



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA.

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES.**

**“LOCOMOCIÓN DE UN ROBOT CUADRÚPEDO BASADA EN REDES
NEURONALES ARTIFICIALES”.**

TESIS DE GRADO.

Previo a la obtención del título de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES.

Presentado por:

CRISTIAN RODRIGO MACHUCA MENDOZA.

LUIS FERNANDO LÓPEZ AYALA.

Riobamba – Ecuador.

2015

Para quienes confiaron en nosotros
incondicionalmente y han sido el motivo de
nuestro esfuerzo: Nuestras familias, amigos y maestros.

FIRMAS DE RESPONSABLES Y NOTA.

NOMBRE	FIRMA	FECHA
Ing. Gonzalo Samaniego. Ph.D.		
DECANO DE LA FACULTAD
DE INFORMÁTICA Y ELECTRÓNICA.		
Ing. Alberto Arellano.		
DIRECTOR DE LA ESCUELA DE
INGENIERÍA ELECTRÓNICA EN		
CONTROL Y REDES INDUSTRIALES.		
Ing. Gloria Vanegas.		
DIRECTORA DE TESIS.
Ing. Fernando Mejía.		
MIEMBRO DEL TRIBUNAL.
DIRECTOR DPTO.
DE DOCUMENTACIÓN.		
NOTA DE LA TESIS:	

TEXTO DE RESPONSABILIDAD.

“Nosotros **CRISTIAN RODRIGO MACHUCA MENDOZA** y **LUIS FERNANDO LÓPEZ AYALA**, somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis, y el patrimonio intelectual de la misma pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**”.

.....

CRISTIAN RODRIGO MACHUCA MENDOZA LUIS FERNANDO LÓPEZ AYALA

ÍNDICE DE ABREVIATURAS.

A	Amperios.
CAD	Diseño Asistido por Computadora.
CC	Corriente Continua.
E/S	Entradas/Salidas.
G	Centro de Gravedad.
GND	Tierra o Masa.
g	Gramos.
Hi-Z	Alta Impedancia.
HMI	Human Machine Interface.
Hz	Hertz.
IA	Inteligencia Artificial.
ICSP	In Circuit Serial Programming.
IDE	Integrated Development Environment.
I2C	Inter-Integrated Circuit.
LCD	Liquid Crystal Display.
MATLAB	MATrix LABoratory.
Mbps	Mega-bits por segundo.
mm	Mili-Metros.
mv	Mili-voltios.
mW	Mili-watts.
PC	Personal Computer.
PE	Processor Element.
PLA	Poliácido Láctico.

PMC	Perceptrón Multicapa.
PWM	Pulse-Width Modulation.
RNA	Red Neuronal Artificial.
Rpm	Revoluciones por minuto.
SD	Secure Digital.
SPI	Serial Peripheral Interface.
TFT	Thin-film transistor.
TTL	Transistor-Transistor Logic.
UART	Universal Asynchronous Receiver-Transmitter.
μs	Micro-segundos.
USB	Universal Serial Bus.
Vcc o Vcd	Voltaje de Corriente Directa.
VRML	V-Realm Builder.
°C	Grados Centígrados.
3D	Tridimensional.
2D	Bidimensional.

ÍNDICE GENERAL.

INTRODUCCIÓN

CAPÍTULO I

1. MARCO REFERENCIAL.....	16
1.1. ANTECEDENTES.....	16
1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS.....	17
1.3. OBJETIVOS.....	19
1.3.1. OBJETIVO GENERAL.....	19
1.3.2. OBJETIVOS ESPECÍFICOS.....	19
1.4. HIPÓTESIS.....	19

CAPÍTULO II

2. MARCO TEÓRICO	20
2.1. DEFINICIÓN DE ROBOT	20
2.2. ROBOTS MÓVILES.....	20
2.2.1. ROBOTS MÓVILES CON PATAS.....	21
2.2.2. ROBOT CUADRÚPEDO.....	21
2.3. TIPOS DE LOCOMOCIÓN.....	22
2.4. CENTRO DE GRAVEDAD (G).....	24
2.4.1. CENTRO DE MASA Y CENTRO DE GRAVEDAD.....	24
2.4.2. CENTRO GEOMÉTRICO Y CENTRO DE MASA.....	25
2.5. CRITERIOS DE ESTABILIDAD	25
2.5.1. ESTABILIDAD ESTÁTICA EN UN ROBOT CUADRÚPEDO.....	25

2.5.2. ESTABILIDAD DINÁMICA EN UN ROBOT CUADRÚPEDO	26
2.6. LOCALIZACIÓN DE PATAS EN ROBOTS CUADRÚPEDOS	26
2.6.1. CONFIGURACIÓN DE PATAS.....	28
2.7. REDES NEURONALES ARTIFICIALES.....	29
2.7.1. EL MODELO BIOLÓGICO.....	29
2.7.2. CARACTERÍSTICAS DE LAS REDES NEURONALES ARTIFICIALES	30
2.7.3. ESTRUCTURA BÁSICA DE UNA RED NEURONAL	31
2.7.4. FUNCIONES DE ACTIVACIÓN.....	32
2.7.5. REDES NEURONALES ARTIFICIALES DE UNA CAPA Y MULTICAPA	32
2.7.6. APRENDIZAJE DE LAS REDES NEURONALES ARTIFICIALES.....	33
2.7.7. APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES....	35
2.8. EL PERCEPTRÓN MULTICAPA (PMC).....	36
2.8.1. ARQUITECTURA DEL PERCEPTRÓN MULTICAPA	36
2.8.2. PROPAGACIÓN DE LOS PATRONES DE ENTRADA	37
2.8.3. ALGORITMO DE RETROPROPAGACIÓN	39
2.9. DISPOSITIVOS Y COMPONENTES DEL ROBOT CUADRÚPEDO	49
2.9.1. ARDUINO MEGA 2560 R3	49
2.9.2. SENSOR ULTRASÓNICO HC-SR04.....	51
2.9.3. BATERÍA LI-PO	52
2.9.4. PANTALLA LCD TFT	52
2.9.5. ACELERÓMETRO DIGITAL AXDL-345	53
2.9.6. SERVOMOTOR DYNAMIXEL AX-12A	54

2.9.7. DISPOSITIVO DE COMUNICACIÓN.....	56
2.9.8. BUFFER TRI-ESTADO.....	57
2.9.8.1. COMUNICACIÓN ENTRE LA TARJETA ARDUINO Y ACTUADORES DYNAMIXEL AX-12A MEDIANTE BUFFER TRI-ESTADO	57
2.9.8.2. FUNCIONAMIENTO DEL CIRCUITO HALF DUPLEX UART.....	58
2.9.9. MICRO SERVOMOTOR MG90S.....	59
2.9.10. MÓDULO DE TRANSMISIÓN INALÁMBRICA XBEE S1	61
2.9.11. XBEE EXPLORER USB	62
2.10. SOFTWARE.....	63
2.10.1. SOFTWARE DE DISEÑO CAD 3D SOLIDWORKS.....	63
2.10.2. MATLAB	66
2.10.2.1. CAJA DE HERRAMIENTAS DE REDES NEURONALES ARTIFICIALES.....	66
2.10.2.2. CAJA DE HERRAMIENTAS DE REALIDAD VIRTUAL	67
2.10.3. LENGUAJE DE PROGRAMACIÓN ARDUINO.....	68
CAPÍTULO III.	71
3. DISEÑO Y CONSTRUCCIÓN DEL ROBOT CUADRÚPEDO.....	71
3.1. DISEÑO CAD DEL ROBOT.....	71
3.2. CONSTRUCCIÓN DEL ROBOT.....	80
CAPÍTULO IV	
4. SIMULACIÓN Y CONTROL DEL ROBOT CUADRÚPEDO.....	82
4.1. DISEÑO DE LA SIMULACIÓN DEL ROBOT EN VRBUILD2.	82
4.1.2. CONEXIÓN DE VRBUILD2 CON MATLAB.	87

4.2. CONTROL DEL ROBOT CUADRÚPEDO.	88
4.2.1. FUENTES DE ALIMENTACIÓN.	90
4.2.2. CONEXIONES DE LOS DISPOSITIVOS.....	90
4.2.3. CONEXIÓN MATLAB-ARDUINO.....	92
4.2.4. CONEXIÓN ARDUINO-ACELERÓMETRO AXDL345.....	93
4.2.5. CONEXIÓN ARDUINO-PANTALLA.....	95
4.2.6. CONEXIÓN ARDUINO - SERVOMOTORES DYNAMIXEL AX-12.	95
4.2.7. CONEXIÓN ARDUINO – XBEE.....	96
4.3. INTERFAZ DE CONTROL Y MONITOREO DEL ROBOT.....	96
4.4. RED NEURONAL ARTIFICIAL PARA GENERACIÓN DE LOCOMOCIÓN.....	100
CAPÍTULO V	
5. COSTOS, PRUEBAS Y RESULTADOS.	101
5.1. COSTO DE LOS MATERIALES.	101
5.2. PRUEBAS REALIZADAS.	102
5.3. ANÁLISIS ESTADÍSTICO.....	103
5.4. RESULTADOS.....	106
CONCLUSIONES.	
RECOMENDACIONES.	
RESÚMEN.	
ABSTRACT.	
BIBLIOGRAFÍA.	

ÍNDICE DE FIGURAS.

Figura II-1: Clasificación de los robots móviles.	21
Figura II-2: Locomoción de un robot cuadrúpedo con 3 patas en el piso.	22
Figura II-3: Locomoción de un robot cuadrúpedo con 2 patas en el piso.	23
Figura II-4: Centro de gravedad (G) de un cubo.	24
Figura II-5: Disposición frontal de patas de un robot cuadrúpedo.	27
Figura II-6: Disposición sagital de patas de un robot cuadrúpedo.	27
Figura II-7: Disposición circular de patas de un robot cuadrúpedo.	28
Figura II-8: Configuración de patas tipo reptil para robots cuadrúpedos.	28
Figura II-9: Configuración de patas tipo arácnido para robots cuadrúpedos.	29
Figura II-10: Neurona biológica.	30
Figura II-11: Unidad básica de una Red Neuronal Artificial.	31
Figura II-12: Funciones de activación más usadas en RNA.	32
Figura II-13: Red neuronal de dos capas.	33
Figura II-14: Arquitectura genérica de una Red Neuronal Artificial.	36
Figura II-15: Conexión de neuronas de capas contiguas con sus parámetros.	38
Figura II-16: Arduino Mega 2560 R3.	50
Figura II-17: Sensor ultrasónico HC-SR04.	51
Figura II-18: LCD TFT.	53
Figura II-19: Acelerómetro AXDL345.	53
Figura II-20: Servomotor Dynamixel AX-12A.	54
Figura II-21: Asignación de pines para servomotor Dynamixel AX-12A.	56
Figura II-22: Comunicación entre servomotores y arduino.	56
Figura II-23: Circuito HALF DUPLEX - UART con buffer tri-estado 74HC126.	58
Figura II-24: Disposición de pines MG90S.	60
Figura II-25: Duración de la señal PWM para control de micro servomotor.	60
Figura II-26: Módulo Xbee Series 1.	61
Figura II-27: Xbee EXPLORER USB.	62
Figura II-28: Módulos de SolidWorks.	63
Figura II-29: Módulo pieza en SolidWorks.	64
Figura II-30: Módulo ensamblaje en SolidWorks.	64
Figura II-31: Módulo dibujo en SolidWorks.	65
Figura II-32: Entorno de programación de Arduino.	69
Figura III-1: Tapa superior de la estructura base.	72

Figura III-2: Tapa inferior de la estructura base.	73
Figura III-3: Placa lateral de la estructura base.	74
Figura III-4: Tibia conformada por dos servomotores.	74
Figura III-5: Placa lateral del pie.	75
Figura III-6: Unión inferior de las placas laterales del pie.	76
Figura III-7: Ejes para la unión de las placas laterales del pie.	76
Figura III-8: Sockets del kit Bioloid.	77
Figura III-9: Ensamblaje completo del robot cuadrúpedo.	78
Figura III-10: Distribución de componentes del robot.	79
Figura III-11: Robot cuadrúpedo terminado de construir.	80
Figura III-12: Vista isométrica del robot cuadrúpedo físico y en SW.	81
Figura III-13: Vista superior del robot cuadrúpedo.	81
Figura IV-1: Cuerpo del robot cuadrúpedo.	83
Figura IV-2: Cuerpo y articulaciones del robot cuadrúpedo.	83
Figura IV-3: Cuerpo, articulaciones y fémur del robot cuadrúpedo.	84
Figura IV-4: Cuerpo, articulaciones, fémur y tibia del robot cuadrúpedo.	84
Figura IV-5: Simulación del robot cuadrúpedo terminado.	85
Figura IV-6: Cuerpo y pata número 1 del robot cuadrúpedo.	85
Figura IV-7: Mundo virtual cargado desde matlab.	87
Figura IV-8: Diagrama funcional de la plataforma robótica.	89
Figura IV-9: Cadena margarita para la conexión de servomotores.	92
Figura IV-10: Ejes y fuerzas que actúan sobre el acelerómetro.	93
Figura IV-11: Interfaz de control y monitoreo del robot.	97

ÍNDICE DE TABLAS.

Tabla II-I: Resumen de expresiones de la Regla Delta Generalizada.	49
Tabla II-II: Características del Servomotor Dynamixel AX-12A.	55
Tabla II-III: Estados lógicos de un buffer tri-estado.	57
Tabla II-IV: Características del Micro Servomotor MG90S.	59
Tabla IV-V: Entidades de la pata 1 de la simulación.	86
Tabla IV-VI: Tabla de conexiones de dispositivos del robot cuadrúpedo.	91
Tabla V-VII: Listado de precios.	101
Tabla V-VIII: Inclinaciones medidas e ideales del robot cuadrúpedo.	103

INTRODUCCIÓN.

La robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas o no por el ser humano o que requieren del uso de inteligencia. En los diferentes campos de investigación de la robótica se encuentra el estudio de los robots móviles; robots que son capaces de desplazarse de manera autónoma por ambientes controlados. De acuerdo al diseño y funcionamiento del robot móvil, existen dos categorías: robots con ruedas y los robots con patas, donde cada categoría presenta diferentes características, ventajas y desventajas cuando se debe desarrollarlos y ponerlos en funcionamiento.

Durante los últimos años las investigaciones de la robótica móvil se han enfocado en el estudio de los diferentes sistemas de locomoción para robots con patas, basándose en estudios de locomoción biológica de animales terrestres, a partir de esto, se ha podido replicar y aplicar este complejo sistema de coordinación de articulaciones y patas en diferentes plataformas robóticas caminantes. El desarrollo y aplicación de la Inteligencia

Artificial en la robótica ha permitido desprender una gran cantidad de posibilidades cuando se debe crear un robot que tenga la capacidad de aprendizaje, y específicamente el uso de Redes Neuronales Artificiales han hecho que sea posible tener un robot autónomo con poder de decisión.

En el presente trabajo de investigación se ha desarrollado un robot cuadrúpedo de 12 grados de libertad, 3 para cada pata, sobre el cual se realizó el estudio de locomoción, la cual está basada en la Red Neuronal Artificial Perceptrón Multicapa, para la detección de obstáculos se usaron 2 sensores ultrasónicos y un acelerómetro para medir las inclinaciones. Todos los actuadores y sensores del robot están conectados a una placa Arduino Mega 2560 y la transmisión y recepción de datos desde y hacia la PC se realiza inalámbricamente con el uso de 2 módulos Xbee S1.

La programación de la Red Neuronal Artificial fue desarrollada en Matlab, desde donde se puede controlar el robot por medio de una interfaz gráfica, además se cuenta con una simulación 3D del robot cuadrúpedo diseñada en vrbuild2.

La metodología usada fue el método científico e inductivo ya que se realizaron pruebas y observaciones para después ser analizadas y poder obtener las respectivas conclusiones de la locomoción del robot cuadrúpedo.

CAPÍTULO I.

1. MARCO REFERENCIAL.

1.1. ANTECEDENTES.

El enfoque de la robótica hacia el desarrollo de máquinas caminantes con patas ha tenido un avance significativo en los últimos años, ya que algunas aplicaciones requieren de una gran movilidad y habilidad para desplazarse sobre terrenos irregulares.

La eficacia y eficiencia de la movilidad de los robots articulados con cuatro patas depende básicamente de dos factores; el diseño estructural y su algoritmo de locomoción, siendo este último la parte más compleja de resolver. A pesar de las diversas técnicas de control, los algoritmos de locomoción existentes presentan problemas a la hora de ser puestos en práctica, por lo cual durante las últimas décadas las RNA's se han utilizado de manera

acelerada resolviendo de manera óptima muchas de las deficiencias de la locomoción de los robots.

Dentro de las diversas técnicas de control de robots articulados se destacan dos enfoques: el enfoque tradicional y el enfoque neuronal. El enfoque tradicional está determinado por las ecuaciones de cinemática inversa y el modelo dinámico, mediante las cuales se trata de obtener los parámetros (posición, velocidad, torque, etc.) de cada motor de las articulaciones, para posicionar cada pata del robot en unas coordenadas determinadas anteriormente. En cambio el enfoque neuronal se basa en el aprendizaje y experiencia durante su etapa de entrenamiento, lo que permite mayor adaptabilidad a cambios en el ambiente donde se desenvuelve el robot.

En nuestro país el desarrollo de este tema de robótica ha sido poco investigado, en especial cuando se trata de robots caminantes. Existe muy poca documentación sobre trabajos realizados y los pocos robots que se han construido apenas cumplen órdenes básicas sin la capacidad de adaptarse a nuevos ambientes o cambios de condiciones del terreno sobre el cual se desplaza.

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS.

La locomoción y estabilidad en robots cuadrúpedos es una vasta área de estudio dentro del campo de la robótica que en conjunto con la Inteligencia

Artificial han tenido un gran auge durante los últimos años, creándose una cantidad considerable de robots y técnicas de control de éstos. Debido a que un robot articulado es capaz de llegar a lugares inalcanzables para un robot con ruedas se concibe la necesidad de desarrollar una plataforma robótica para el análisis de locomoción.

Las Redes Neuronales Artificiales han surgido como parte importante de los sistemas de control, convirtiéndose en una técnica alternativa a las tradicionales las cuales poseen un único punto de partida y de llegada. La adaptabilidad, la casi nula existencia de incertidumbre e imprecisiones, la construcción de una buena aproximación a los problemas a partir de un conjunto de datos significativos de entrada y el manejo de datos distintos a los de entrenamiento convierten a las Redes Neuronales Artificiales en una técnica bastante eficiente para el control de robots, debido a que permite resolver el movimiento coordinado de las articulaciones de forma simple, efectiva y que a la vez sea suficientemente flexible para adaptarse a cambios en el ambiente.

1.3. OBJETIVOS.

1.3.1. OBJETIVO GENERAL.

- Diseñar y construir un robot cuadrúpedo de 12 grados de libertad con locomoción basada en la RNA Perceptrón Multicapa.

1.3.2. OBJETIVOS ESPECÍFICOS.

- Analizar, escoger y diseñar el modelo óptimo de estructura mecánica para el robot cuadrúpedo.
- Montar y poner en marcha el sistema electrónico controlador del robot cuadrúpedo.
- Obtener, simular y construir el modelo de RNA para el aprendizaje de la locomoción.
- Entrenar la Red Neuronal Artificial hasta que logre la convergencia.

1.4. HIPÓTESIS.

“La locomoción de un robot cuadrúpedo basada en la Red Neuronal Artificial Perceptrón Multicapa permite que el robot sea estable tanto en terrenos regulares, irregulares y con pequeños obstáculos”.

CAPÍTULO II.

2. MARCO TEÓRICO.

2.1. DEFINICIÓN DE ROBOT.

Un robot es una máquina programable que tiene la capacidad de imitar el comportamiento de seres humanos o animales y que a su vez puede realizar actividades repetitivas y peligrosas, además puede integrar una gran variedad de sensores que le permite adaptarse a nuevas situaciones y entornos de trabajo (Ospina, 2012).

2.2. ROBOTS MÓVILES.

Los robots móviles son aquellos provistos de un sistema de automoción que les permite desplazarse de un sitio a otro. Se clasifican según la figura II-1.

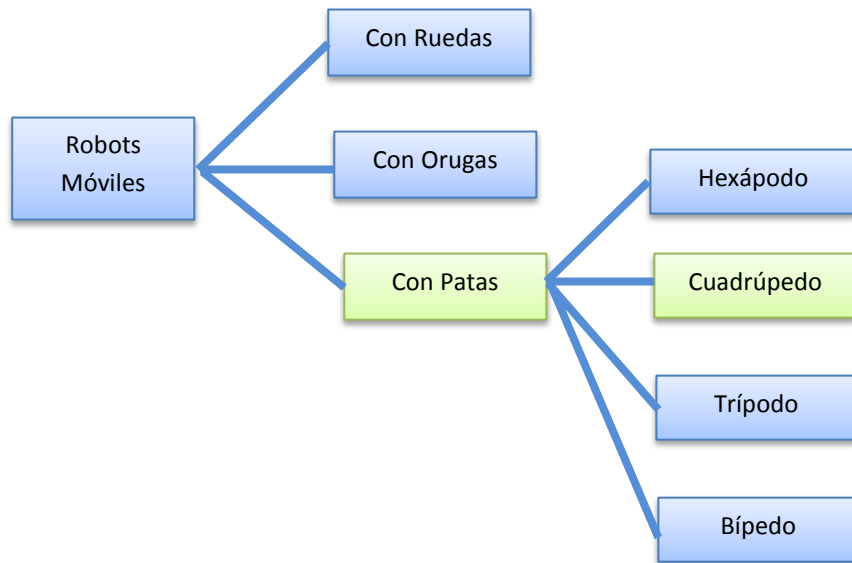


Figura II-1: Clasificación de los robots móviles.

Fuente: Los Autores.

2.2.1. ROBOTS MÓVILES CON PATAS.

Dentro del grupo de robots móviles se tiene a los robots con patas, los cuales pueden presentarse de forma antropomorfa o zoomorfa, en base a la figura II-1 se clasifican en: robots hexápodos (seis patas), cuadrúpedos (cuatro patas), trípodos (tres patas) y bípedos (dos patas).

2.2.2. ROBOT CUADRÚPEDO.

Este tipo de robot con patas ha sido el más estudiado y desarrollado en los últimos tiempos y según Sedeño (2011) un robot cuadrúpedo presenta ventajas de estabilidad cuando debe desplazarse sobre terrenos irregulares, es más fácil de controlar en comparación al robot bípedo y es menos costoso que

un robot hexápodo ya que cuenta con un menor número de patas, por lo cual se reduce el consumo de energía. Una de sus desventajas es que dependiendo de su morfología puede ser muy lento al generar locomoción, esto se puede solucionar con sistemas de locomoción basados en el andar de animales y en los diferentes tipos de diseño de su estructura.

2.3. TIPOS DE LOCOMOCIÓN.

La palabra locomoción hace referencia al desplazamiento que realiza una persona, animal, o máquina para trasladarse de un lugar a otro en el espacio.

Investigaciones recientes se han centrado en el estudio de locomoción de animales cuadrúpedos y se han identificado dos tipos básicos.

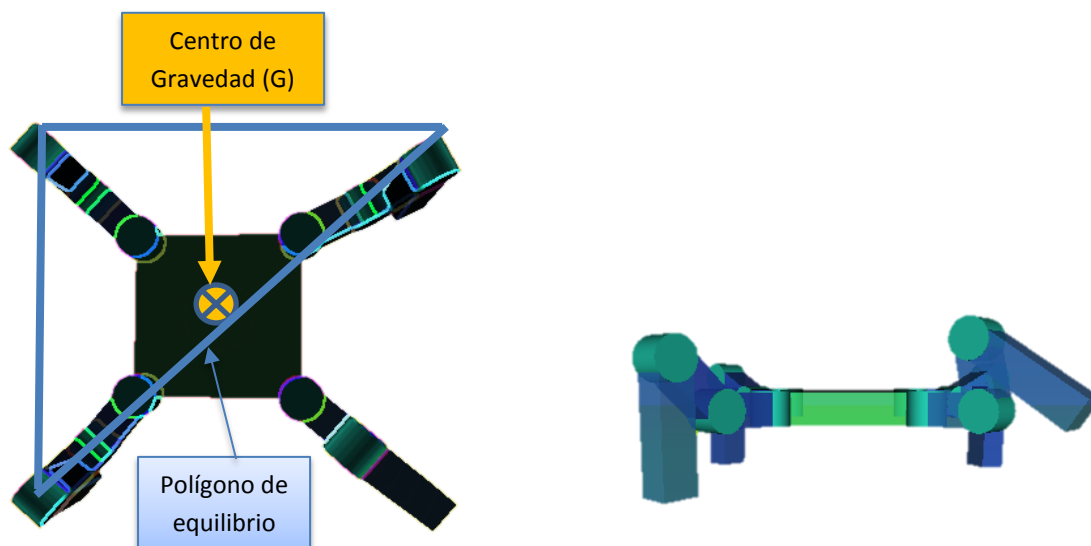


Figura II-2: Locomoción de un robot cuadrúpedo con 3 patas en el piso.

Fuente: Los Autores.

La primera locomoción se basa en mantener el centro de gravedad (G) dentro del polígono de equilibrio que se genera al estar en contacto con el piso 3 de las 4 patas del robot en un periodo de tiempo como se puede ver en la Figura II-2. Este tipo de locomoción presenta como ventaja una buena estabilidad del robot porque siempre se mantiene el centro de gravedad (G) dentro del triángulo, pero como desventaja presenta un avance lento debido a que solo se mueve una de las 4 patas a la vez.

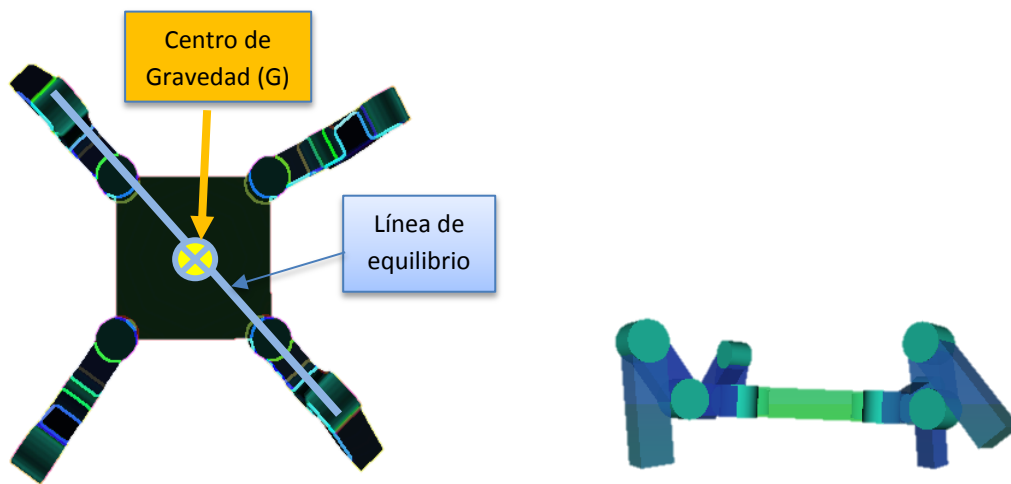


Figura II-3: Locomoción de un robot cuadrúpedo con 2 patas en el piso.

Fuente: Los Autores.

La segunda locomoción se basa en mantener el centro de gravedad en todo momento dentro de una línea de equilibrio que se forma a partir de alternar sobre el piso 2 de las 4 patas del robot en un periodo de tiempo dado como se observa en la Figura II-3. Esta locomoción presenta como ventaja una mayor velocidad de caminata pero el robot se vuelve inestable ya que el centro de gravedad (G) está sobre la línea de equilibrio y un ligero movimiento por parte

de la estructura del robot puede desbalancear todos los pesos del robot y por ende ocasionar su caída hacia uno de sus lados.

2.4. CENTRO DE GRAVEDAD (G).

El centro de gravedad (G) es el centro de distribución del peso de un cuerpo sobre el que actúa la fuerza gravitatoria. Es decir, es el punto alrededor del cual un objeto puede girar o rotar permaneciendo en un perfecto estado de equilibrio (Mazurek, 2010).

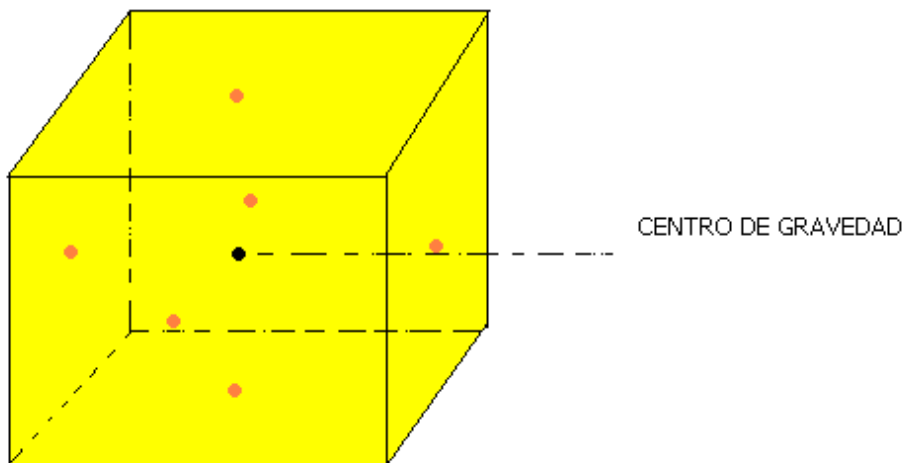


Figura II-4: Centro de gravedad (G) de un cubo.

Fuente: http://dc313.4shared.com/doc/JHeex5Pn/preview_html_1218f113.png

2.4.1. CENTRO DE MASA Y CENTRO DE GRAVEDAD.

El centro de masa y el centro de gravedad estarán en el mismo lugar sólo si el campo gravitatorio sobre el cuerpo es uniforme; es decir, en todos los puntos

del cuerpo el vector del campo gravitatorio en magnitud y dirección es constante.

2.4.2. CENTRO GEOMÉTRICO Y CENTRO DE MASA.

El centro geométrico de un cuerpo coincidirá con el centro de masa solamente si el objeto tiene densidad uniforme (homogeneidad) o si la distribución de materia es simétrica.

2.5. CRITERIOS DE ESTABILIDAD.

La estabilidad puede ser dinámica o estática. La estabilidad dinámica implica que para que el robot se mantenga de pie y no caiga debe mantenerse en movimiento, mientras que la estabilidad estática no implica que el robot deba moverse para evitar caer.

2.5.1. ESTABILIDAD ESTÁTICA EN UN ROBOT CUADRÚPEDO.

La estabilidad estática sucede cuando el centro de masa del robot está verticalmente proyectado sobre el polígono de equilibrio formado por sus 3 patas que continúan en contacto con el piso; es por esto que mientras el robot permanezca en balance estático no caerá (Beer, Johnston, Mazurek & Eisenberg, 2010).

2.5.2. ESTABILIDAD DINÁMICA EN UN ROBOT CUADRÚPEDO.

En este caso la proyección vertical del centro de masa no necesariamente debe encontrarse dentro del área de soporte formada por las 3 patas del robot cuadrúpedo, lo que hará que el robot sufra caídas pero solo por pequeños períodos de tiempo controlados.

2.6. LOCALIZACIÓN DE PATAS EN ROBOTS CUADRÚPEDOS.

El estudio biológico de animales terrestres ha permitido la adquisición y clasificación de información del movimiento y locomoción, que sirve como base fundamental cuando se debe diseñar y construir un robot cuadrúpedo.

Analizando la caminata básica de un mamífero cuadrúpedo se puede considerar que requiere gran estabilidad para no caer, pero posee una alta velocidad cuando debe desplazarse por terrenos con irregularidades, en cambio la caminata de un reptil es más lenta que la de los mamíferos pero es más estable, ya que ellos poseen un centro de gravedad más cercano a la tierra que les permite desplazarse sin dificultades sobre terrenos deslizantes, la gran desventaja que presenta su locomoción es el alto consumo de energía debido al peso del cuerpo que deben soportar las patas. La localización de las patas de un robot caminante cuadrúpedo se establece en base a los puntos de ubicación y orientación que estas presentan con respecto a la estructura teniendo las siguientes localizaciones:

Localización frontal.- Las patas se encuentran perpendiculares a la dirección de avance del vehículo.

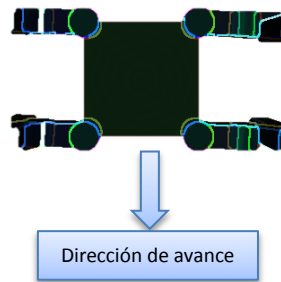


Figura II-5: Disposición frontal de patas de un robot cuadrúpedo.

Fuente: Los Autores.

Localización sagital.- Las patas se encuentran situadas paralelas a la dirección de avance del vehículo.

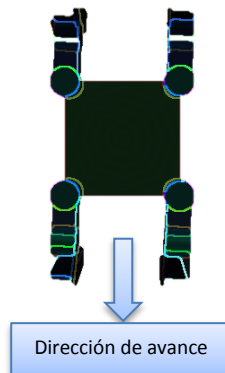


Figura II-6: Disposición sagital de patas de un robot cuadrúpedo.

Fuente: Los Autores.

Localización circular.- Las patas se sitúan alrededor del cuerpo de forma radial facilitando el desplazamiento en cualquier dirección. Esta disposición no se encuentra en la naturaleza.

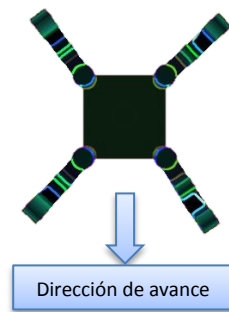


Figura II-7: Disposición circular de patas de un robot cuadrúpedo.

Fuente: Los Autores.

2.6.1. CONFIGURACIÓN DE PATAS.

Configuración tipo mamífero: Las patas se encuentran ubicadas debajo del cuerpo pudiendo presentar las rodillas de varias formas.

Configuración tipo reptil: Las patas se encuentran ubicadas hacia ambos extremos del cuerpo, sobresaliendo las rodillas a los costados del robot.

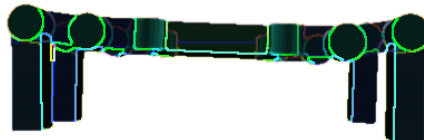


Figura II-8: Configuración de patas tipo reptil para robots cuadrúpedos.

Fuente: Los Autores.

Configuración tipo arácnido: Las patas se encuentran ubicadas hacia los costados del cuerpo, sobresaliendo las rodillas en la parte superior del robot.



Figura II-9: Configuración de patas tipo arácnido para robots cuadrúpedos.

Fuente: Los Autores.

2.7. REDES NEURONALES ARTIFICIALES.

Las Redes Neuronales biológicas que forman el cerebro han inspirado la creación de un modelo computacional llamado Redes Neuronales Artificiales (RNA) que son capaces de emular ciertas cualidades de éste, una de ellas es la capacidad de resolver problemas con la experiencia obtenida en el pasado de problemas similares, algo que con la computación tradicional no se podría realizar (Alfonso, Cazorla, Colomina, Escolano & Lozano, 2003).

2.7.1. EL MODELO BIOLÓGICO.

Las redes neuronales artificiales tienen características similares a las redes neuronales biológicas tanto en su organización y funcionamiento. La neurona es la unidad fundamental del cerebro, y se encuentra interconectada a otras por medio de sus dendritas (terminales receptoras) que son ramificaciones del axón (elemento transmisor), formando uniones llamadas sinapsis.

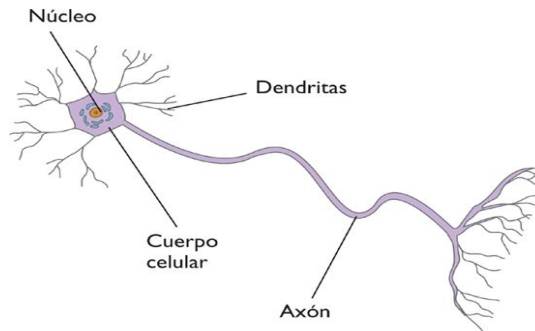


Figura II-10: Neurona biológica.

Fuente: <http://microrespuestas.com/wp-content/uploads/2012/05/partes-de-una-neurona.png>

Las comunicación entre neuronas (cada neurona se comunica con al menos otras mil por medio de la sinapsis) para la transmisión de información se realiza mediante neurotransmisores químicos captados por las dendritas los cuales dan la orden de excitación o inhibición de la acción de otras neuronas.

2.7.2. CARACTERÍSTICAS DE LAS REDES NEURONALES ARTIFICIALES.

- **Aprendizaje:** Obtienen conocimiento a través de una serie de datos de entrada, ya sea por estudio, repetición o experiencia.
- **Generalización:** Después de entrenada, la Red puede tomar decisiones aunque se le presenten datos distintos a los de aprendizaje.
- **Abstracción:** Las RNA tienen la capacidad de aislar o extraer cualidades esenciales de un conjunto de datos que aparentemente no tienen aspectos comunes, de esta forma pueden manejar datos incompletos o con distorsión.

- Memoria distribuida: La tolerancia a los fallos se debe a que el conocimiento de la red se encuentra distribuido en las múltiples conexiones existentes, lo que además le da una gran robustez comparado con la programación convencional donde un pequeño fallo puede invalidar todo el programa.

2.7.3. ESTRUCTURA BÁSICA DE UNA RED NEURONAL.

La unidad básica de una RNA es la neurona, también denominada nodo o elemento procesador (PE).

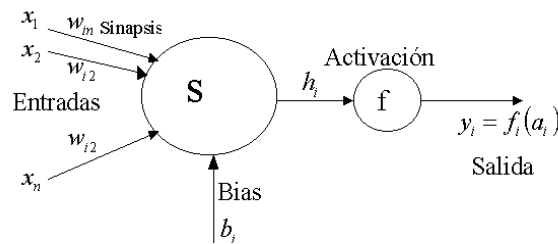


Figura II-11. Unidad básica de una Red Neuronal Artificial.

Fuente: <http://www.alergoaragon.org/ft2001/0112f02.gif>

Cada neurona posee varias entradas como puede verse en la Figura II-11, donde la suma algebraica de las x_n entradas, ponderadas con los pesos sinápticos w_{in} genera la entrada ponderada total de la neurona, donde los pesos sinápticos determinan que tan fuerte es la conexión entre las neuronas. La función de activación f modifica la diferencia entre la entrada ponderada total y el umbral b_i obteniendo la salida y_i de la neurona.

2.7.4. FUNCIONES DE ACTIVACIÓN.

Funciones de activación más usuales:

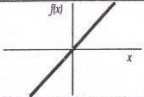
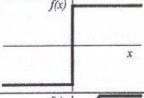
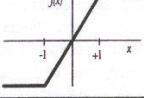
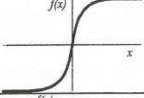
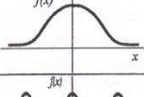
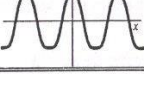
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Figura II-12: Funciones de activación más usadas en RNA.

Fuente: http://softwarelibre.unsa.edu.ar/docs/descarga/2003/curso/htmls/redes_neuronales/funca.jpg

2.7.5. REDES NEURONALES ARTIFICIALES DE UNA CAPA Y MULTICAPA.

Las múltiples conexiones que poseen las neuronas de una RNA hacen que sean muy robustas y tengan una gran capacidad de cálculo.

La red más simple es un grupo de neuronas ordenadas en una capa. Como se observa en la Figura II-13 los nodos cuadrados sólo son distribuidores de las entradas y no deben ser considerados parte de una capa.

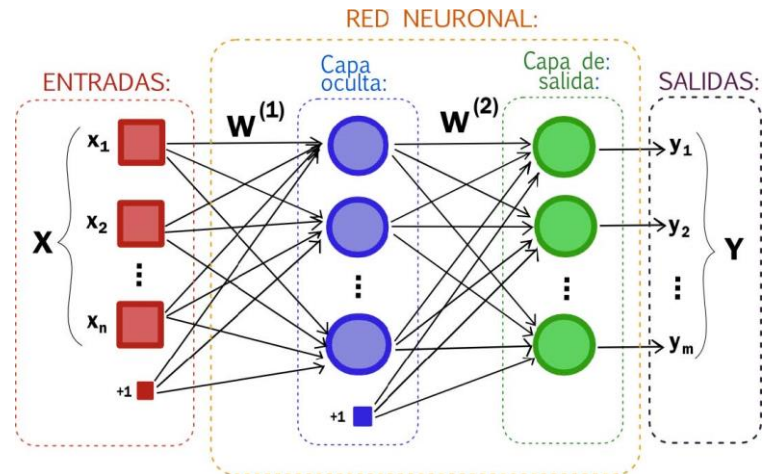


Figura II-13: Red neuronal de dos capas.

Fuente: http://ceres.ugr.es/~alumnos/esclas/imagenes/index_img_4.jpg

Cada una de las entradas se encuentra conectada, con su correspondiente peso a cada neurona de la capa siguiente, no obstante en la práctica suelen existir conexiones eliminadas, pudiendo haber o no, interacción entre neuronas de una misma capa, mientras más capas e interconexiones existan en la Red mayor será la capacidad de cálculo. Las redes multicapa están formadas por un grupo de capas intermedias, entre la capa de entrada y la capa de salida (capas ocultas), siendo la salida de una capa la entrada de la siguiente.

2.7.6. APRENDIZAJE DE LAS REDES NEURONALES ARTIFICIALES.

La capacidad de aprendizaje es una de las principales características de las RNA y una de las muchas ventajas sobre la computación tradicional, lo que permite que la red pueda modificar sus respuestas para adaptarse a estímulos externos.

Para que la RNA logre aprender, debe someterse a un proceso de entrenamiento, el cual consiste en presentar a la red una serie de valores o vectores de entrada con el objetivo que los pesos de las interconexiones se ajusten según un procedimiento predeterminado y que converjan gradualmente hacia los valores que hacen que se genere el vector de salida deseado ante un vector de entrada determinado (Flórez & Fernández, 2008).

Los entrenamientos o aprendizajes para el ajuste de valores de las conexiones de las RNA se pueden clasificar en tres grupos:

Aprendizaje supervisado: En este entrenamiento se da una serie de pares de ejemplo; es decir se le presenta a la red el vector de entradas con su respectivo vector de salidas, se calcula la salida de la red, la cual se compara con la salida deseada, la diferencia resultante entre salidas es el valor del error, el cual realimenta la red con el objetivo de que un algoritmo lo minimice mediante el ajuste de los pesos.

Aprendizaje no supervisado o auto organizado: El conjunto de valores de entrenamiento presentado a la red consiste en el vector de entradas sin adjuntar un vector de salidas, la RNA a partir de los valores ingresados extrae propiedades estadísticas, rasgos o patrones agrupándolos según su similitud.

Aprendizaje híbrido: Una RNA no está limitada a un solo tipo de entrenamiento, ya que podría existir los dos tipos de aprendizajes básicos, supervisado y no supervisado en distintas capas de neuronas.

2.7.7. APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES.

Existe un extenso campo de aplicación para las RNA debido a su adaptabilidad y robustez. Algunas de las aplicaciones comerciales son:

- Análisis financiero
 - Análisis del fracaso empresarial.
 - Predicción en los mercados financieros.
 - Tendencias financieras.

- Procesamiento de imágenes.
 - Clasificación de imágenes.
 - Reconocimiento de patrones.
 - Inspección industrial.
 - Visión de robots.

- Control y Robótica.
 - Control de robots.
 - Control de estabilidad de robots.
 - Control de posición y precisión de servomotores.

Como se puede notar las RNA han sido y pueden ser utilizadas en diversas aplicaciones, tienen un amplio desarrollo en el área militar, médica y de manufactura y se ha comprobado su buen desempeño en el procesamiento de voz, diagnósticos y biónica entre otros.

2.8. EL PERCEPTRÓN MULTICAPA (PMC).

Es uno de los tipos de RNA más comunes y es una generalización del perceptrón simple, pudiendo tener una o más capas ocultas. Surge como respuesta a la incapacidad del perceptrón simple para resolver problemas que no sean linealmente separables.

2.8.1. ARQUITECTURA DEL PERCEPTRÓN MULTICAPA.

La arquitectura del PMC se basa en la del perceptrón simple, con la diferencia de la inclusión de una o varias capas ocultas, y su característica principal es que puede funcionar como aproximador universal de funciones continuas.

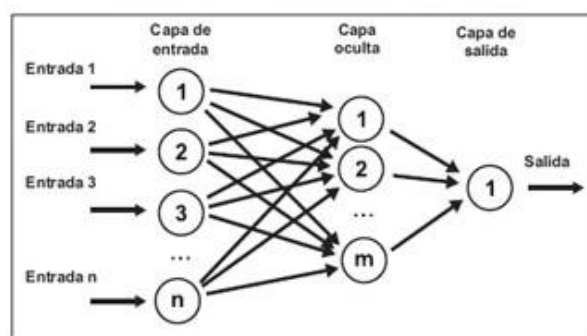


Figura II-14: Arquitectura genérica de una Red Neuronal Artificial.

Fuente: <http://www.scielo.org.bo/img/revistas/rlde/n15/a02fig06.jpg>

Las conexiones son hacia adelante, por lo general todos los nodos que forman una capa se conectan con todos los nodos de la capa siguiente (dependiendo de la red existen conexiones no consecutivas o simplemente no se conectan a otras neuronas) propagando las señales, no existen interconexiones dentro de una capa como lo muestra la Figura II-14.

2.8.2. PROPAGACIÓN DE LOS PATRONES DE ENTRADA.

El PMC propaga hacia adelante los valores del vector de entrada obteniendo así su vector de valores de salida, cada neurona al recibir la información por sus entradas, la procesa y produce una respuesta la cual es transmitida a través de sus interconexiones a la siguiente capa de neuronas que hará el mismo proceso hasta llegar a las neuronas de la capa de salida, en donde obtendremos la respuesta de la red.

Para propagar los patrones de entrada es necesario obtener la matriz de pesos y la matriz de umbrales:

$$W^c = (w_{ij}^c) = \begin{pmatrix} w_{11}^c & w_{12}^c & \cdots & w_{n_{c+1}}^c \\ w_{21}^c & w_{22}^c & \cdots & w_{2n_{c+1}}^c \\ \cdots & \cdots & \ddots & \vdots \\ w_{n_1c}^c & w_{n_2c}^c & \cdots & w_{n_cn_{c+1}}^c \end{pmatrix} \quad (1)$$

$$U^c = (u_i^c) = \begin{pmatrix} u_1^c \\ u_2^c \\ \vdots \\ u_n^c \end{pmatrix} \quad (2)$$

Donde c son las capas, n_c son las neuronas en la capa c , es la matriz de pesos de la capa c a la capa $c + 1$ y (2) es el vector de umbrales de la capa c .

Una vez obtenidos los valores anteriores se obtendrán las activaciones de las neuronas de cada capa, en este caso se obtendrá la activación de la neurona i de la capa c : a_i^c de la red de la Figura II-15.

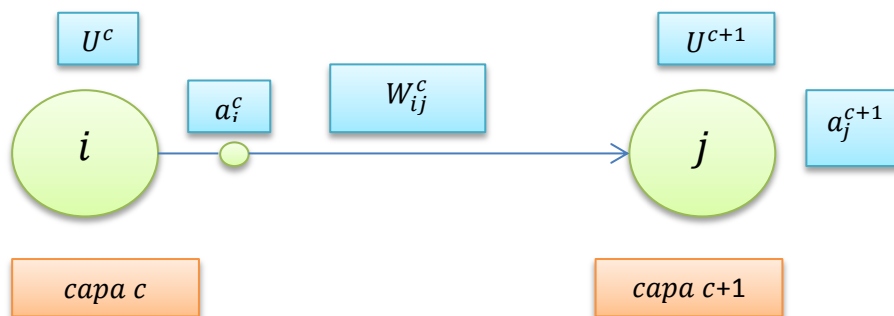


Figura II-15: Conexión de neuronas de capas contiguas con sus parámetros.

Fuente: Los Autores.

La activación de la capa de entrada consiste en:

$$a_i^1 = x_i, \text{ donde } x \text{ es cada uno de los valores de entrada.}$$

La activación de la capa oculta de neuronas corresponde a:

$$a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right), \text{ para } i = 1, 2, \dots, n_c \text{ y } c = 2, 3, \dots, C - 1 \quad (3)$$

Donde a_j^{c-1} son las activaciones de la capa $c - 1$ y f es la función de activación.

La activación de las neuronas de la capa de salida corresponde a:

$$Y_i = a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right), \text{ para } i = 1, 2, \dots, n_c \quad (4)$$

Donde $Y = (y_1, y_2, \dots, y_{n_c})$ es el vector de salida de la red.

2.8.3. ALGORITMO DE RETROPROPAGACIÓN.

El PMC es una red neuronal con aprendizaje supervisado, por lo cual se mostrará a la red el vector de entrada y el de salida, los pesos y umbrales deben ajustarse paulatinamente con el objetivo que la función de error se minimice y la salida de la red sea lo más próximo posible a la salida deseada.

El error cometido en la salida de la red es propagado hacia atrás y se transforma en un error para cada una de las neuronas ocultas, el error medio para todos los patrones está dado por:

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (5)$$

$$e(n) = \frac{1}{2} \sum_{n=1}^{n_c} (s_i(n) - y_i(n))^2 \quad (6)$$

Dónde: $s_i(n)$ es la salida deseada, $y_i(n)$ es la salida obtenida de la red y N el número de patrones.

El uso de funciones de activación no lineales y un error medio cuadrático producen una respuesta no lineal, por lo tanto el error se reduce por medio de técnicas no lineales las cuales se basan en ajustar los parámetros siguiendo una dirección de búsqueda (gradiente de error), en este caso negativa. Todos los pesos deben ajustarse según la ley de aprendizaje (Algoritmo de Retropropagación o Regla Delta Generalizada) minimizando los errores para cada patrón $e(n)$ en lugar de minimizar el error global E , matemáticamente quedaría expresada:

$$\Delta w = -\alpha \frac{\partial e(n)}{\partial w} \quad (7)$$

$$w(n) = w(n - 1) - \alpha \frac{\partial e(n)}{\partial w} \quad (8)$$

Donde α es la tasa de aprendizaje, si α es grande; el error puede oscilar alrededor del mínimo y si α es pequeño; la convergencia es más lenta.

Ahora se debe evaluar el valor del gradiente para los pesos de las neuronas de la capa de salida y está dado por:

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n - 1) - \alpha \frac{\partial e(n)}{\partial w_{ji}^{c-1}} \quad (9)$$

Para actualizar el parámetro se debe evaluar la derivada del error $e(n)$ en dicho punto:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = \frac{\partial e(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} \quad (10)$$

El peso w_{ji}^{c-1} solo afecta a y_i , el resto de salidas no depende de él.

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 = \frac{1}{2} (s_1(n) - y_1(n))^2 + \dots + \frac{1}{2} (s_i(n) - y_i(n))^2 \quad (11)$$

Entonces,

$$\frac{\partial e(n)}{\partial y_i(n)} = -(s_i(n) - y_i(n)) \quad (12)$$

Por lo tanto,

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = -(s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} \quad (13)$$

La salida y_i esta dada por:

$$y_i = a_i^c = f \left(\sum_{j=1}^{n_c} w_{ji}^{c-1} a_{ji}^{c-1} + u_i^c \right) \quad (14)$$

Solo j depende de w_{ij} , por lo tanto su derivada será:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = -(s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) a_j^{c-1} \quad (15)$$

Sea δ el término asociado a la neurona i de la capa C , está definido de la siguiente manera:

$$\delta_i^C = (s_i(n) - y_i(n))f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \quad (16)$$

Por lo tanto se tiene que:

$$\frac{\partial e(n)}{\partial w_{ji}^{C-1}} = -\delta_i^C a_j^{C-1} \quad (17)$$

De este modo el nuevo peso será:

$$w_{ji}^{C-1}(n) = w_{ji}^{C-1}(n-1) + \alpha \delta_i^C a_j^{C-1}(n) \quad (18)$$

De la misma forma ocurre para los umbrales de las neuronas de la capa de salida:

$$u_i^C(n) = u_i^C(n-1) + \alpha \delta_i^C \quad (19)$$

Una vez obtenidos los pesos y umbrales de la capa $C - 1$ a la capa oculta, se procederá a calcular los pesos y umbrales de la capa $C - 2$ a la capa $C - 1$.

$$w_{kj}^{c-2}(n) = w_{kj}^{c-2}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{kj}^{c-2}} \quad (20)$$

Todas las salidas dependen de éste peso por lo tanto la derivada del error será:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} \quad (21)$$

A continuación se calcula la derivada de y_i con respecto al peso w_{kj}^{c-2} , como y_i está dado por:

$$y_i = a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \quad (22)$$

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) w_{ji}^{c-1} \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (23)$$

Por lo tanto:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) w_{ji}^{c-1} \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (24)$$

Donde δ_i^C está dado por:

$$\delta_i^C = (s_i(n) - y_i(n))f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \quad (25)$$

Se tiene:

$$\frac{\partial e(n)}{\partial w_{kj}^{C-2}} = - \sum_{i=1}^{n_C} \delta_i^C w_{ji}^{C-1} \frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} \quad (26)$$

La derivada de la activación de la neurona de la capa $C - 1$ es:

$$\frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} = f' \left(\sum_{k=1}^{n_{C-2}} w_{kj}^{C-2} a_k^{C-2} + u_j^{C-1} \right) a_k^{C-2} \quad (27)$$

Entonces:

$$\frac{\partial e(n)}{\partial w_{kj}^{C-2}} = - \sum_{i=1}^{n_C} \delta_i^C w_{ji}^{C-1} f' \left(\sum_{k=1}^{n_{C-2}} w_{kj}^{C-2} a_k^{C-2} + u_j^{C-1} \right) a_k^{C-2} \quad (28)$$

Donde:

$$\delta_j^{C-1} = \sum_{i=1}^{n_C} \delta_i^C w_{ji}^{C-1} f' \left(\sum_{k=1}^{n_{C-2}} w_{kj}^{C-2} a_k^{C-2} + u_j^{C-1} \right) \quad (29)$$

De acuerdo a esto se tiene que:

$$\frac{\partial e(n)}{\partial w_{kj}^{C-2}} = -\delta_j^{C-1} a_k^{C-2} \quad (30)$$

Por lo tanto el nuevo peso y umbrales de la penúltima capa serán:

$$w_{kj}^{C-2}(n) = w_{kj}^{C-2}(n-1) + \alpha \delta_j^{C-1}(n) a_k^{C-2}(n) \quad (31)$$

$$u_j^{C-1}(n) = u_j^{C-1}(n-1) + \alpha \delta_j^{C-1}(n) \quad (32)$$

Se puede generalizar la ley, obteniendo:

$$w_{kj}^C(n) = w_{kj}^C(n-1) + \alpha \delta_j^{C+1}(n) a_k^C(n) \quad (33)$$

Para $k = 1, 2, \dots, n_C, j = 1, 2, \dots, n_{C+1}$ y $c = 1, 2, \dots, C-2$ donde $a_k^c(n)$ es la activación de la que parte la conexión.

La generalización para la obtención de los umbrales queda de esta manera:

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n) \quad (34)$$

Mientras que el término δ queda de esta forma:

$$\delta_j^{c+1}(n) = f' \left(\sum_{k=1}^{n_c} w_{kj}^c a_k^c + u_j^{c+1} \right) \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} w_{ji}^{c+1} \quad (35)$$

Derivando la función de activación sigmoial:

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (36)$$

$$f_1'(x) = \frac{1}{1 + e^{-x}} \left(\frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} \right) \quad (37)$$

$$f_1'(x) = f_1(x)(1 - f_1(x)) \quad (38)$$

Como δ para la última capa está dado por:

$$\delta_i^c = (s_i(n) - y_i(n))f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \quad (39)$$

Y:

$$f_1'(x) = f_1(x)(1 - f_1(x)) \quad (40)$$

Se tiene que para la última capa que:

$$\delta_i^c = (s_i - y_i)y_i(1 - y_i) \quad (41)$$

Y para el resto de capas:

$$\delta_i^{c+1} = a_j^{c+1}(1 - a_j^{c+1}) \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} w_{ji}^{c+1} \quad (42)$$

En la regla delta generalizada o algoritmo de Retropropagación cada neurona de salida distribuye hacia atrás su error a todas las neuronas que tienen conexión con ella, de esta forma, cada neurona oculta recibe un cierto error de cada neurona de salida, todos estos valores se propagan hacia atrás hasta llegar a la primera capa.

En la Tabla II-I puede observarse el resumen de expresiones para el ajuste de parámetros aplicando el algoritmo de retropropagación o regla delta generalizada.

Última capa	
Pesos	$w_{kj}^{c-1}(n) = w_{kj}^{c-1}(n-1) + \alpha \delta_j^c(n) a_k^{c-1}(n)$
Umbrales	$u_j^c(n) = u_j^c(n-1) + \alpha \delta_j^c(n)$
Valor δ	$\delta_i^c = (s_i - y_i) y_i (1 - y_i)$
Resto de capas	
Pesos	$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_j^{c+1}(n) a_k^c(n)$
Umbrales	$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n)$
Valor δ	$\delta_i^{c+1} = a_j^{c+1} (1 - a_j^{c+1}) \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} w_{ji}^{c+1}$

Tabla II-I: Resumen de expresiones de la Regla Delta Generalizada.

Fuente: Los Autores.

2.9. DISPOSITIVOS Y COMPONENTES DEL ROBOT CUADRÚPEDO.

2.9.1. ARDUINO MEGA 2560 R3.

Es una placa microcontroladora basada en el ATmega2560. Posee 54 entradas/salidas digitales (de las cuales 15 pueden ser utilizadas como salidas

PWM), 16 entradas analógicas, 4 UARTs (puertos serie por hardware), un cristal oscilador de 16 MHz, una conexión USB, un entrada de alimentación, un conector ICSP y un botón de reset .

Características principales:

- Microcontrolador ATmega2560.
- Voltaje de operación 5V.
- Voltaje de entrada (recomendado) 7-12Vcd.
- Voltaje de entrada (límite) 5-20Vcd.
- Pines E/S digitales 54 (15 proporcionan PWM).
- Pines de entrada analógica 16.
- Corriente por pin E/S 40 mA.
- Corriente por pin a 3.3Vcd 50mA.
- Velocidad de reloj de 16 Mhz.

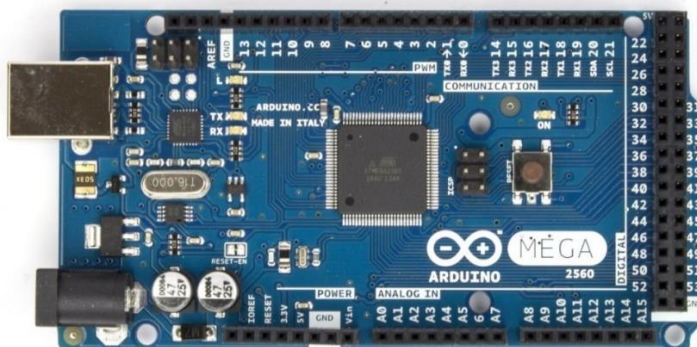


Figura II-16: Arduino Mega 2560 R3.

Fuente: http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Fronte.jpg

2.9.2. SENSOR ULTRASÓNICO HC-SR04.

Este sensor de cuatro pines (Vcc, Trig, Echo, GND) utiliza un sonar para la determinación de la distancia hacia un objeto, el pin Trig del sensor debe recibir un pulso en alto (5V) durante al menos 10us, esto lo iniciará y hará que transmita 8 ciclos de ráfagas de ultrasonidos a 40 kHz y esperará por la ráfaga ultrasónica reflejada. Cuando sea detectado el ultrasonido en el receptor, el pin Echo se pondrá en alto (5V) y pausará durante un período (ancho) proporcional a la distancia. Para obtener la distancia, se mide el ancho del pulso (Ton) del pin Echo.



Figura II-17: Sensor ultrasónico HC-SR04.

Fuente:<http://cdn.instructables.com/F90/JYNW/H7UR7RCS/F90JYNWH7UR7R>

CS.LARGE.jpg

Características principales:

- ♦ Alimentación: +5V DC.
- ♦ Corriente en Reposo: <2mA.
- ♦ Corriente de trabajo: 15mA.
- ♦ Ángulo eficaz: <15°.

- ♦ Rango de distancia: 2cm – 400 cm/1" - 13ft.
- ♦ Resolución: 0.3 cm.
- ♦ Ángulo de medida: 30°.
- ♦ Ancho de pulso del disparo de entrada: 10uS.
- ♦ Dimensiones: 45mm x 20mm x 15mm.

2.9.3. BATERÍA LI-PO.

Es una batería de polímero de litio, está formada por elementos o celdas de 3.7v conectadas en serie, puede tener entre 1 y más de 8 celdas. La descarga del amperaje de la batería está dada por la multiplicación de la corriente que suministra y el valor "C" especificado en las instrucciones de ésta. Estas baterías presentan grandes ventajas a sus antecesoras de Níquel o Níquel Cadmio en cuanto a capacidad, peso, volumen y voltaje.

2.9.4. PANTALLA LCD TFT.

Es una pantalla con retroiluminación LCD TFT en la cual se puede dibujar texto, imágenes, y formas, la pantalla posee conexión SPI y tiene una resolución de 160 pixeles de ancho por 128 pixeles de alto, tiene la capacidad de mostrar colores de 16 bits, además de poseer una ranura para tarjeta micro SD en la parte posterior.



Figura II-18: LCD TFT.

Fuente: <http://arduino.cc/en/uploads/Guide/ImageOnTFT.jpg>

2.9.5. ACELERÓMETRO DIGITAL AXDL-345.

Este dispositivo es capaz de medir aceleraciones de hasta $\pm 16g$ en los 3 ejes, posee una resolución de 13 bits por lo cual podría medir variaciones menores a 1.0° , los datos son digitales, soporta interfaces SPI o I2C, es ideal para la medición de aceleración estática, inclinación, vibración y caída libre.



Figura II-19: Acelerómetro AXDL345.

Fuente: http://5hertz.com/images/50006_A.jpg

Posee funciones para detección de aceleraciones mayores a un umbral determinado en alguno de sus ejes, detección si el dispositivo está cayendo y la de detección de taps sencillos o dobles.

2.9.6. SERVOMOTOR DYNAMIXEL AX-12A.

El servomotor Dynamixel AX-12A creado por ROBOTIS para el desarrollo e investigación en el campo robótico, es un actuador modular inteligente que incorpora un microcontrolador ATmega8 el cual es capaz de entender 50 comandos, la mayoría de los cuales fijan o leen parámetros que definen su comportamiento, como la capacidad de controlar su velocidad, temperatura, posición, tensión y carga soportada además de tener un circuito de control especializado para red.



Figura II-20: Servomotor Dynamixel AX-12A.

Fuente: <http://www.trossenrobotics.com/resize/shared/images/PImages/FRS-B-AX-12A-a.jpg?bw=1000&bh=1000>

En Tabla II-II se muestran las características generales del servomotor Dynamixel AX-12A.

CARACTERÍSTICAS AX-12A.	
Voltaje de operación	9-12 V (Recomendado 11.1)
Peso	55g
Corriente Máxima	900 mA
Temperatura de operación	-5°C ~ 85°C
Resolución	0.29°
Velocidad sin carga	59 rpm (a 12 V)
Protocolo de Comunicación	Comunicación serie asíncrona half duplex (8 bit, 1 de parada, paridad nula)
Grados de Giro	0° ~ 300° o rotación continua
Conexión Física	TTL Level Multi Drop (conector tipo daisy chain (cadena de margarita))
Velocidad de comunicación	7343bps ~ 1Mbps
Retroalimentación	Posición, temperatura, voltaje de entrada
ID	0 ~ 253
Material	Engranajes y cuerpo de plástico
Motor	Motor con núcleo CC

Tabla II-II: Características del Servomotor Dynamixel AX-12A.

Fuente: <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>

Los pines del servomotor Dynamixel AX-12A se encuentran distribuidos como se muestra en la Figura II-21, existen 2 conectores con 3 pines, la conexión que se realiza para este servomotor es en cascada (cadena de margarita).

El conector A (Figura II-21) es la entrada de uno de los servomotores o del controlador mientras que el conector B se conecta a otro servomotor o se lo puede dejar libre si es el último de un conjunto de servomotores.

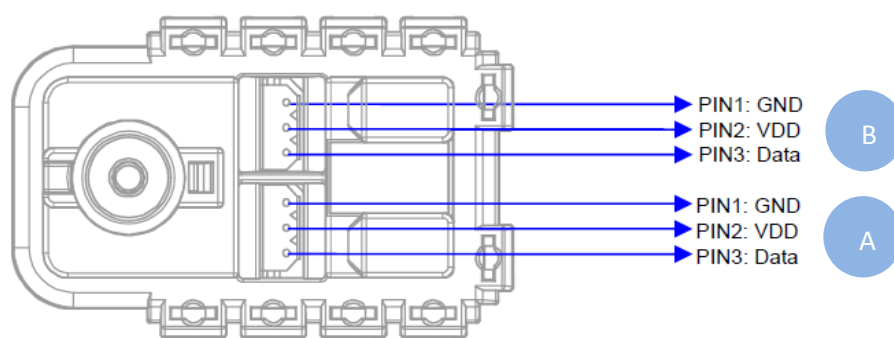


Figura II-21: Asignación de pines para servomotor Dynamixel AX-12A.

Fuente: AX-12A Technical Reference Manual.

2.9.7. DISPOSITIVO DE COMUNICACIÓN.

Un servomotor Dynamixel AX-12A puede ser conectado directamente a un módulo Arduino, pero cuando se requiere conectar varios servomotores presenta dificultades, de hecho se requiere de un circuito adicional para comunicar la tarjeta Arduino (protocolo serial full duplex) a los doce servomotores (protocolo serial half duplex), siendo necesario el uso de un buffer tri-estado como se muestra en la Figura II-22.

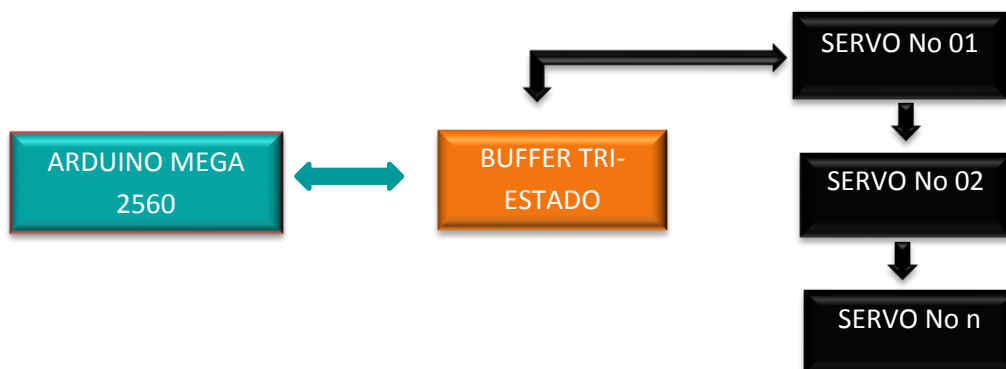


Figura II-22: Comunicación entre servomotores y arduino.

Fuente: Los Autores.

2.9.8. BUFFER TRI-ESTADO.

Es un circuito lógico con una única salida con tres estados posibles Alto (1), Bajo (0), y de alta Impedancia o Hi-Z.

Este dispositivo funciona como un interruptor, cuando en B hay un 1, funciona como si el interruptor estuviese activado, mientras que si hay en B un 0, actúa como si estuviese desactivado.

ENTRADA		SALIDA
A	B	C
0	0	Z
0	1	0
1	0	Z
1	1	1

Tabla II-III: Estados lógicos de un buffer tri-estado.

Fuente: Los Autores.

2.9.8.1. COMUNICACIÓN ENTRE LA TARJETA ARDUINO Y ACTUADORES DYNAMIXEL AX-12A MEDIANTE BUFFER TRI-ESTADO.

Para el control de los actuadores Dynamixel AX-12A mediante la tarjeta Arduino, es necesario convertir las señales emitidas por el puerto UART de la tarjeta (full duplex) a tipo half duplex y viceversa, esto se lo realiza por medio del buffer tri-estado 74HC126 con la configuración mostrada en la Figura II-23.

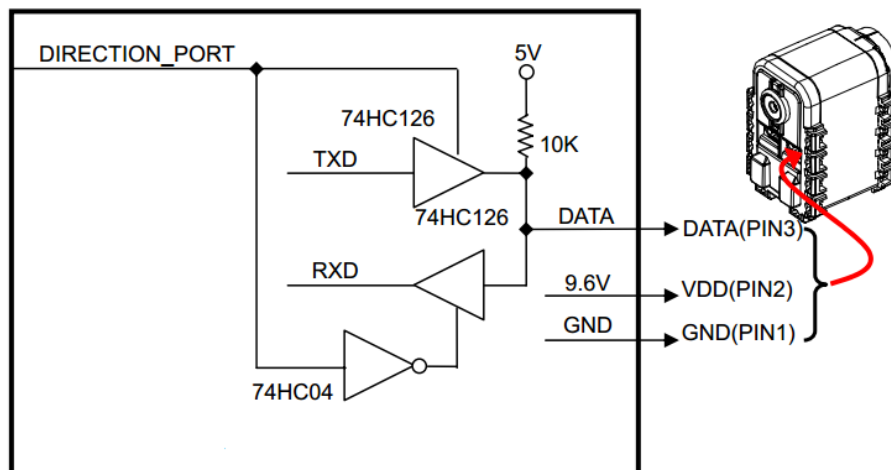


Figura II-23: Circuito HALF DUPLEX - UART con buffer tri-estado 74HC126.

Fuente: <http://www.generationrobots.com/media/Dynamixel-AX-12-user-manual.pdf>

2.9.8.2. FUNCIONAMIENTO DEL CIRCUITO HALF DUPLEX UART.

La activación de las señales de datos en TXD y RXD depende del estado que se encuentre el bit de control en el puerto de dirección (DIRECTION_PORT), teniendo lo siguiente:

- Si el puerto de dirección tiene un estado alto (1): se activa la transmisión de datos TXD.
- Si el puerto de dirección tiene un estado bajo (0): se activa la recepción de datos RXD.

2.9.9. MICRO SERVOMOTOR MG90S.

El micro servomotor MG90S de Tower Pro (Figura II-24), es un servomotor de engranajes metálicos y gran torque, presenta un tamaño compacto con un bajo consumo de energía, además posee un potenciómetro de realimentación. Este es un servomotor de posición lo cual significa que uno se le indicará a qué posición debe ir, con un rango de aproximadamente 180 grados. Tienen tres cables de conexión eléctrica; Vcc, GND, y entrada de control PWM.

Características generales:

CARACTERÍSTICAS SERVOMOTOR MG90S.	
Voltaje de operación	4.8 V a 6 V
Velocidad de Operación	0.1 s/60° (4.8 V), 0.08 s/60° (6 V)
Torque	1.8 Kgf.cm (4.8 V), 2.2 kgf.cm (6 V)
Peso	13.5 g
Resolución	0.29°
Velocidad sin carga	59 rpm (a 12 V)
Protocolo de Comunicación	Comunicación serie asíncrona half duplex (8 bit, 1 stop, no parity)
Grados de Giro	0° ~ 180°
Sistema de control	Ancho de Pulso
Material	Engranajes metálicos y cuerpo de plástico
Dimensiones	Largo 22.5 mm, ancho 12 mm, altura 35.5 mm aprox.

Tabla II-IV: Características del Micro Servomotor MG90S.

Fuente: <http://www.electronicoscaldas.com/motores/451-micro-servo-motor-mg90s.html>

En la Figura II-24 se muestra la disposición de pines del micro servomotor.

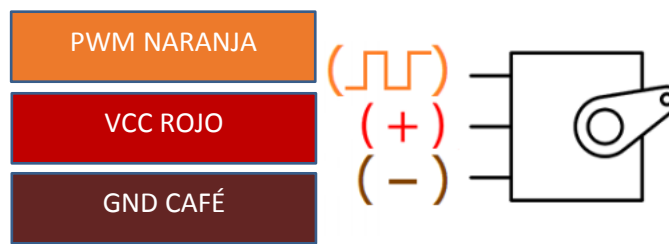


Figura II-24: Disposición de pines MG90S.

Fuente: Los Autores.

El micro servomotor presenta un circuito de control interno que depende del ancho del pulso que se envíe en el pin de control para posicionarlo y está dado de acuerdo al tiempo que esté la señal PWM en nivel alto (4.8 V y 6 V), aproximadamente los niveles varían de 0.6 a 2ms para 0 y 180° respectivamente como se puede ver en la Figura II-25:

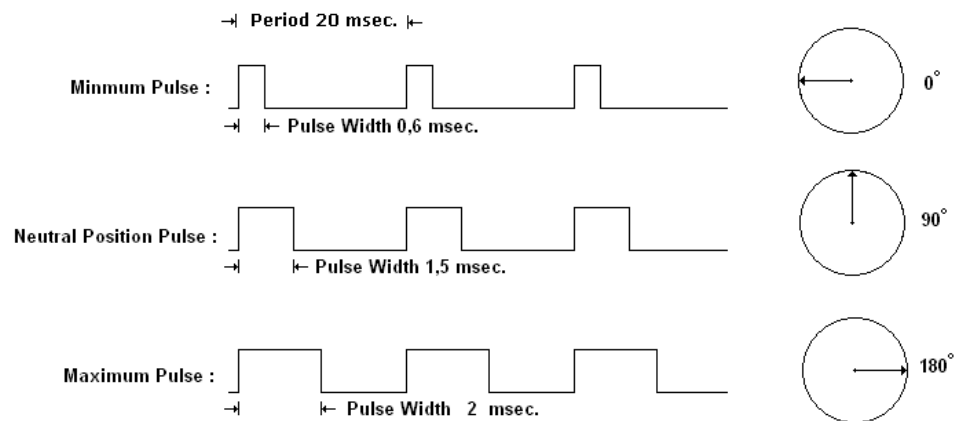


Figura II-25: Duración de la señal PWM para control de micro servomotor.

Fuente: http://picmania.garcia-cuervo.net/images/hitec_servo_rc_pwm.gif

2.9.10. MÓDULO DE TRANSMISIÓN INALÁMBRICA XBEE S1.

Xbee permite comunicar inalámbricamente diferentes dispositivos, ya sean industriales o domóticos. Los módulos Xbee utilizan el protocolo de comunicación inalámbrica IEEE 802.15.4 mejor conocido como ZigBee.

Cada uno de los dispositivos Xbee tiene una dirección única de 64 bits que está dada por defecto de fábrica.

Existen diferentes topologías de red para comunicar los módulos Xbee entre éstas; punto a punto, punto a multipunto, estrella y malla.

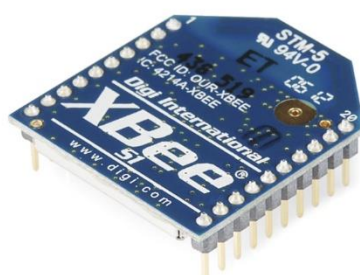


Figura II-26: Módulo Xbee Series 1.

Fuente: <http://5hertz.com/images/10401.jpg>

El módulo Xbee Series 1 presenta las siguientes características:

- 250kbps máxima velocidad de datos
- 1mW de salida
- 100 m rango de línea abierta, 30 metros en interiores
- 6 pines de 10 bits para entrada ADC

- 8 pines de E/S digitales
- Bajo consumo < 50 mA cuando están en funcionamiento y <10 uA cuando están en modo sleep.
- Voltaje de operación 2.8 V a 3.3 V.

2.9.11. XBEE EXPLORER USB.

Xbee Explorer USB es una tarjeta que sirve para adaptar los transmisores inalámbricos Xbee a una computadora por medio de conexión USB mediante cable, en sus presentaciones estándar soporta Xbee series 1, 2 y pro, no necesita alimentación externa ya que la toma directamente del puerto USB.



Figura II-27: Xbee EXPLORER USB.

Fuente: <https://www.sparkfun.com/products/retired/8687>

Características generales:

- Conversor USB a Serial.
- Conector estándar para Xbee S1, S2, Pro.
- Led indicadores (TX, RX).

2.10. SOFTWARE.

2.10.1. SOFTWARE DE DISEÑO CAD 3D SOLIDWORKS.

SolidWorks es una herramienta CAD (Diseño Asistido por Computadora) utilizado para el diseño mecánico en 3D.

El software se usa para diseñar y modelar piezas mecánicas, además presenta una herramienta de ensamblaje y otra para desarrollar planos técnicos de las piezas o conjuntos desarrollados. Se puede realizar estudio de movimiento de ensamblaje es decir animar movimiento de piezas.

SolidWorks presenta 3 módulos: Pieza, Ensamblaje y Dibujo tal como se muestra en la Figura II-28.

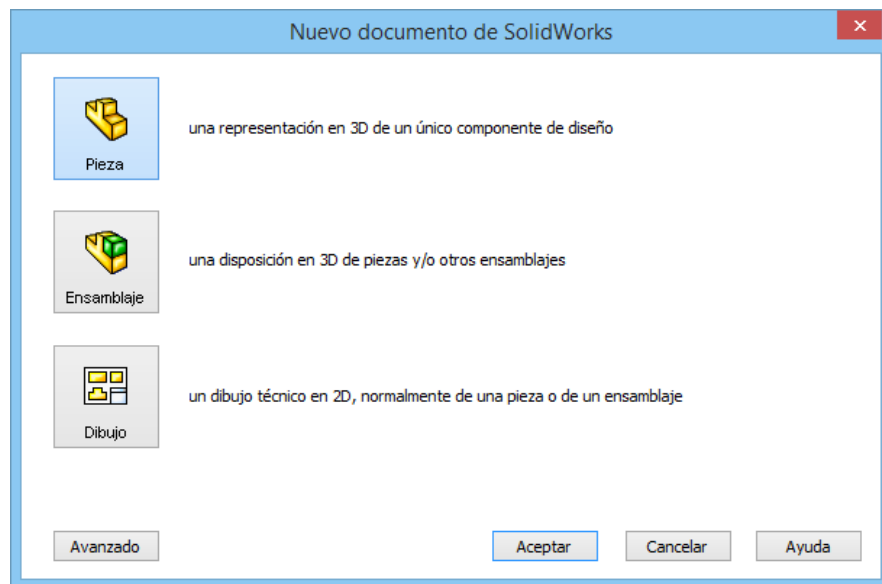


Figura II-28: Módulos de SolidWorks.

Fuente: Los Autores.

Módulo pieza: En este módulo se realiza una pieza en 3D utilizando las diferentes herramientas que presenta este software, las cuales luego pueden ser ensambladas para formar un dibujo más detallado, complejo y completo.

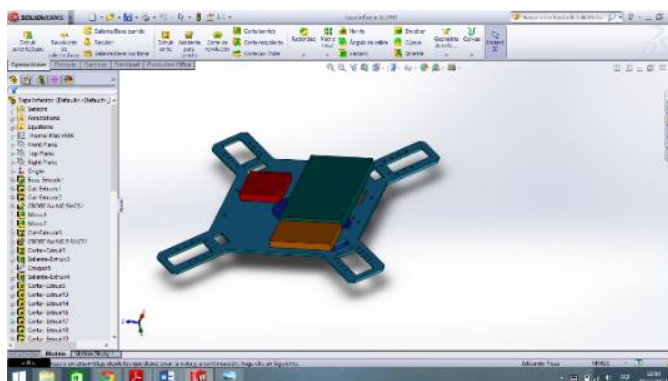


Figura II-29: Módulo pieza en SolidWorks.

Fuente: Los Autores.

Módulo ensamblaje: En este módulo se ubican las piezas realizadas mediante la inserción de componentes, se coloca cada pieza en un mismo módulo y mediante la herramienta de Relación de Posición se realiza el ensamblaje de todas las piezas dándoles una función específica.

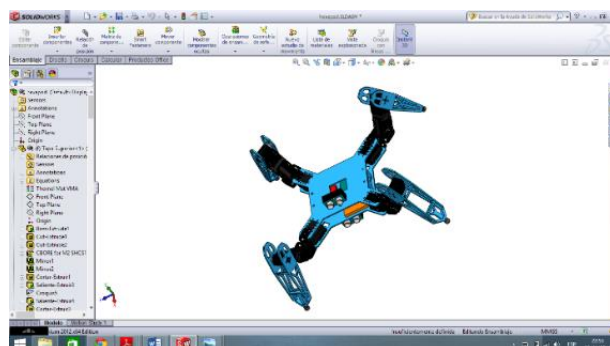


Figura II-30: Módulo ensamblaje en SolidWorks.

Fuente: Los Autores.

Módulo dibujo: En este módulo se procede a realizar un plano técnico en 2D del mecanismo realizado donde se puede ubicar las acotaciones del mismo mediante la herramienta de Cota Inteligente.

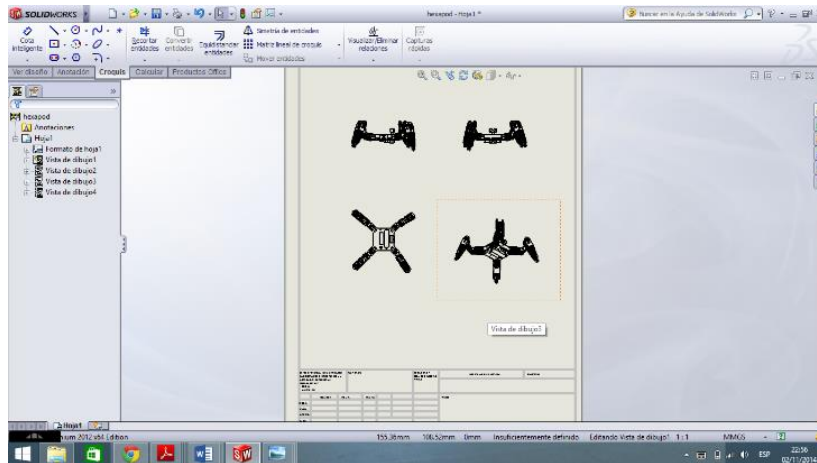


Figura II-31: Módulo dibujo en SolidWorks.

Fuente: Los Autores.

El entorno de SolidWorks presenta un área de dibujo bien definida. Se presentan 3 planos: Alzado, Planta y Vista Lateral, que permiten desarrollar el diseño a partir de cualquiera de esas vistas además posee herramientas para visualizar el dibujo desde cualquier ángulo o posición.

Para la creación de los modelos presenta una gran cantidad de herramientas funcionales que permiten desarrollar modelos en 2D y 3D.

2.10.2. MATLAB.

MATLAB es una herramienta de software (IDE) de programación para la resolución de cálculos matemáticos complejos y el tratamiento de datos, es usado en el área de ingeniería por sus grandes capacidades de cálculo, capacidad de comunicarse con otros lenguajes y hardware, representación y análisis de gráficas, creación de interfaces gráficas de usuario.

Presenta una gran cantidad de cajas de herramientas entre ellas la de creación de mundos virtuales y la de creación y entrenamiento de redes neuronales artificiales.

2.10.2.1. CAJA DE HERRAMIENTAS DE REDES NEURONALES ARTIFICIALES.

La caja de desarrollo Neuronal Network Toolbox de MATLAB permite la creación de Redes Neuronales Artificiales y entrenarlas hasta que logren la convergencia, para crear una RNA con conexiones hacia adelante se utiliza la función *newff*, los argumentos de entrada para esta función son:

R: Es una matriz de $R \times 2$, el número de filas indica el número de entradas de la red, y se debe señalar cual es el valor máximo y mínimo que puede tomar cada una de las entradas de la red.

[Slayer1, Slayer2,...]: Este vector define el número de salidas y el número de neuronas de las capas ocultas, por lo tanto también se define el número de capas ocultas de la red.

{Funciones}: Se indica la función de activación para cada una de las capas ocultas de la red neuronal artificial.

El último argumento de la función es el indicador de que tipo de entrenamiento va a recibir la red.

Cuando ya se ha definido la red y sus argumentos y se ha obtenido los vectores de entrada y salida se procede a realizar el entrenamiento con la función *sim*.

2.10.2.2. CAJA DE HERRAMIENTAS DE REALIDAD VIRTUAL.

“La caja de desarrollo VRML *vrbuild2* es un software usado para diseñar mundos virtuales y dibujar e importar objetos virtuales” (Khaled, 2012). Una vez creado el mundo virtual se puede manipular los objetos y la escena virtual a través de los comandos de MATLAB. Las propiedades de los objetos como rotación, traslación, escala y color también pueden ser cambiados por medio de MATLAB.

2.10.3. LENGUAJE DE PROGRAMACIÓN ARDUINO.

La tarjeta Arduino se programa mediante el uso de un lenguaje de alto nivel basado en C++ (en Windows).

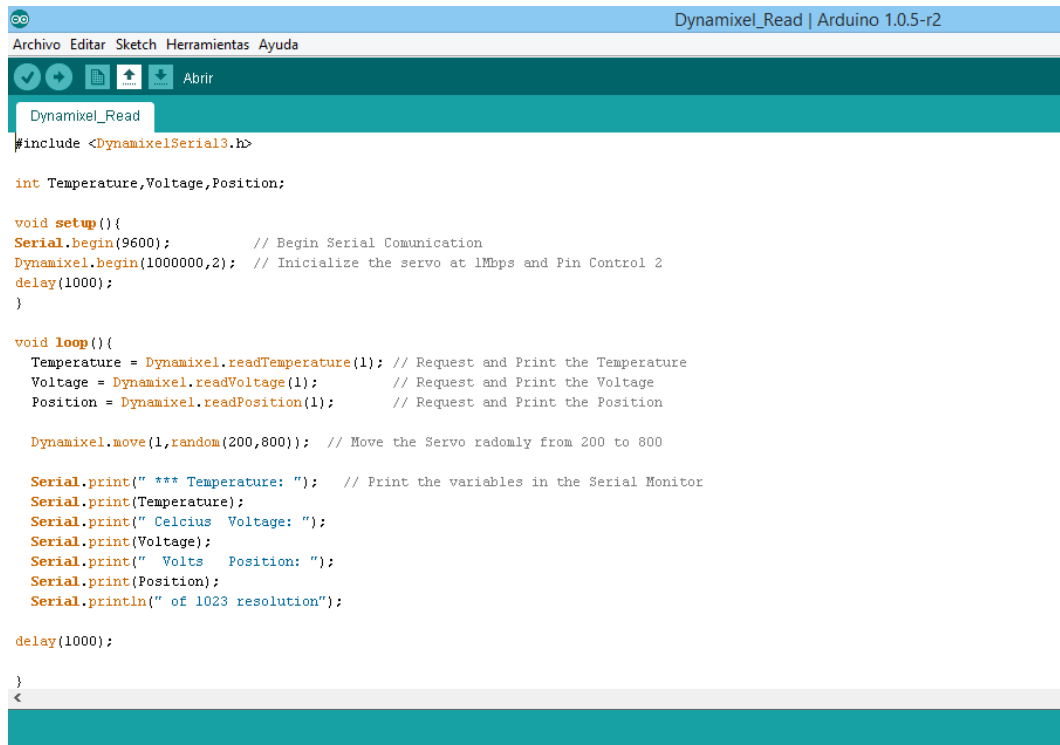
También es posible utilizar otros lenguajes de programación gracias a que Arduino usa transmisión serial de datos soportada por la mayoría de los lenguajes entre éstos:

- C#
- Matlab
- Flash (Mediante ActionScript)
- Java
- Processing

El entorno de programación presentado por Arduino (Figura II-32) se divide en dos funciones básicas que encierran los bloques de declaración; void setup () y void loop().

Void setup (): La función o bloque setup(), por lo general debe contener la declaración de todas las variables a utilizar por el programa, direcciones de dispositivos, configuración de comunicaciones y modo de los pines E/S.

Este bloque es el primero en ejecutarse y lo hace durante una sola vez al iniciar el dispositivo.

The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads 'Dynamixel_Read | Arduino 1.0.5-r2'. Below the title bar is a menu bar with 'Archivo', 'Editar', 'Sketch', 'Herramientas', and 'Ayuda'. A toolbar with icons for 'Guardar', 'Abrir', and 'Subir' is visible. The main text area contains the following C++ code:

```
#include <DynamixelSerial3.h>

int Temperature,Voltage,Position;

void setup(){
  Serial.begin(9600);           // Begin Serial Communication
  Dynamixel.begin(1000000,2);  // Inicialize the servo at 1Mbps and Pin Control 2
  delay(1000);
}

void loop(){
  Temperature = Dynamixel.readTemperature(1); // Request and Print the Temperature
  Voltage = Dynamixel.readVoltage(1);        // Request and Print the Voltage
  Position = Dynamixel.readPosition(1);      // Request and Print the Position

  Dynamixel.move(1,random(200,800)); // Move the Servo radomly from 200 to 800

  Serial.print(" *** Temperature: "); // Print the variables in the Serial Monitor
  Serial.print(Temperature);
  Serial.print(" Celcius Voltage: ");
  Serial.print(Voltage);
  Serial.print(" Volts Position: ");
  Serial.print(Position);
  Serial.println(" of 1023 resolution");

  delay(1000);
}
```

Figura II-32: Entorno de programación de Arduino.

Fuente: Los Autores.

Void loop (): La función o bloque loop (), se ejecuta después de la función setup (), este bloque se ejecuta continuamente leyendo entradas y activando salidas, esta parte de bloque es el que se genera la mayor parte de trabajo en el programa.

Sintaxis básica:

- Delimitadores: `;`, `{}`
- Comentarios: `//`, `/* */`
- Cabeceras: `#define`, `#include`
- Operadores aritméticos: `+`, `-`, `*`, `/`, `%`
- Asignación: `=`
- Operadores de comparación: `==`, `!=`, `<`, `>`, `<=`, `>=`
- Operadores Booleanos: `&&`, `||`, `!`
- Operadores de acceso a punteros: `*`, `&`
- Operadores de bits: `&`, `|`, `^`, `~`, `<<`, `>>`
- Operadores compuestos:
- Incremento y decremento de variables: `++`, `--`
- Asignación y operación: `+=`, `-=`, `*=`, `/=`, `&=`, `|=`

Estructuras de Control:

- ◆ Condicionales: `if`, `if...else`, `switch case`.
- ◆ Bucles: `for`, `while`, `do... while`.
- ◆ Bifurcaciones y saltos: `break`, `continue`, `return`, `goto`.

Tipos de datos: `boolean`, `char`, `unsigned char`, `byte`, `int`, `unsigned int`, `word`, `long`, `unsigned long`, `float`, `double`, `string`, `array`.

CAPÍTULO III.

3. DISEÑO Y CONSTRUCCIÓN DEL ROBOT CUADRÚPEDO.

En este capítulo se presenta el diseño CAD en SolidWorks 2012 de cada uno de los componentes que conforman la estructura del robot, sus dimensiones y las funciones que deben cumplir, además se muestra la distribución interna de los dispositivos electrónicos y la presentación final del robot una vez construido.

3.1. DISEÑO CAD DEL ROBOT.

A continuación se presentan las diferentes partes y piezas que componen el robot con sus respectivos planos.

Estructura base: Está conformada por la tapa superior (Figura III-1) y la tapa inferior (Figura III-2), es la encargada de contener los dispositivos electrónicos, baterías y circuitería, necesarios para el funcionamiento del robot. Cuatro de los doce servomotores se encuentran alojados dentro de la estructura base.

La figura III-1 muestra la tapa superior, en cada uno de los cuatro vértices posee un apéndice el cual posee diez orificios que están destinados a sostener al servomotor por medio de tornillos. En la parte central de la tapa existe un orificio rectangular donde se montará la pantalla LCD TFT, la profundidad de esta tapa y de la tapa inferior es de 3mm lo que brinda buena resistencia mecánica y rigidez a las piezas.

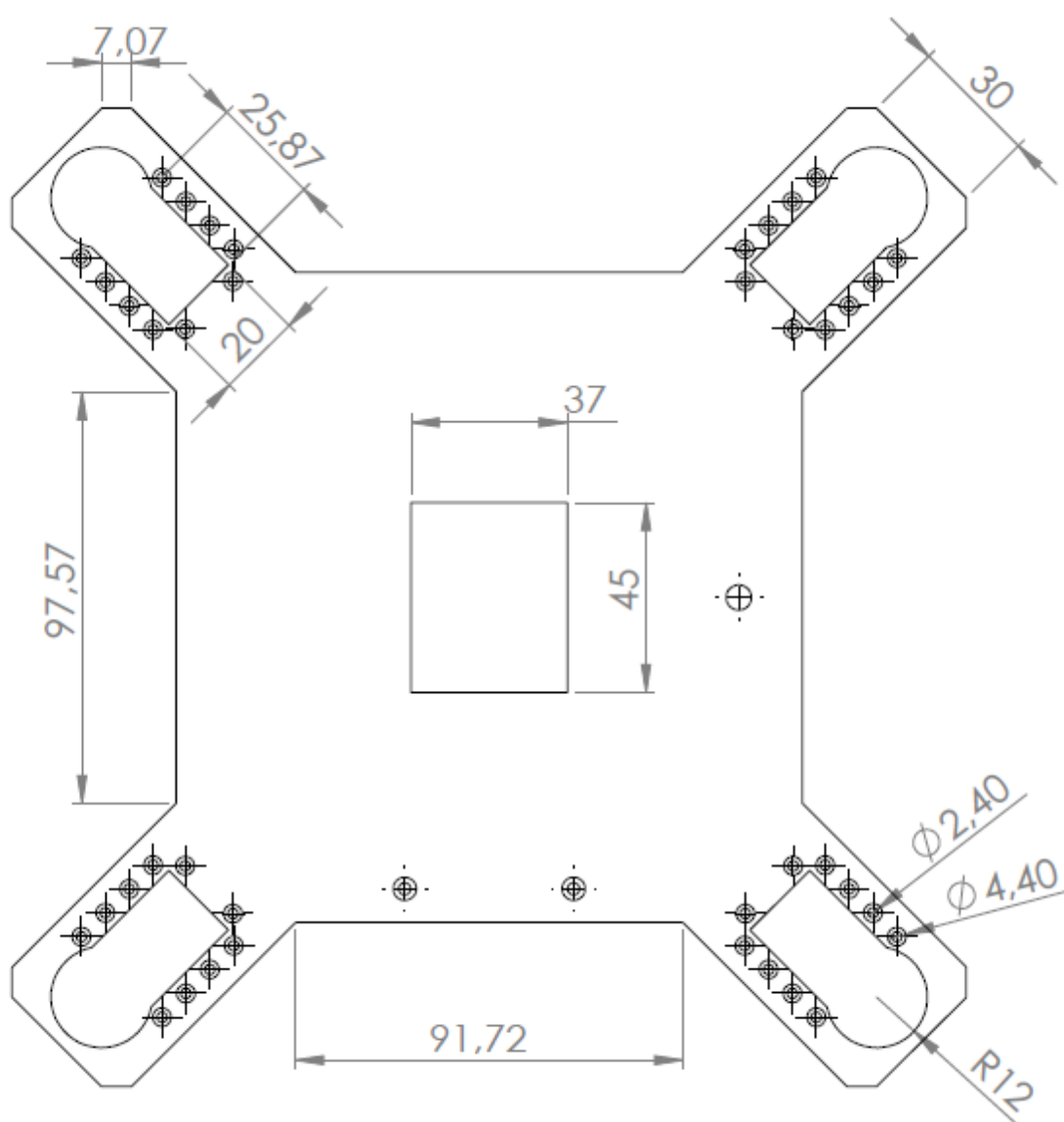


Figura III-1: Tapa superior de la estructura base.

Fuente: Los autores.

Como se puede ver la tapa inferior según lo muestra la figura III-2 es muy similar a la tapa superior, con la diferencia que ya no posee un orificio rectangular sino que es circular y tiene como función dar acceso a los dispositivos alojados dentro del robot, sobre los apéndices se colocan los servomotores que atornillados también a la tapa superior mantienen fija la estructura base.

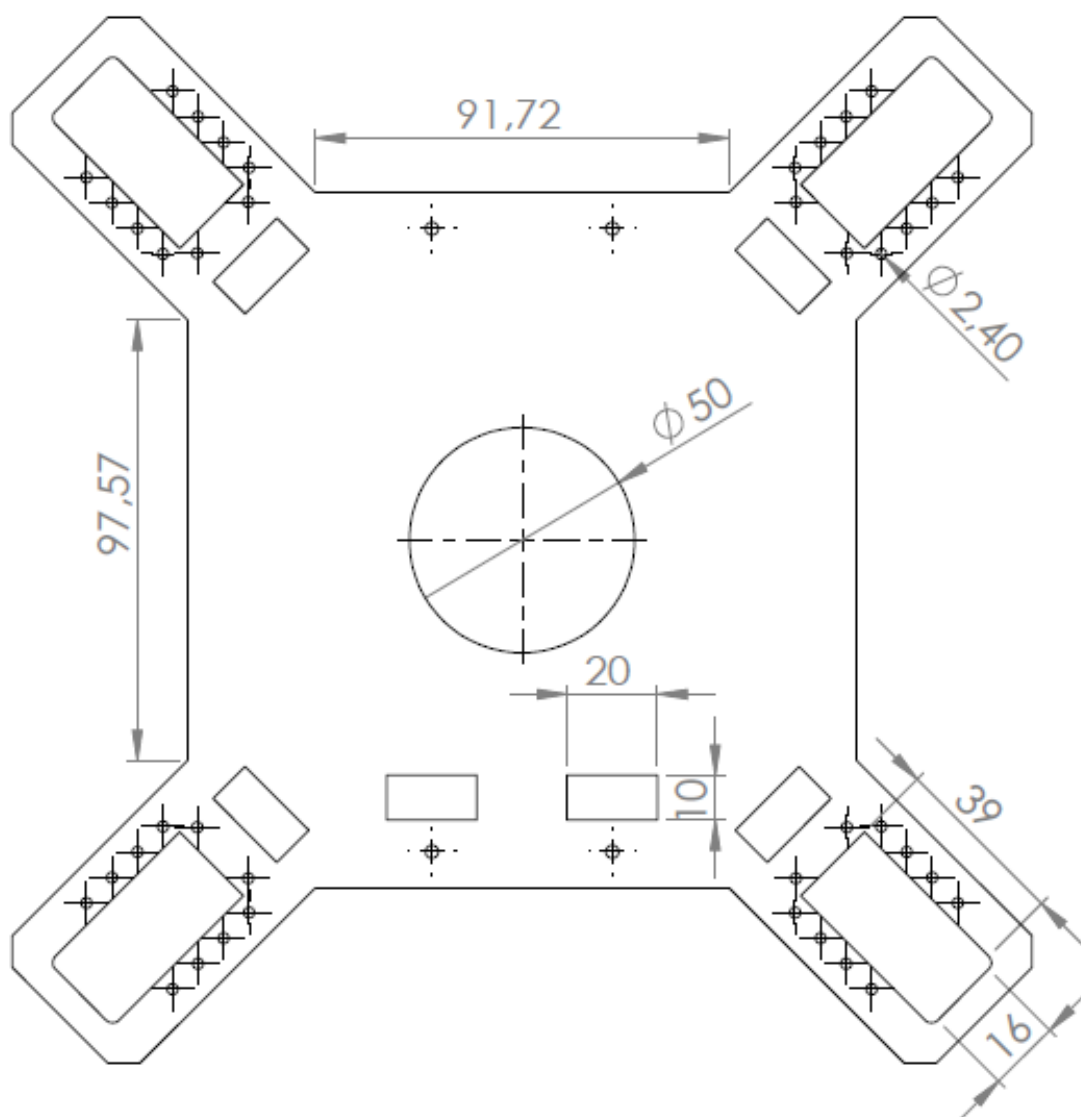


Figura III-2: Tapa inferior de la estructura base.

Fuente: Los autores.

Para darle mayor fijación a las tapas que conforman la estructura base se diseñaron dos placas laterales (figura III-3) que deben ser atornilladas a los costados de la estructura, la altura de la placas es de 32 mm, por lo tanto la estructura base tendrá esa misma altura para alojar los dispositivos y componentes.

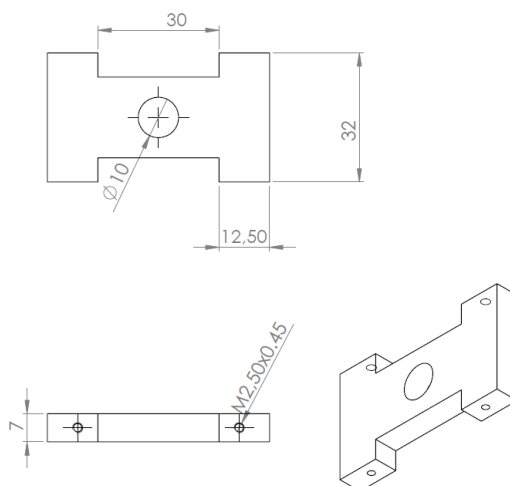


Figura III-3: Placa lateral de la estructura base.

Fuente: Los autores.

Tibia: Es la encargada de conectar la estructura base con el pie del robot, está formada por la unión de dos servomotores, se necesitan cuatro tibias; una para cada pata del robot.

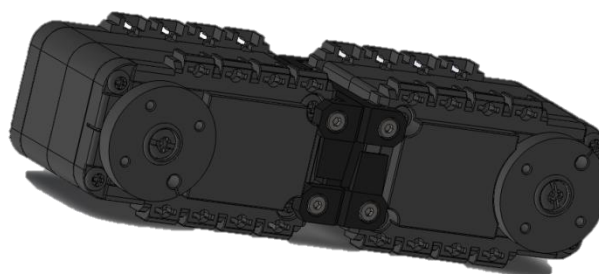


Figura III-4: Tibia conformada por dos servomotores.

Fuente: Los autores.

Pie: Sobre los pies se posará el robot, y está conectado a la tibia, cada pie está formado por dos placas laterales (Figura III-5), la altura total del pie es de 170 mm por lo cual el robot tendrá la capacidad de desplazarse por terrenos con obstáculos relativamente altos, los cuatro orificios circulares ubicados en la parte superior del pie son usados para fijar uno de los servomotores de la tibia por medio de tornillos, mientras que el orificio rectangular de la parte inferior es utilizado para unir las dos placas laterales.

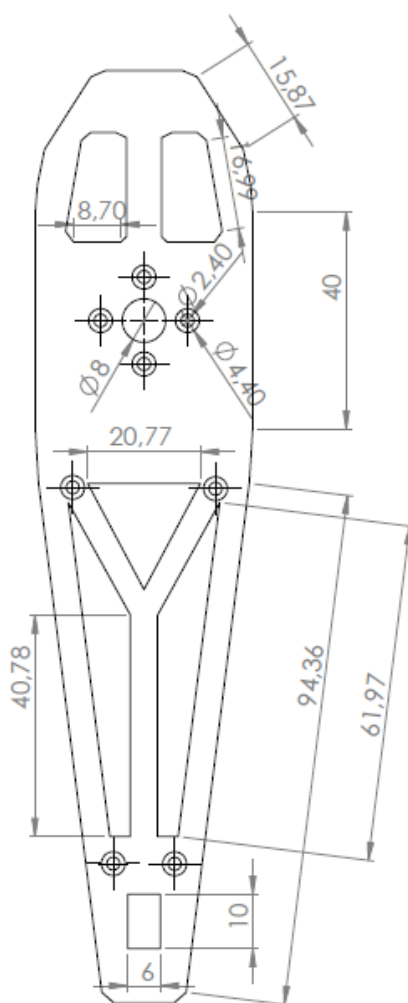


Figura III-5: Placa lateral del pie.

Fuente: Los autores.

Para fijar las placas laterales del pie se diseñaron las uniones inferiores (figura III-6) las cuales van a estar en contacto con el piso y le darán mayor altura al pie del robot y lo mantendrán fijo.

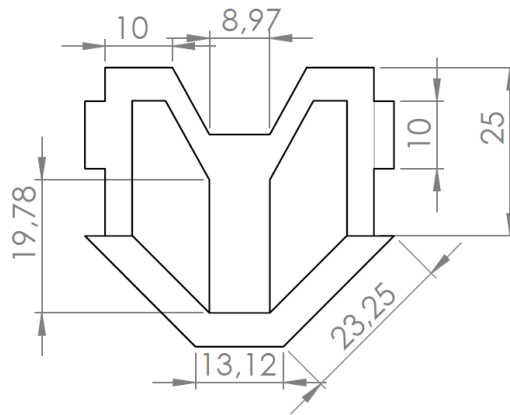


Figura III-6: Unión inferior de las placas laterales del pie.

Fuente: Los autores.

Para que los pies del robot tengan mayor fijación además de la unión inferior se utilizarán tres ejes transversales por cada uno de ellos, la longitud de 40 mm permitirá que el servomotor quepa dentro del pie sin sufrir daños por roce.

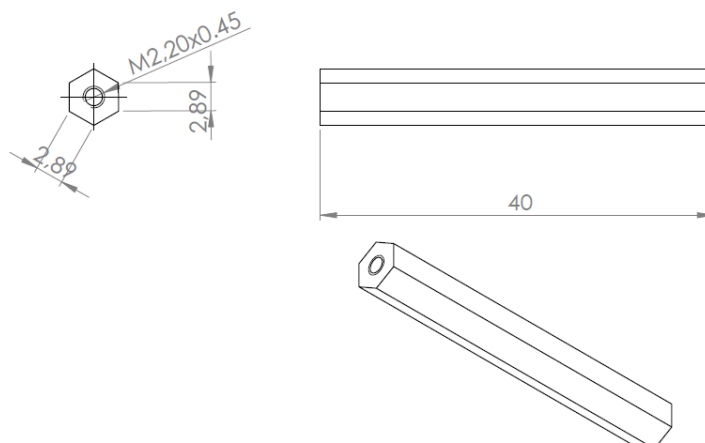


Figura III-7: Ejes para la unión de las placas laterales del pie.

Fuente: Los autores.

Para la unión de los servomotores de la tibia a la estructura base se usaron sockets del kit Bioloid (figura III-8), por medio de éstos se transmite el movimiento hacia la tibia y el pie, las medidas de los sockets están destinadas para el uso con servomotores Dynamixel AX-12A por lo cual no existe problemas de compatibilidad.

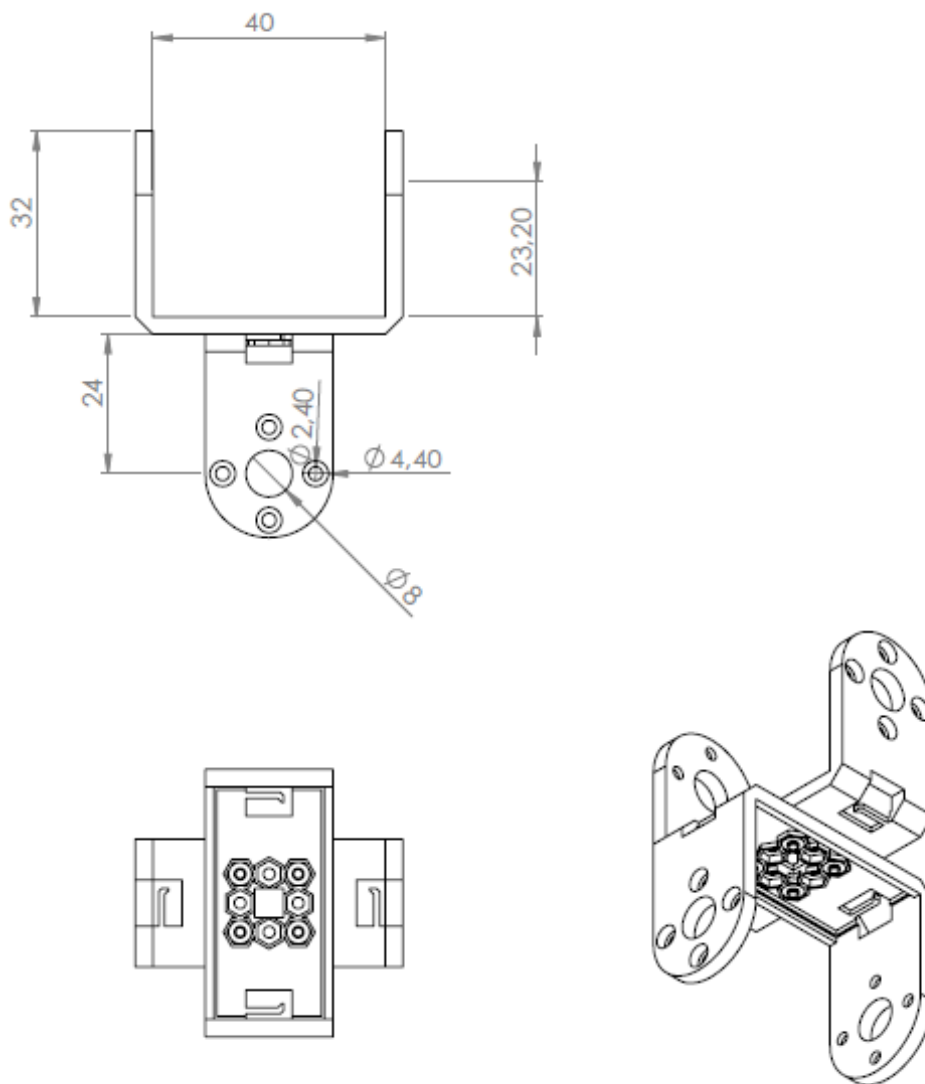


Figura III-8: Sockets del kit Bioloid.

Fuente: Los autores.

Diseño final en SolidWorks: Desarrolladas las partes que componen el robot individualmente se procedió a realizar el ensamblaje final obteniendo como resultado el modelo mostrado en la figura III-9.

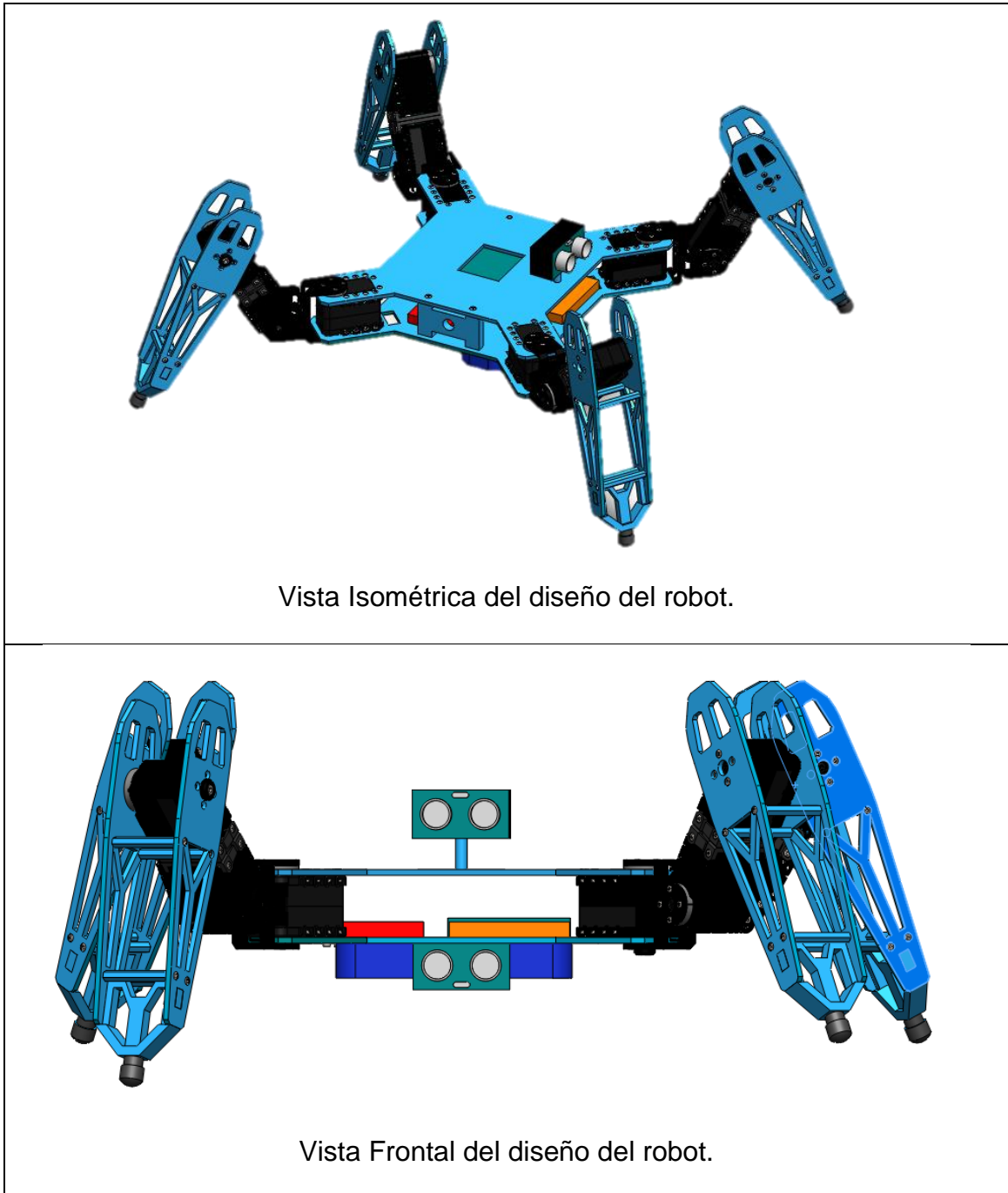


Figura III-9: Ensamblaje completo del robot cuadrúpedo.

Fuente: Los autores.

En la figura III-10 se pueden ver los componentes que forman parte del robot y como están distribuidos.

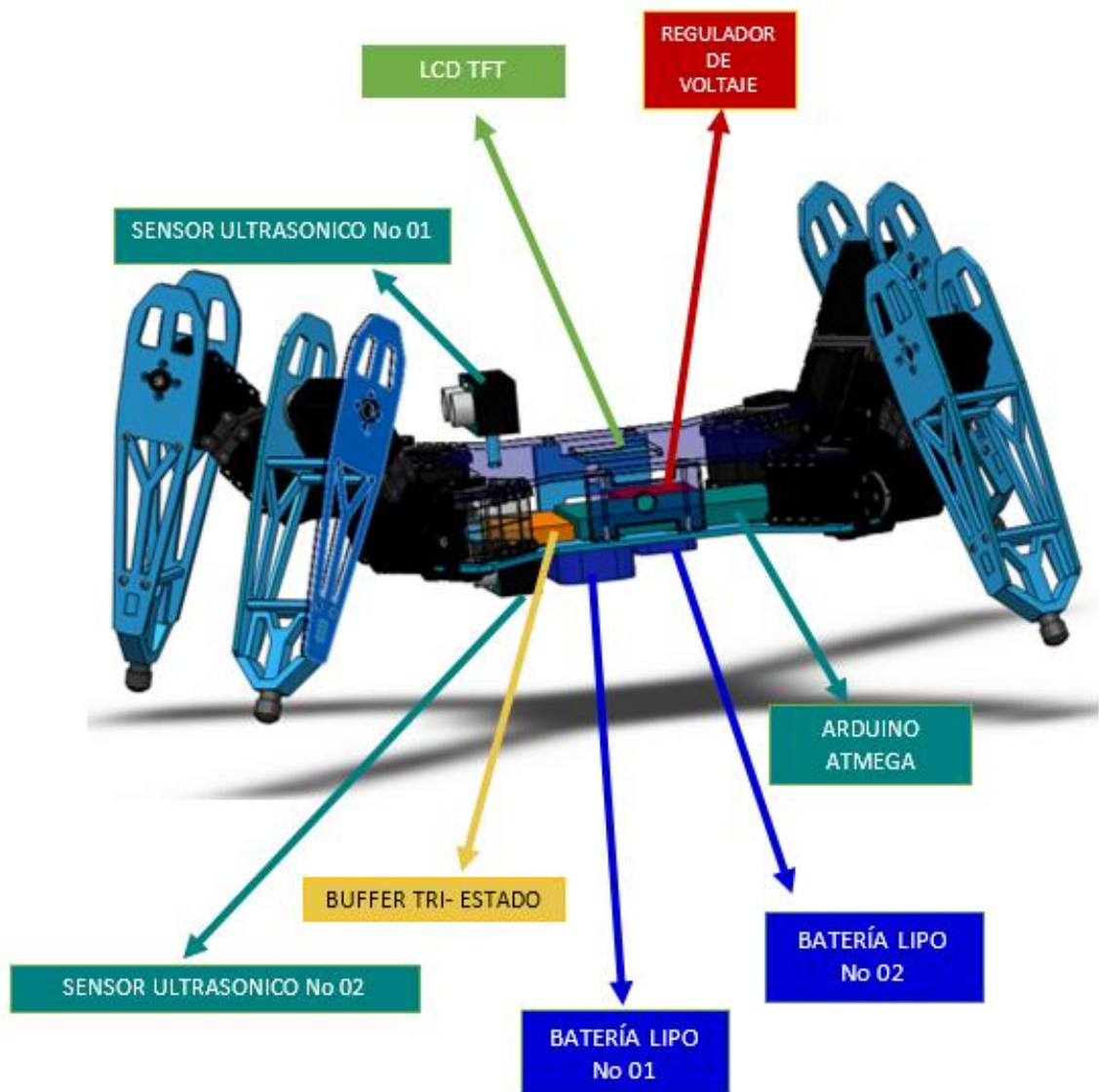


Figura III-10: Distribución de componentes del robot.

Fuente: Los autores.

3.2. CONSTRUCCIÓN DEL ROBOT.

Una vez realizado el diseño final del robot se procedió a construir cada una de las piezas, para mayor precisión y mejor acabado estético se optó por imprimir en 3D en PLA cada una de las partes que conforman la estructura base y los pies, para luego armar la circuitería interna y colocar las baterías y resto de dispositivos según el diseño realizado, obteniendo como resultado el robot que se muestra en la Figura III-11.



Figura III-11: Robot cuadrúpedo terminado de construir.

Fuente: Los autores.

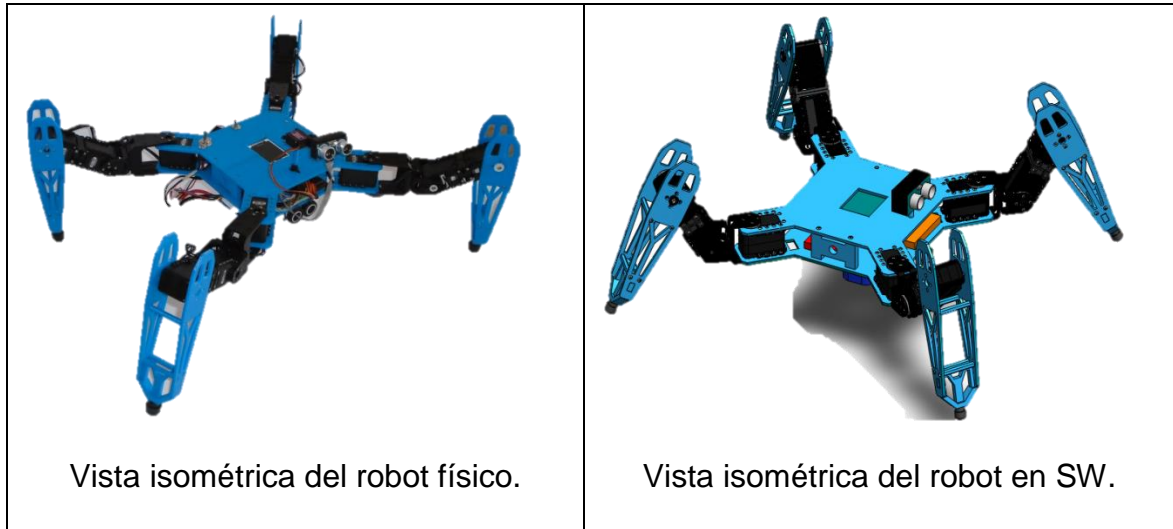


Figura III-12: Vista isométrica del robot cuadrúpedo físico y en SW.

Fuente: Los autores.



Figura III-13: Vista superior del robot cuadrúpedo.

Fuente: Los autores.

CAPÍTULO IV.

4. SIMULACIÓN Y CONTROL DEL ROBOT CUADRÚPEDO.

En este capítulo se muestra el diseño de la simulación 3D del robot cuadrúpedo mediante la herramienta vrbuild2 de Matlab, se explica el funcionamiento y conexión de los dispositivos que componen el robot, además se detalla cada una de las herramientas de la interfaz gráfica de control y monitoreo del robot.

4.1. DISEÑO DE LA SIMULACIÓN DEL ROBOT EN VRBUILD2.

Una vez terminado el diseño y la construcción del robot cuadrúpedo, se creó una simulación de éste usando la caja de herramientas de realidad virtual vrbuild2 de Matlab, la cual permite diseñar mundos virtuales que pueden ser controlados mediante un script desde el mismo Matlab.

Para el diseño en 3D del modelo se debieron definir las relaciones padre e hijo, ejes de rotación, tamaño y centros de las entidades o nodos, en el robot la

entidad padre es el cuerpo, puesto que desde éste nacen cada una de las siguientes partes del dibujo. En la figura IV-1 se muestra el cuerpo del robot, para crear este elemento se debe dar clic en la herramienta *insert box*, seguido en la opción *geometry* del dibujo se insertan las dimensiones para cada uno de sus ejes.

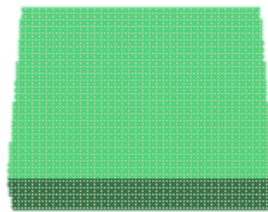


Figura IV-1: Cuerpo del robot cuadrúpedo.

Fuente: Los Autores.

Finalizado el cuerpo se dibujaron las cuatro primeras articulaciones, para insertarlas se debe marcar la opción *children* de la entidad anterior, y se procede a darle clic a la herramienta *insert cylinder*, ya insertado el cilindro se procede en la sección *geometry* a insertar las dimensiones del cubo. Para que se ubiquen las articulaciones en los vértices del cuerpo se debe insertar en la sección *translation* las coordenadas necesarias.

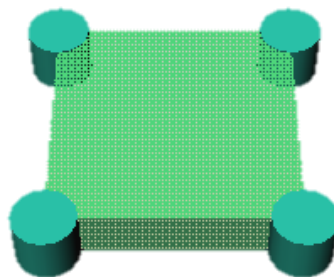


Figura IV-2: Cuerpo y articulaciones del robot cuadrúpedo.

Fuente: Los Autores.

Las articulaciones que se encuentran en el cuerpo se unen a las siguientes por medio del fémur, para agregar esta entidad al dibujo se debe marcar la sección *children* de las articulaciones anteriores y darle clic a la herramienta *insert box*, luego se definen las dimensiones y ejes de rotación. Las articulaciones en el extremo del fémur se agregan de la misma forma como se agregaron las articulaciones del cuerpo, teniendo en cuenta los ejes de rotación, centros y traslación de las entidades.

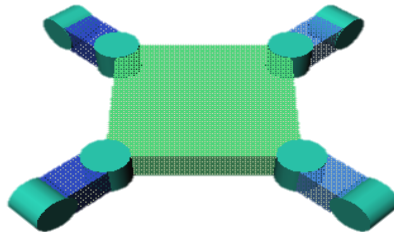


Figura IV-3: Cuerpo, articulaciones y fémur del robot cuadrúpedo.

Fuente: Los Autores.

La tibia es la encargada de unir las articulaciones del fémur a las siguientes y se agregan de la misma forma como se agregaron los fémures, variando en cuanto a dimensiones, centros, traslación.

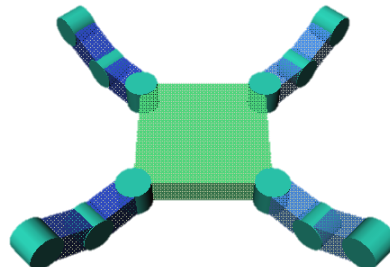


Figura IV-4: Cuerpo, articulaciones, fémur y tibia del robot cuadrúpedo.

Fuente: Los Autores.

Para finalizar el dibujo de la simulación se agregan los pies los cuales siguen el mismo proceso de creación que los fémures y tibias.

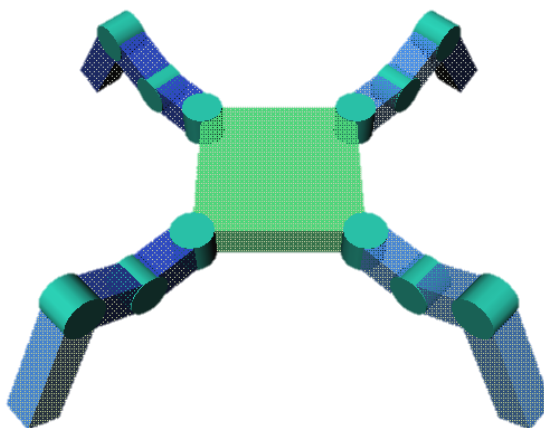


Figura IV-5: Simulación del robot cuadrúpedo terminado.

Fuente: Los Autores.

En la Tabla IV-1 se detallan las relaciones de las entidades, medidas, y ejes de rotación de las entidades según los nombres que se les ha dado en vrbuid2 para la pata número uno.

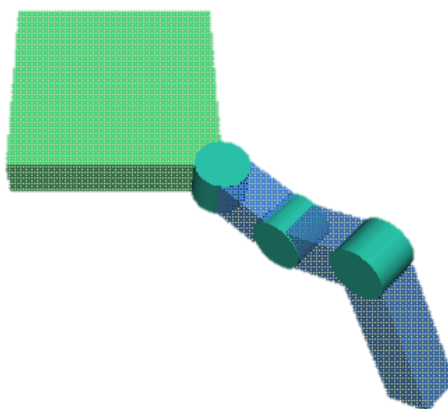


Figura IV-6: Cuerpo y pata número 1 del robot cuadrúpedo.

Fuente: Los Autores.

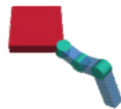
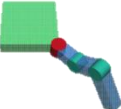
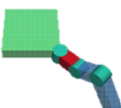
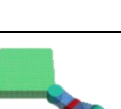
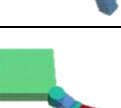

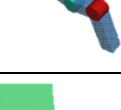
Entidad	Hijo de:	Centro (x,y,z)	Rota- ción (x,y,z)	Medidas (x,y,z) o (r,h)	Traslación (x,y,z)	Dibujo (Rojo)
base1	Bigpuppy	(0,0,0)	(0,0,1)	(0.01,0.01,0.01)	(0,0,0)	-
Base	base1	(0,0,0)	(0,0,1)	(1.9,0.4,1.9)	(0,0,0)	
Articulacion1	Base	(0,0,0)	(0,-1,0)	(0.25,0.42)	(0.95,0,0.95)	
Femur1	Articulacion1	(0,0,0)	(0,0,1)	(0.8,0.4,0.4)	(0.4,0,0)	
Rotor1	Femur1	(0,0,0)	(1,0,0)	(0.25, 0.42)	(0.4,0,0)	
Tibia1	Articulacion1	(-0.4,0,0)	(0,0,1)	(0.85, 0.4,0.4)	(1.225,0,0)	
Rotor2	Tibia1	(0,0,0)	(1,0,0)	(0.25,0.42)	(0.425,0,0)	
Pie1	Tibia1	(- 0.725,0,0)	(0,0,-1)	(1.45,0.4, 0.4)	(1.15,0,0)	

Tabla IV-V: Entidades de la pata 1 de la simulación.

Fuente: Los Autores.

Para las 3 patas restantes las relaciones padre e hijo son las mismas y solo varían las traslaciones para ubicarlas en los diferentes extremos del robot.

4.1.2. CONEXIÓN DE VRBUILD2 CON MATLAB.

Entre matlab y vrbuid2 existe la posibilidad de conexión para crear movimiento en la simulación por medio de una serie de comandos simples, el primer paso es asociar la simulación con matlab, para iniciar se tiene que cargar el mundo virtual; es decir cargar el archivo creado en vrbuid2 con extensión .wrl, para el efecto se creó una variable en donde se almacena el mundo virtual, seguido a esto se lo abre y carga en una ventana de matlab como se muestra en la Figura IV-7. Se puede fijar una serie de parámetros, entre ellos vistas (los cuales deben ser creados desde vrbuid2), zoom y modo de navegación.

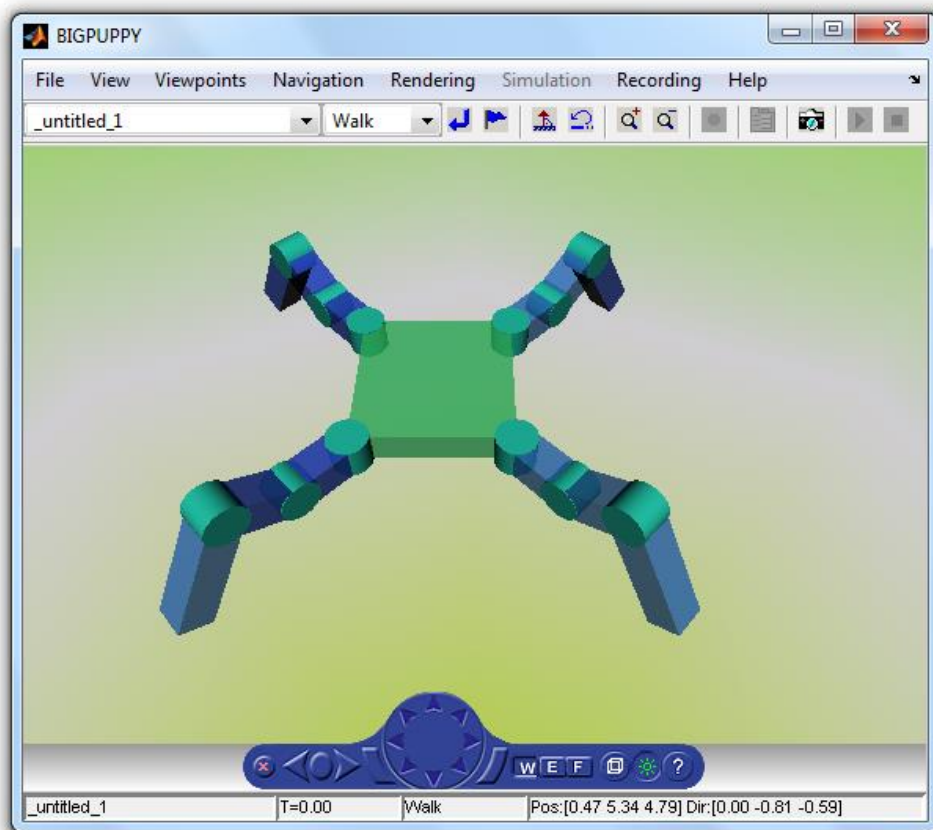


Figura IV-7: Mundo virtual cargado desde matlab.

Fuente: Los Autores.

Cargado el mundo virtual, se puede mover cada una de las articulaciones de la simulación, como punto de partida se conocerá el ángulo al cual se va a mover la articulación, este valor se carga en el nodo (entidad) de la simulación que se elija mover.

Indicar si el movimiento de la articulación es traslacional o rotacional (en el caso del robot cuadrúpedo todas las articulaciones son rotacionales) y sobre que eje rotará ella y sus hijos es el siguiente paso, finalmente actualizar el mundo virtual permite que se muestren los cambios de ángulos de las articulaciones.

4.2. CONTROL DEL ROBOT CUADRÚPEDO.

El control general del robot está desarrollado en Matlab, y como placa de adquisición de datos de los sensores y motores se utilizó un Arduino Mega, la comunicación entre el robot y la PC se realiza inalámbricamente mediante dispositivos Xbee.

Mediante el gráfico de la Figura IV-8 se muestra el funcionamiento, conexiones y comunicaciones del robot cuadrúpedo.

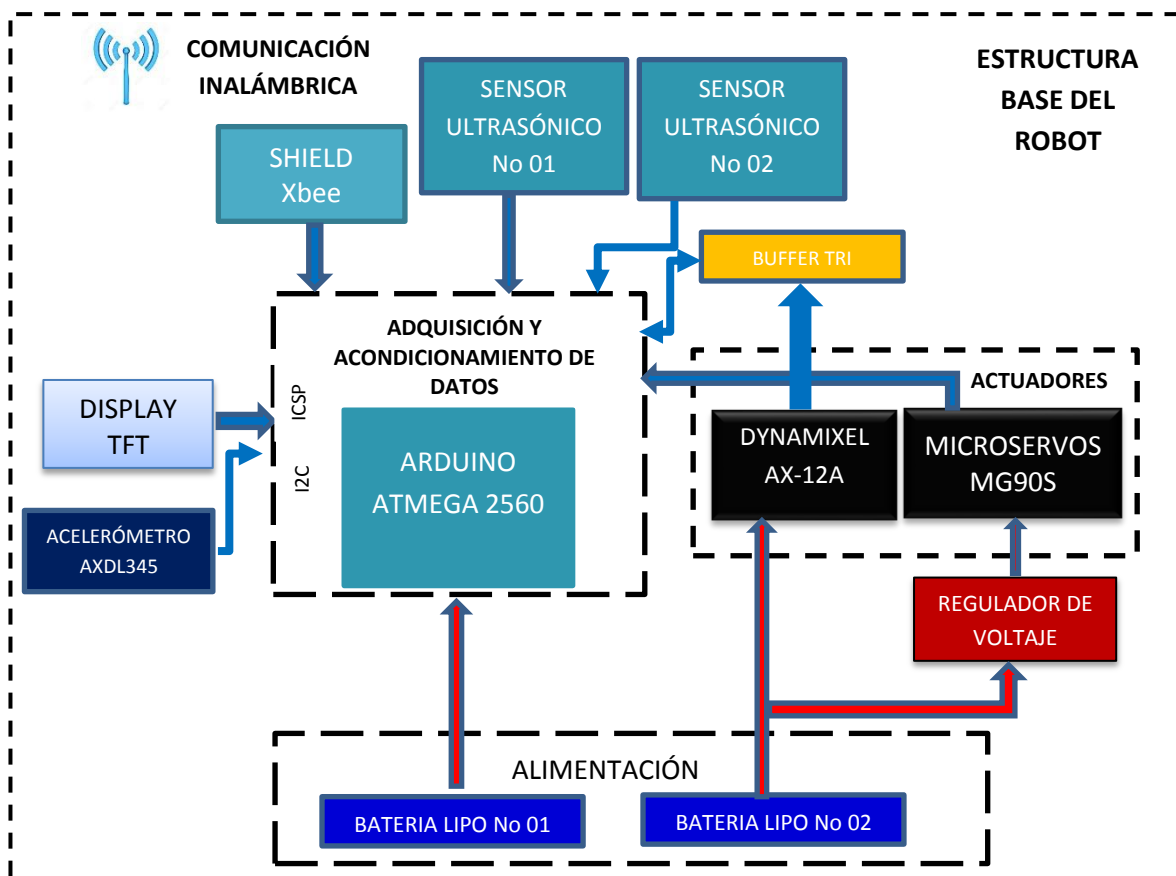
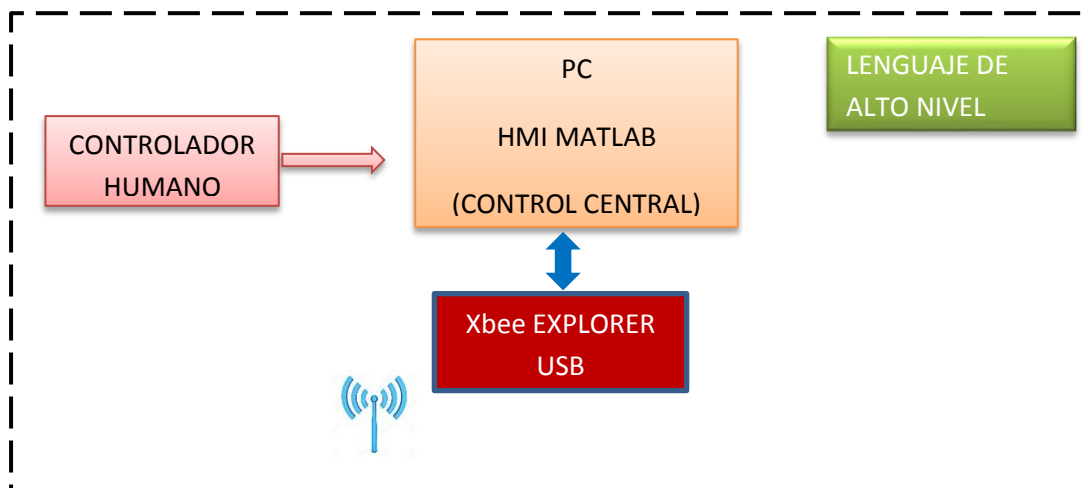


Figura IV-8: Diagrama funcional de la plataforma robótica.

Fuente: Los autores.

4.2.1. FUENTES DE ALIMENTACIÓN.

El robot cuadrúpedo tiene dos baterías LIPO para la alimentación de los diferentes circuitos y actuadores, la batería No. 01 de 7.4 V a 1600 mA suministra energía a los siguientes dispositivos:

- Tarjeta Arduino ATMEGA 2560.
- Display TFT.
- Shield Xbee S1.
- 2 sensores ultrasónicos.
- Acelerómetro AXDL345.
- Buffer Tri estado.

La batería No. 02 de 11.1 V a 3000 mA alimenta a los 12 servomotores Dynamixel AX-12A, y mediante un regulador de voltaje de 5 V energiza los 2 micro servos MG90S que forman la estructura Pan y Tilt.

4.2.2. CONEXIONES DE LOS DISPOSITIVOS.

El centro de conexiones es la tarjeta Arduino MEGA 2560 que está configurada para administrar los diferentes componentes del robot cuadrúpedo y enviar inalámbricamente mediante el módulo Xbee S1 los datos adquiridos de los dos sensores ultrasónicos, acelerómetro digital y datos de temperatura, posición y voltaje de los servomotores Dynamixel AX-12A hacia la PC, donde el software

Matlab procesa la información y envía una respuesta para que los servomotores realicen los movimientos de acuerdo a las condiciones existentes.

Las conexiones de los dispositivos se muestran en la Tabla IV-VI, donde están especificados los dispositivos y las conexiones a los pines de la tarjeta Arduino.

DISPOSITIVO	ARDUINO (PINES)	OBSERVACIÓN	Fuente de Voltaje
Sensor ultrasónico No 01	(30,31)	(ECHO, TRIG)	Batería No. 01
Sensor ultrasónico No 02	(43,45)	(ECHO, TRIG)	Batería No. 01
Display TFT	(50,52,51,10,4,9,8)	(MISO,SCK,MOSI,LCD CS,SD CS,D/C,RESET)	Batería No. 01
Shield Xbee S1	(0,1)	(TX0, RX0)	Batería No. 01
Acelerómetro AXDL345	(20,21)	(SDA,SCL)	Batería No. 01
Buffer Tri Estado	(2,14,15)	(Control, software Serial 3 de librería Dynamixel)	Batería No. 01
Micro servos MG90S N 01	(35,34)	(señal PWM para micro servo 1 y 2 respectivamente)	Batería No. 02

Tabla IV-VI: Tabla de conexiones de dispositivos del robot cuadrúpedo.

Fuente: Los Autores.

Los doce servomotores están conectados en red mediante el buffer triestado; el cual está conectado al tercer puerto UART de la tarjeta Arduino y al pin digital 2 que se encarga de enviar el bit de estado (1 para lectura y 0 para escritura). Desde el buffer triestado se acopla el primer servomotor al cual se conectan en cadena margarita el resto de servomotores como se muestra en la Figura IV-9.

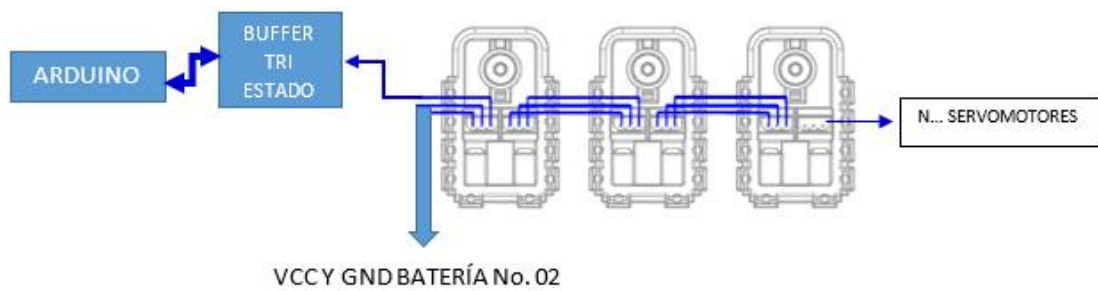


Figura IV-9: Cadena margarita para la conexión de servomotores.

Fuente: Los Autores.

4.2.3. CONEXIÓN MATLAB-ARDUINO.

La conexión entre Matlab y la tarjeta Arduino para el envío y recepción de datos se realiza inalámbricamente mediante los módulos Xbee que están configurados a una velocidad de transmisión de datos de 38400 bps.

Para que exista comunicación entre el dispositivo controlador y el software primero debe abrirse el puerto serial al que está conectado el módulo Xbee en el ordenador, cuando ya se ha abierto el puerto, se puede comenzar a enviar o recibir datos, pero antes se debe definir si los pines de arduino trabajarán como entradas o salidas ya sean analógicas o digitales.

Para realizar la lectura o escritura en cualquiera de los pines de Arduino se envía el comando de Matlab con el tipo de dato, si es de escritura o lectura y la dirección del pin, dentro del controlador Arduino se encuentra corriendo todo el tiempo un conjunto de instrucciones que detectan cuando existe alguna

interrupción, una vez detectada la interrupción que es generada cuando desde Matlab se requiere escribir o leer datos, ésta pasa a otra parte del código en donde se reciben los requerimientos (lectura o escritura) y se realiza la acción.

4.2.4. CONEXIÓN ARDUINO-ACELERÓMETRO AXDL345.

Como se vio anteriormente el acelerómetro es digital por lo tanto su comunicación con arduino se realiza mediante el protocolo I2C, en el código de arduino se debe especificar la librería para la comunicación y la dirección del dispositivo ya que con este protocolo de comunicaciones pueden conectarse en serie más dispositivos.

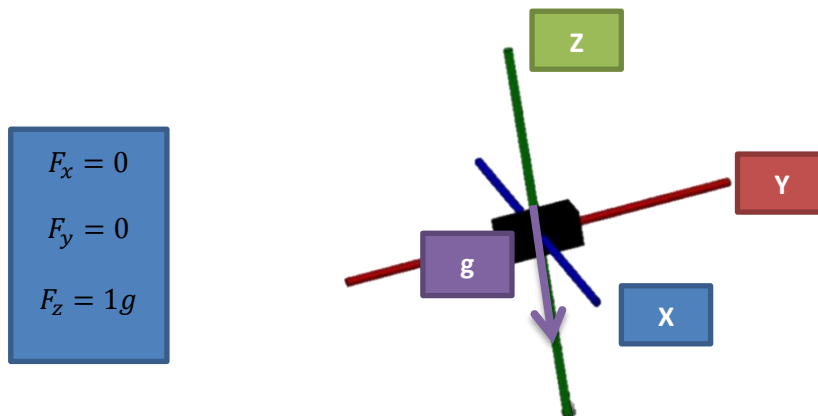


Figura IV-10: Ejes y fuerzas que actúan sobre el acelerómetro.

Fuente: Los Autores.

El acelerómetro en el robot cuadrúpedo está siendo utilizado para medir ángulos de inclinación, para lo cual se leen las aceleraciones de los ejes que

han sido provocadas por una fuerza externa que en este caso es la fuerza de la gravedad, si el dispositivo estuviera estático y en la posición que se muestra en la Figura IV-10 las fuerzas que actúan sobre él serían cero en los ejes X e Y, mientras que en el eje Z la fuerza es igual a la de la gravedad.

Cuando el acelerómetro se inclina no indica su posición pero si las componentes vectoriales de la fuerza de gravedad en los ejes X, Y, Z, sabiendo esto mediante simple trigonometría se pueden obtener los valores de los ángulos de inclinación para los ejes X e Y.

$$\tan\phi = \frac{F_z}{F_x}$$

$$\phi = \text{atan}\left(\frac{F_z}{F_x}\right)$$

Pero como se desea encontrar el ángulo complementario se tiene que:

$$\text{Inclinación}_x = \text{atan}\left(\frac{F_x}{F_z}\right)$$

$$\text{Inclinación}_y = \text{atan}\left(\frac{F_y}{F_z}\right)$$

Una vez obtenidos los datos de los ángulos de inclinación son enviados al programa desarrollado en Matlab, donde son procesados.

4.2.5. CONEXIÓN ARDUINO-PANTALLA.

Cuando existe un bajo nivel de voltaje de la batería de alimentación de los servomotores la pantalla cambia de tonalidad de color advirtiendo al operador del robot que las baterías necesitan recargarse para evitar el daño de alguna de las celdas, el algoritmo lee el voltaje de la batería y lo compara con un valor referencial, si es más bajo que éste quiere decir que la batería necesita carga, por lo tanto el mensaje en la pantalla se pinta de una tonalidad roja, mientras que si los niveles de voltaje están por sobre el valor de voltaje de referencia la tonalidad se mantiene en azul.

4.2.6. CONEXIÓN ARDUINO - SERVOMOTORES DYNAMIXEL AX-12.

Los servomotores y la tarjeta Arduino intercambian información por medio de comunicación serial configurada a una velocidad de transmisión de datos de 1000000 bps, para iniciar la transferencia de datos con los servomotores se debe incluir la librería de comunicación serial para Dynamixel de Arduino y detallar la velocidad a la que se transmitirá los datos (máximo 1 Mbps) y el pin de control para decidir si se lee o escribe en el motor. La librería permite controlar velocidad, ángulo y torque así como también ofrece la opción de leer valores de posición angular, temperatura y voltaje. Cada uno de los motores posee un ID para identificarlo, ya que todos se conectan en serie, el cual puede ser modificado desde arduino ya que por defecto todos vienen con un ID de 1.

4.2.7. CONEXIÓN ARDUINO – XBEE.

El módulo de comunicación inalámbrica Xbee está conectado al puerto serial principal de la tarjeta Arduino y su única función es la de enviar y recibir inalámbricamente los datos que deberían pasar por el puerto serial, es decir; una vez conectado y configurado el módulo Xbee (velocidad de transmisión de 38400 bps y dirección del otro dispositivo Xbee al cual se conectará) funcionará como un cable de datos serial (conexión punto a punto), por lo tanto no se debe agregar código adicional en Arduino o el software de control del robot del ordenador.

4.3. INTERFAZ DE CONTROL Y MONITOREO DEL ROBOT.

Para controlar y monitorear el robot cuadrúpedo se desarrolló una interfaz gráfica con la herramienta Guide de Matlab, los datos provenientes del robot (sensores ultrasónicos, acelerómetro y servomotores Dynamixel AX-12A) son leídos y mostrados al operador, y de acuerdo a éstos las acciones y respuestas son generadas.

En la Figura IV-11 se pueden ver las herramientas y componentes que forman parte de la interfaz gráfica de usuario.

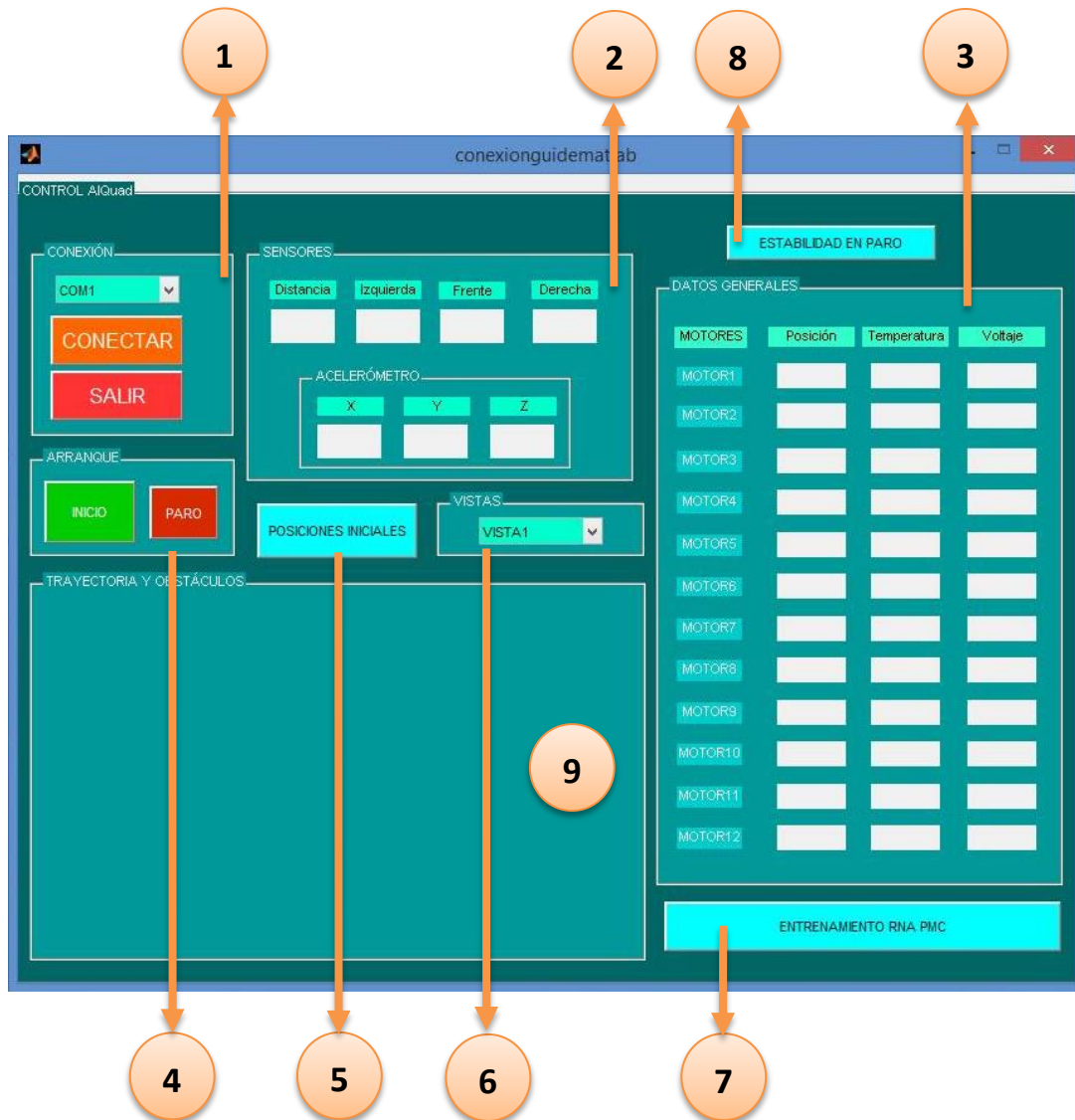


Figura IV-11: Interfaz de control y monitoreo del robot.

Fuente: Los Autores.

A continuación se detalla cada uno de los paneles y botones que conforman la interfaz gráfica de control y monitoreo del robot cuadrúpedo, de acuerdo a la numeración mostrada en la figura IV-11.

1. Panel de conexión: En este panel existe un listbox del cual se debe seleccionar en qué puerto serial del ordenador está conectado el módulo Xbee Explorer USB para comunicarse con el robot, el botón CONECTAR es el que inicia la comunicación por lo que se lo debe pinchar una vez que ya haya sido escogido el puerto serial correcto. El botón SALIR cierra las ventanas de la interfaz y la simulación.

2. Panel de sensores: En este panel el primer cuadro de texto muestra la distancia que existe entre el robot y objetos ubicados al frente y es medida por el sensor ultrasónico que permanece estático, mientras que los siguientes 3 cuadros de texto muestran las distancias hacia el frente y costados que han sido medidas por el sensor ultrasónico que se encuentra montado sobre el soporte pan y tilt, por último los tres cuadros de texto de la parte inferior del panel muestran las inclinaciones a las que es sometido el robot cuadrúpedo en los tres ejes del plano (X,Y,Z).

3. Panel de datos de los servomotores: Se muestran los datos de los servomotores para que quien opere el robot conozca si éstos se encuentran dentro de los rangos permisibles de voltaje y temperatura y además sepa la posición exacta de cada uno de ellos.

4. Panel de arranque: En esta parte de la interfaz se encuentran dos botones; el botón INICIO que hace que el sensor ultrasónico montado sobre la estructura pan y tilt mapee el frente del robot para detectar obstáculos y luego inicie su

caminata, y el botón PARO hace que se detenga totalmente sin finalizar la aplicación.

5. Botón de posiciones iniciales: Este botón hace que cada servomotor se posicione en el ángulo inicial ideal para comenzar la caminata.

6. Panel de vistas: En este panel solo se encuentra un listbox, del cual se puede escoger una de las tres vistas o viewpoints creados para mejorar la visualización de la simulación del robot cuadrúpedo.

7. Botón de entrenamiento de la RNA: Al pinchar este botón inicia el entrenamiento de la red neuronal artificial en base a los vectores de entrada y salida preestablecidos.

8. Botón de estabilidad en paro: Cuando el robot está en sus posiciones iniciales pero no caminando, al presionar el botón los motores se posicionaran tratando de buscar la estabilidad del robot.

9. Panel de trayectorias y obstáculos: Mediante un axes (cuadro para dibujo) se muestra una simulación de las ráfagas de ultrasonido enviadas y reflejadas por algún objeto del sensor montado en el soporte pan y tilt, ayudando a saber que trayectoria debería tomar el robot.

4.4. RED NEURONAL ARTIFICIAL PARA GENERACIÓN DE LOCOMOCIÓN.

La locomoción del robot cuadrúpedo está basada en la RNA Perceptrón Multicapa, la cual tiene un aprendizaje supervisado por lo que se debe presentar a la red los datos de entrada y el vector de salidas. Los valores del vector de entrada están dados por las mediciones del acelerómetro de las inclinaciones del robot en los ejes X e Y, mientras que los datos del vector de salida están dados por los ángulos en los que los motores deben posicionarse para lograr la estabilidad.

Luego de obtener los vectores de entrada y salida se diseñó la RNA, la cual posee dos capas ocultas, la primera cuenta con ocho neuronas, la segunda con doce y la capa de salida posee ocho neuronas, a continuación se muestra parte del código utilizado para crear la red y entrenarla.

```
net=newff([0 360; 0 360],[8 12 8],{'purelin','purelin','purelin'})
```

net = train(net,A,B); donde A es el vector de entrada y B el de salida.

CAPÍTULO V.

5. COSTOS, PRUEBAS Y RESULTADOS.

En este capítulo se encuentran los costos incurridos para la realización de este proyecto, las pruebas a las cuales fue sometido el robot, el análisis estadístico de los datos obtenidos durante la fase de pruebas y la demostración de la hipótesis por medio de los resultados.

5.1. COSTO DE LOS MATERIALES.

Para la construcción del robot cuadrúpedo se incurrió en gastos de dispositivos electrónicos, herramientas, accesorios y servicios de impresión 3D, los cuales se indican en la tabla V-VII.

Tabla V-VII: Listado de precios.

LISTA DE PRECIOS ROBOT CUADRÚPEDO			
MATERIALES	CANT.	P.U.	TOTAL
Servomotores Dynamixel Ax-12A.	12	65	780
Impresión 3D de la estructura	1	100	100

Pantalla TFT	1	35	35
Xbee S1	2	40	80
Microservo MG90S	2	13	26
Acelerómetro Adxl 345	1	15	15
Batería LIPO No 01	1	45	45
Batería LIPO No 02	1	20	20
Arduino Mega 2560	1	50	50
Programador Xbee USB	1	18	18
Xbee arduino shield	1	18	18
Sensor Ultrasónico Hc-sr04	2	12	24
USB2Dynamixel	1	50	50
Gastos Varios	1	200	200
TOTAL			1461

Tabla V-VII: Listado de precios (continuación).

Fuente: Los Autores.

5.2. PRUEBAS REALIZADAS.

Durante la fase de pruebas de la locomoción del robot cuadrúpedo se obtuvieron los datos para el análisis estadístico, las pruebas consistían en aplicarle obstáculos de diversos tamaños a la superficie sobre la que se desplazaba el robot y mediante el acelerómetro tomar las mediciones de los ángulos de inclinación en los ejes X e Y luego de que el algoritmo basado en redes neuronales artificiales trate de estabilizarlo.

5.3. ANÁLISIS ESTADÍSTICO.

Para la comprobación de hipótesis se compararon los datos de inclinación del robot en los ejes X e Y con la inclinación ideal que debería tener el robot para mantenerse estable, en este caso; 0° en el eje X y 0° en el eje Y.

Las medidas del acelerómetro para treinta condiciones de terreno diferentes fueron las siguientes:

Tabla V-VIII: Inclinaciones medidas e ideales del robot cuadrúpedo.

DATOS DE ACELERÓMETRO EN GRADOS		INCLINACIONES IDEALES EN GRADOS	
X	Y	X	Y
1	1	0	0
2	2	0	0
2	2	0	0
0	-3	0	0
4	1	0	0
-2	-1	0	0
-2	0	0	0
3	-1	0	0
-2	-1	0	0
-1	0	0	0
-2	-1	0	0
0	-1	0	0
0	0	0	0
2	-2	0	0
1	1	0	0
-1	0	0	0
1	1	0	0
0	-3	0	0
4	1	0	0
1	1	0	0
1	0	0	0
-2	0	0	0

3	-1	0	0
3	2	0	0
1	0	0	0
2	-2	0	0
0	-1	0	0
-2	-1	0	0
3	2	0	0
0	0	0	0

**Tabla V-VIII: Inclinaciones medidas e ideales del robot cuadrúpedo
(continuación).**

Fuente: Los Autores

Se utilizó la distribución t de Student para determinar si la media de los datos medidos tienen o no una diferencia significativa con respecto a la media de los datos ideales, pero primero se debe comprobar que los datos estén distribuidos normalmente y para ello se aplicó el test de Shapiro-Wilk, para facilitar los cálculos se utilizó el software estadístico R obteniendo para el test de normalidad los siguientes resultados:

Para los datos del eje X:

$W=0.9294$

A un nivel de confianza del 95% y con una muestra $n=30$ se tiene un valor crítico de 0.927. Por lo tanto como el valor del estadístico W es mayor al valor crítico se puede concluir que los datos medidos del eje X siguen una distribución normal.

Para los datos del eje Y:

$$W=0.9342$$

A un nivel de confianza del 95% y con una muestra $n=30$ se tiene un valor crítico de 0.927. Por lo tanto como el valor del estadístico W es mayor al valor crítico se puede concluir que los datos medidos del eje Y siguen una distribución normal.

Una vez determinado que los datos de inclinaciones de los ejes X y Y siguen una distribución normal, se aplicó el test t de Student para la comprobación de la hipótesis la cual es:

H_0 : La locomoción de un robot cuadrúpedo basada en la Red Neuronal Artificial Perceptrón Multicapa permite que el robot sea estable tanto en terrenos regulares, irregulares y con pequeños obstáculos.

En caso de no comprobarse la hipótesis nula planteada se tiene la alternativa:

H_1 : La locomoción de un robot cuadrúpedo basada en la Red Neuronal Artificial Perceptrón Multicapa no permite que el robot sea estable tanto en terrenos regulares, irregulares y con pequeños obstáculos.

Para el eje X se comparó la media de los datos obtenidos con la media ideal para saber si existe una diferencia significativa entre las dos y saber si el robot llega a ser estable.

Aplicando la test t de Student se obtuvo:

$$t=1.9407$$

A un nivel de confianza del 95% y con 29 grados de libertad el valor crítico es de 2.045. Debido a que el estadístico t es menor al valor crítico se acepta la hipótesis nula, con los datos del eje Y debe suceder lo mismo así que se aplica el mismo test obteniendo:

$$t=-0.5281$$

A un nivel de confianza del 95% y con 29 grados de libertad el valor crítico es de 2.045. Por lo tanto también se acepta la hipótesis nula para el caso de los datos del eje Y.

5.4. RESULTADOS.

Al analizar los datos se obtuvo que el robot cuadrúpedo al tener su locomoción basada en redes neuronales artificiales no muestra una variación significativa de inclinaciones comparadas con la media ideal, por lo cual mantiene la estabilidad a pesar de irregularidades y obstáculos presentes en el terreno sobre el cual se desplaza.

CONCLUSIONES.

- 1.** El uso de herramientas CAD en conjunto con la técnica de impresión 3D, permitieron crear las partes y piezas del robot con gran precisión y un buen acabado estético, brindando seguridad y robustez a la plataforma robótica.
- 2.** Los diferentes protocolos de comunicación soportados por los dispositivos (I2C, SPI) y actuadores del robot (Serial Half Duplex), reducen la complejidad de conexión de éstos a la tarjeta controladora Arduino Mega 2560, la cual posee integrados los puertos de comunicación necesarios para conectarlos.
- 3.** La buena capacidad de generalización, tolerancia a fallos y flexibilidad de las RNA, las convierten en una técnica ideal para el control de robots caminante, debido a que se puede resolver de manera coordinada y estable el problema de la locomoción.
- 4.** Con la simulación del robot se logró tener una plataforma virtual para el estudio de locomoción, sin exponer el modelo físico a daños causados por posibles fallos durante las fases de desarrollo y pruebas de caminata, permitiendo así minimizar el riesgo de daños de la plataforma física.
- 5.** La correcta distribución de los dispositivos y componentes, así como también la ubicación inicial de las patas del robot son esenciales para que la caminata

sea eficiente, ya que podrían causar que el centro de gravedad esté muy alejado del centro del robot provocando su caída.

6. La estabilidad de un robot cuadrúpedo depende de varios factores, entre ellos la distribución de masas, diseño estructural, centro de gravedad y el tipo de locomoción, siendo el último el de mayor importancia debido a que si existen deficiencias en los factores anteriores se pueden corregir mediante éste.

7. La locomoción del robot cuadrúpedo basada en redes neuronales artificiales, presenta durante su caminata sobre superficies irregulares y con obstáculos una media de inclinaciones de $0,67^\circ$ para el eje X y de $-0,13^\circ$ para el eje Y, inclinaciones muy cercanas a las ideales que corresponden a 0° para el eje X y 0° para el eje Y.

8. Mediante el uso de la RNA y gracias a su buena generalización se logró obtener inclinaciones muy cercanas a las ideales, ya que los datos de entrada (inclinaciones del acelerómetro en los ejes X e Y) y salida (ángulos de los motores para llegar a la estabilidad) durante la etapa de entrenamiento fueron significativos y representaban una gran cantidad de situaciones que el robot podría enfrentar durante su caminata.

RECOMENDACIONES.

1. Que las conexiones sean las correctas, las condiciones de trabajo sean las adecuadas, que el voltaje y corriente que reciben los dispositivos del robot sean los ideales, evitará el daño de éstos, por lo tanto; es necesario revisar la hoja de características (datasheet) con el objetivo de prolongar la vida útil y proteger la integridad de los dispositivos.

2. Separar los circuitos de potencia y control es de suma importancia debido a los requerimientos de voltaje de los diferentes dispositivos, así como también para evitar daños provocados por cortocircuitos, sobretensiones o picos de corriente.

3. Utilizar un software de diseño CAD brinda grandes ventajas al diseñador ya que posee una serie de herramientas que permiten crear proyectos acordes a las necesidades del diseño y ofrece la posibilidad de extraer datos técnicos y optimizar el tiempo de modelado.

4. Debido a que el robot está destinado al estudio de locomoción, está en movimiento constante, lo que podría causar que los tornillos se desajusten, por lo que es vital realizar periódicamente una revisión de la estructura con el objetivo de que los dispositivos y partes que componen el robot estén fijados correctamente.

5. Usar Redes Neuronales Artificiales permite tener un mejor control de plataformas robóticas, por lo cual su aplicación permite obtener una salida muy próxima a la deseada.

RESUMEN.

Se diseñó y construyó un robot cuadrúpedo con locomoción basada en la red neuronal artificial perceptrón multicapa, para conocer si logra mantener la estabilidad durante la caminata sobre terrenos irregulares. Se utilizó el método inductivo para el desarrollo del diseño estructural, la circuitería interna, el algoritmo de control y la interfaz de monitoreo. La aplicación del método experimental permitió verificar el funcionamiento del robot. Las piezas que componen el cuerpo y las patas del robot fueron construidas mediante la técnica de impresión 3D con plástico PLA. El algoritmo de control para la locomoción del robot y la interfaz de monitoreo de datos de: sensores ultrasónicos, acelerómetro y velocidad, temperatura y voltaje de cada uno de los motores fueron desarrollados en MATLAB R2010a. La comunicación se realiza inalámbricamente entre la placa microcontroladora Arduino del robot y el ordenador mediante dispositivos Xbee y como fuente de alimentación se utilizaron dos baterías Li-Po. Mediante el análisis de los datos se comprobó que el robot cuadrúpedo no muestra una variación significativa de inclinaciones comparadas con las ideales, por lo tanto mantiene la estabilidad a pesar de irregularidades y obstáculos presentes en el terreno sobre el cual se desplaza. El robot cuadrúpedo podría utilizarse en procesos industriales de transporte de materiales o sustancias peligrosas y exploración de ambientes con terrenos irregulares, de difícil acceso o contaminados.

Palabras clave: ROBÓTICA/ INTELIGENCIA ARTIFICIAL/ REDES NEURONALES ARTIFICIALES/ ROBOT CUADRÚPEDO/ ALGORITMOS DE LOCOMOCIÓN.

ABSTRACT.

A quadruped robot with locomotion was designed and build based on the artificial neural network perceptron multilayer, to find out if it can keep the stability while walking on uneven terrain. The inductive method was used for the structural design, the internal circuitry, the control algorithm and interface monitoring. The experimental method application allowed verifying the operation of the robot. The component parts of the body and legs of the robot were constructed by 3D printing technique with PLA (Poly Lactic Acid) plastic. The control algorithm for the robot locomotion and monitoring interface data of: ultrasonic sensors, accelerometer and speed, temperature and voltage of each one of the servomotors were developed in MATLAB R2010a. Communication in done wirelessly between the Arduino motherboard plaque and the computer through Xbee devices and as a main power supply two batteries Li-Po were used. By analyzing the data it was found that the quadruped robot shows no significant variation compared with the ideal inclinations, therefore maintains stability despite irregularities and obstacles in the terrain over which it travels. The quadruped robot could be used in industrial processes transporting hazardous materials or substances and exploration of irregular terrain, inaccessible or contaminated.

Keywords: ROBOTICS/ ARTIFICIAL INTELLIGENCE/ NEURONAL NETWORKS/ QUADRUPED ROBOT/ LOCOMOTION ALGORITHMS.

BIBLIOGRAFÍA.

- 1. ABUSSAID, S.**, Diseño y control reactivo de robots caminantes sobre terreno natural., Tesis Doctoral., Madrid-España., Universidad Complutense de Madrid., Facultad de Ciencias Físicas., 2009., pp. 1-22.
- 2. BASTARDO, M.**, Diseño y construcción de un prototipo de robot cuadrúpedo., Tesis Ing. Mec., Caracas-Venezuela., Universidad Simón Bolívar., Decanato de Estudios Profesionales Coordinación de Ingeniería Mecánica., 2009., pp. 9-22.
- 3. BLUM, J.**, Exploring Arduino: tools and techniques for engineering wizardry., Primera ed., Nueva York: Jhon Wiley & Sons., 2013., 384 p.
- 4. ESCOLANO, F. et al.**, Inteligencia artificial: modelos, técnicas y áreas de aplicación., Primera ed., Madrid-España: Paraninfo., 2003., pp. 91-100.

5. **FLÓREZ, R. y FERNÁNDEZ, J.**, Las redes neuronales artificiales: fundamentos teóricos y aplicaciones prácticas., Primera ed., La Coruña-España: NETBIBLO S. L., 2008., pp. 60-73.

6. **GIRÓN, D.**, Diseño y simulación de un robot cuadrúpedo en la herramienta Solid Edge para implementarlo en máquinas de acompañamiento a personas discapacitadas., Tesis Ing. en Sistemas y Telecomunicaciones., Pereira-Colombia., Universidad Católica de Pereira., Facultad de Ciencias Básicas e Ingeniería., 2013., pp. 7-40.

7. **KHALED, M.**, Virtual reality and animation for MATLAB and simulink users: visualization of dynamic models and control simulations., Primera ed., Nueva York: Springer-Verlag., 2012., pp. 1-28.

8. **MAZUREK, B.**, Mecánica vectorial para ingenieros: estática., 9a. ed., México, D. F.: McGraw-Hill., 2010., pp. 259-260.

9. **PAREDES, W.**, Cómo desarrollar una tesis. Primera ed., Riobamba-Ecuador: BIGcopy: 2009., 201 p.

10. **PIEDRA, J.**, Aplicación de los sistemas neurodifusos a la interpretación automática de imágenes de satélite., Tesis Doctoral., Almería-España., Universidad de Almería., Departamento de Lenguajes y Computación., 2005., pp. 317-333.

11. RICO, E., Analysis and application of methods of displacement in articulated platforms based on CPG's (Central Pattern Generator)., Tesis Maestría en Ing. Área Automatización Industrial., Medellín-Colombia., Universidad Nacional., Facultad de Ingeniería., 2012., pp. 34-46.

12. SEDEÑO, E., Locomoción de un robot cuadrúpedo: un enfoque a celdas neuronales analógicas., Tesis Maestría., Cuernavaca-México., Centro Nacional de Investigación y Desarrollo Tecnológico., Coordinación de Mecatrónica., 2011., pp. 29-37.

13. ACELERÓMETRO DIGITAL.

< http://5hertz.com/index.php?main_page=product_info&products_id=409>

[Consulta: 20 Mayo 2014].

<<http://fuenteabierta.teubi.co/2013/03/inclinometro-digital-con-arduino-uso-de.html>>

[Consulta: 20 Mayo 2014].

14. ARDUINO MEGA 2560.

<<http://arduino.cc/en/Main/arduinoBoardMega2560>>.

[Consulta: 04 Junio 2014].

15. BUFFER TRI-ESTADO.

<<http://electronica-teoriaypractica.com/category/salida-triestado/>>

[Consulta: 11 Junio 2014].

<<http://es.scribd.com/doc/36601026/Salidas-Logicas-Triestados>>

[Consulta: 11 Junio 2014].

16. LENGUAJE DE PROGRAMACIÓN ARDUINO.

<http://www.uco.es/aulassoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf>

[Consulta: 10 Julio 2014].

17. MICRO SERVOMOTOR MG90S.

<<http://electronilab.co/tienda/micro-servo-9g-towerpro/>>

[Consulta: 16 Julio 2014].

<<http://www.info-ab.uclm.es/labelec/solar/electronica/elementos/servomotor.htm>>

[Consulta: 16 Julio 2014].

18. PANTALLA TFT ARDUINO.

< <http://arduino.cc/en/Guide/TFT>>.

[Consulta: 22 Mayo 2014].

< <http://arduino.cc/en/Reference/TFTLibrary>>

[Consulta: 22 Mayo 2014].

19. PROTOCOLO DE COMUNICACIÓN ENTRE ARDUINO Y SERVOMOTORES AX-12A CON BUFFER TRI-ESTADO.

<<http://www.generationrobots.com/media/Dynamixel-AX-12-user-manual.pdf>>

[Consulta: 09 Junio 2014].

20. ROBOT.

<http://estefaniaospina774p.blogspot.com/2012_10_01_archive.html>

[Consulta: 14 Mayo 2014].

<http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.1.htm>

[Consulta: 14 Mayo 2014].

21. ROBOT CUADRÚPEDO.

<http://www.unicauca.edu.co/ai/publicaciones/Aguilar_ACA09.pdf>

[Consulta: 26 Mayo 2014].

22. ROBOT MÓVIL.

<http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.5.htm>

[Consulta: 14 Mayo 2014].

23. SENSOR ULTRASÓNICO.

< https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit>

[Consulta: 08 Junio 2014].

24. SERVOMOTOR DYNAMIXEL AX-12A.

<<http://ro-botica.com/es/Producto/Actuador-Dynamixel-AX-12A/>>

[Consulta: 2 Junio 2014].

<<http://robbottini.altervista.org/dynamixel-ax-12a-and-arduino-how-to-use-the-serial-port>>

[Consulta: 2 Junio 2014].

<<http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx>>

[Consulta: 2 Junio 2014].

<<http://www.generationrobots.com/media/Dynamixel-AX-12-user-manual.pdf>>

[Consulta: 2 Junio 2014].

25. XBEE EXPLORER USB.

<<https://www.sparkfun.com/products/11812>>

[Consulta: 18 Julio 2014].

26. XBEE S1.

<<http://www.xbee.cl/>>

[Consulta: 18 Julio 2014].

<<http://www.andresduarte.com/arduino-y-xbee>>

[Consulta: 18 Julio 2014].