. Salcedo, «Movilidad, locomoción y transporte,» Madrid, 1999.
- [8] O. M. d. I. Salud, «Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud.,» Ginebra Suiza, 2001.
- [9] E. S. Taylor, An interim Report on Engineering Design, Massachusetts, 1959.
- [10] Instituto Uruhuayo de normas técnicas, «UNIT-ISO 9999:2011 Productor de apoyo para personas con discapacidad - Clasificación y terminología,» Montevideo Uruguay, 2011.
- [11] Organización Mundial de la Salud, «Documento de posición conjunta sobre suministro de dispositivos de movilidad en entornos con recursos escasos,» Ginebra Suiza, 2012.
- [12] Comisión Nacional de los Derechos Humanos, «Muestra-Diagnostico Nacional de Accesibilidad en Inmuebles de la Administración Pública Federal,» México, 2009.
- [13] Organización Mundial de la Salud, «Pautas para el suministro de sillas de ruedas manuales en entornos de menores recursos,» Ginebra Suiza, 2008.
- [14] O. M. d. I. Salud, «Pautas para el suministro de sillas de ruedas manuales en entornos de menores recursos,» p. Página 23 de 141, 2008.

- [15] K. Janowsky, «Health Policy and Systems Development,» Ginebra Suiza, 1996.
- [16] Subsecretaría de Innovación y Calidad, «Programa de Acción Específico 2007-2012 Gestión de Equipo Médico,» D.R. Secretaría de Salud, D.F. México, 2007-2012.
- [17] C. Ferrante, «Cuerpo, discapacidad y violencia simbólica: un acercamiento a la experiencia de la discapacidad motriz como relación de dominación encarnada,» Argentina, 2009.
- [18] H. Congreso de la Unión, «Ley eneral para la inclusión de las personas con discapacidad.,» Secretaria de Servicios Parlamentarios, DF, México, 2011.
- [19] Instituto Mexicano del Seguro Social, «Normas para la Accesibilidad de las Personas con Discapacidad,» D.F. México, 2000.
- [20] S. Bennett, «A Brief History of Automatic Control,» IEEE Control Systems, United Kindom, 1996.
- [21] C. Bissell, «A history of Automatic Control,» de *Springer Handbook of Automation*, 2009, p. 4 de 1812.
- [22] C. Bissell, «A Histoy of Autmatic Control,» de *A History of Automatic Control*, Nof, S.Y., 2009, p. 4.
- [23] S. Bennett, «A Brief History of Automatic Control,» de *IEEE Control Systems*, united Kindom, 1996, p. 18.
- [24] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 4 párrafo 4.
- [25] S. Bennett, «A Brief History of Automatic Control,» de *IEEE Control Systems*, United Kindom, 1996, p. 17.
- [26] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 5.
- [27] S. Bennett, «A Brief History of Automatic Control,» de *IEEE Control Systems*, United Kindom, 1996, p. 18.
- [28] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 10.
- [29] S. Bennett, «A brief History of Automatic Control,» de *IEEE Control Systems*, United Kindom, 1996, p. 20.

- [30] S. Bennet, «A Brief History of Automatic Control,» de *IEEE Control Systems*, United Kingdom, 1996, p. 20 párrafo 5.
- [31] S. Bennet, «A Brief History of Automatic Control,» de *IEEE Control Systems*, United Kingdom, 1996, p. 21.
- [32] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 12.
- [33] S. Bennett, «A Brief History of Automatic Control,» de *IEEE Control Systems*, United Kingdom, 1996, p. 17.
- [34] L. A. Zadeh, «The concept of a linguistic variable and its application to approximate reasoning,» de *Information Sciences*, 1975.
- [35] R. D. Sotelo León, Control Difuso para la Estabilidad de un Sistema Eléctrico de Potencia, D.F., México: Instituto Politécnico Nacional, 2012.
- [36] S. Yasunobu, S. Miyamoto y H. Ihara, «A Fuzzy Control for Train Automatic Stop Control,» Society of Instrument and Control Engineers, 2002.
- [37] J. M. Pinheiro Ferreira, «Lógica Difusa (Fuzzy),» 2007.
- [38] T. J. Ross, Fuzzy Logic with engineering applications, New Mexico, USA: Wiley, 2010.
- [39] E. Cox, The fuzzy systems handbook: a practitioner's guide to building, using, maintaining fuzzy systems., Boston EUA: AP Professional, 1994.
- [40] J. Vernon y M. A. Cisneros Pérez, «Sistema de lógica Difusa,» Gadalajara México.
- [41] P. S. Szczepaniak, P. J. Lisboa y J. Kacprzyk, Fuzzy Systems in Medicine, Liverpool UK: Springer-Verlag Berlin Heidelberg GmbH, 2000.
- [42] T. J. Ross, Fuzzy Logic with Engineering Applications, John Wiley & Sons, Ltd, 2004.
- [43] J. M. Angulo Usategui y I. Angulo Martínez, Microcontroladores. Diseño práctico de aplicaciones., España: Mc Graw Hill, 2003.
- [44] T. Bajad, M. Mihelj, J. Lenarcic, A. Stanovnik y M. Muni, Robotics, London-NewYork: Springer, 2010.
- [45] F. Sanchez Martín, F. Rodríguez, J. Salvador Bayarri, J. Palou Redorta, F. Rodríguez Escovar, S. Esquena Fernández y H. Villavicencio Mavrich, «Historia de la robótica,» España, 2007.

- [46] R. Barea, «Introducción a la robótica,» Departamento de Electrónica de la universidad de Alcalá, Alcalá España.
- [47] E. Lemoine, «Note sur le losange articulé du commandant du génie Peaucellier, destiné à remplacer le parallélogramme de Watt.,» France, HAL, 1873, pp. 130-134.
- [48] T. Koetsier y M. Ceccarelli, Explorations in the History of Machines and Mechanism; Proceedings of HMM2012, London-New York: Springer, 2012.
- [49] C. S. Ogilvy, Excursions in Geometry, New York EUA: Dover, 1990.
- [50] G. S. Coxeter HSM, Geometry Revisited, Washington: MAA, 1967.
- [51] R. S. Hartenberg y J. Denavit, Kinematic synthesis of linkages, New York: McGraw-Hill, 1964.
- [52] Mabuchi Motor, «FF-050SB Datasheet».
- [53] Dongguan City DS-model Servo Factory, «DS3115 Metal Digital Servo Datasheet».
- [54] K. Ogata, Ingeniería de control moderna, Minessota EUA: Pearson. Prentice Hall, 2003.
- [55] STMicroelectronics, «L298 Datasheet».
- [56] Atmel, «Atmel 8-bit Microcontroller with 4/8/16/32KBytes Datasheet».
- [57] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 2 de 1812.
- [58] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 1 de 1812.
- [59] C. Bissell, «A History of Automatic Control,» de *Springer handbook of Automation.*, Nof, S.Y., 2009, pp. 2, párrafo 2.
- [60] S. Bennett, «A Brief History of Automatic Control,» de *IEEE Control Systems*, United Kindom, 1996, p. 17 párrafo 9.
- [61] C. Bissell, «A History of Automatic Control,» de *Springer Handbook of Automation*, Nof, S.Y., 2009, p. 13.