



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y  
MECÁNICA**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
MECATRÓNICA**

**AUTOR: TABANGO OBACO, RICARDO DANIEL**

**TEMA: DESARROLLO DE UN SISTEMA DE ROBÓTICA COLECTIVA  
CON PROCESAMIENTO CENTRALIZADO ENTRE DOS ROBOTS  
HUMANOIDES BIOLOID PREMIUM**

**DIRECTOR: ING. CHACÓN, ALEJANDRO**

**CODIRECTOR: ING. TAPIA, MELTON**

**SANGOLQUÍ, ENERO 2014**

# **CERTIFICACIÓN DE LA ELABORACIÓN DEL PROYECTO**

El proyecto “**DESARROLLO DE UN SISTEMA DE ROBÓTICA COLECTIVA CON PROCESAMIENTO CENTRALIZADO ENTRE DOS ROBOTS HUMANOIDES BIOLOID PREMIUM**” fue realizado en su totalidad por Ricardo Daniel Tabango Obaco, como requerimiento parcial para la obtención del título de Ingeniero en Mecatrónica.

**Ing. Alejandro Chacón**  
**DIRECTOR**

**Ing. Melton Tapia**  
**CODIRECTOR**

**Sangolquí, enero de 2014**

## **AUTORÍA DE RESPONSABILIDAD**

Autor: Ricardo Daniel Tabango Obaco

DECLARA QUE:

El proyecto de grado: “DESARROLLO DE UN SISTEMA DE ROBÓTICA COLECTIVA CON PROCESAMIENTO CENTRALIZADO ENTRE DOS ROBOTS HUMANOIDES BIOLOID PREMIUM”, ha sido desarrollado citando las fuentes correspondientes y respetando las disposiciones legales que protegen los derechos de autor vigente.

Por tal razón, las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo son exclusiva responsabilidad del autor.

**Ricardo Daniel Tabango Obaco**

Sangolquí, enero de 2014

## **AUTORIZACIÓN**

Yo, Ricardo Daniel Tabango Obaco,

Autorizo a la Universidad de las Fuerzas Armadas “ESPE” a la publicación, en la biblioteca virtual de la institución del proyecto de grado titulado: “DESARROLLO DE UN SISTEMA DE ROBÓTICA COLECTIVA CON PROCESAMIENTO CENTRALIZADO ENTRE DOS ROBOTS HUMANOIDES BIOLOID PREMIUM”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

**Ricardo Daniel Tabango Obaco**

Sangolquí, enero de 2014

## DEDICATORIA

A mis padres y mi hermana  
por su amor incondicional durante toda mi vida

## AGRADECIMIENTOS

Agradezco a mis padres Patricio y Estela, y a mi hermana Lorena, que me han apoyado toda la vida y mucho más durante el desarrollo de este proyecto.

A mis tíos Oscar, Cumandá, Diego y Teresa que me brindaron su confianza al ser participantes en el crédito educativo que me ayudó a seguir estudiando.

A mis mejores amigos Marco y Santiago que siempre estuvieron ayudándome o ejerciendo presión para que este proyecto saliera adelante.

A mi director y codirector que me guiaron adecuadamente durante el desarrollo de este proyecto.

Gracias a todos ellos.

## ÍNDICE DE CONTENIDOS

CERTIFICACIÓN DE LA ELABORACIÓN DEL PROYECTO .....	i
AUTORÍA DE RESPONSABILIDAD .....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTOS.....	v
ÍNDICE DE CONTENIDOS .....	vi
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS.....	xi
ANEXOS .....	xiv
RESUMEN.....	xv
ABSTRACT.....	xvi
CAPÍTULO 1.....	1
1.1. ANTECEDENTES.....	1
1.2. JUSTIFICACIÓN E IMPORTANCIA .....	3
1.3. ALCANCE DEL PROYECTO.....	5
1.4. OBJETIVOS .....	6
1.4.1. OBJETIVO GENERAL.....	6
1.4.2. OBJETIVOS ESPECÍFICOS .....	6
1.5. DESCRIPCIÓN DEL PROYECTO.....	7

1.5.1.	SISTEMA MECÁNICO .....	7
1.5.2.	SISTEMAS DE CONTROL .....	7
1.5.3.	SISTEMA ELECTRÓNICO .....	8
1.5.4.	SOFTWARE PARA IMPLEMENTACIÓN Y ANÁLISIS .....	9
CAPÍTULO 2: MARCO TEÓRICO .....		10
2.1	ROBÓTICA HUMANOIDE .....	11
2.1.1	ORIGEN .....	12
2.1.2	ROBOTS BÍPEDOS .....	13
2.1.3	PROPÓSITO .....	16
2.2	ROBÓTICA COLECTIVA Y COOPERATIVA .....	17
2.2.1	ORIGEN .....	19
2.2.2	CARACTERÍSTICAS .....	19
2.2.3	CLASIFICACIÓN .....	22
2.3	PROYECTOS RELACIONADOS .....	26
CAPÍTULO 3: HARDWARE .....		29
3.1	DESCRIPCIÓN .....	30
3.1.1	CONTROLADOR .....	31
3.1.2	ACTUADORES .....	32
3.1.3	SENSORES .....	33
3.1.4	MÓDULO DE COMUNICACIÓN .....	34



3.2	ANÁLISIS MECÁNICO .....	36
3.2.1	CINEMÁTICA DEL ROBOT .....	36
3.2.2	ANÁLISIS DE DEFORMACIONES.....	45
CAPÍTULO 4: CONTROL DEL SISTEMA.....		50
4.1	DISEÑO .....	51
4.1.1	TRABAJO REQUERIDO .....	51
4.1.2	MATERIALES .....	52
4.1.3	CONTROL.....	55
4.1.4	ARQUITECTURA.....	56
4.2	PROGRAMACIÓN EMBEBIDA DE LAS RTU .....	58
4.2.1	MOVIMIENTO DE LAS RTU .....	58
4.2.2	COMPORTAMIENTO DE LAS RTU.....	63
4.3	PROGRAMACIÓN DE LA MPU.....	71
4.3.1	COMUNICACIÓN ENTRE LA MPU Y LAS RTU .....	71
4.3.2	VISIÓN ARTIFICIAL.....	74
4.3.3	ROBÓTICA COLABORATIVA .....	81
CAPÍTULO 5: RESULTADOS.....		95
5.1	POSIBILIDADES DEL ALGORITMO .....	96
5.1.1	EXPANSIÓN DEL SISTEMA.....	96
5.1.2	CAMBIO DE LOS ESPACIOS DIVIDIDOS DE TRABAJO .....	97

5.1.3	TRABAJO CON UN ROBOT .....	97
5.2	PRUEBAS.....	99
5.2.1	UBICACIÓN .....	99
5.2.2	MOVIMIENTO DE LOS ROBOTS .....	102
5.3	EVALUACIÓN DEL PROYECTO.....	104
	CAPÍTULO 6.....	105
	CONCLUSIONES Y RECOMENDACIONES .....	105
6.1	CONCLUSIONES .....	105
6.2	RECOMENDACIONES.....	107
	REFERENCIAS BIBLIOGRÁFICAS.....	109
	ANEXOS.....	112

## ÍNDICE DE TABLAS

Tabla 2.1: Características de los sistemas multi-robot.....	19
Tabla 3.1: Características del actuador AX-12A .....	33
Tabla 4.1: Ventajas y desventajas del nivel de rigidez.....	61
Tabla 4.2: Paquetes de comunicación del AX-12A .....	68
Tabla 4.3: Comandos de orden al robot.....	70
Tabla 4.4: Configuración del puerto de comunicación .....	72
Tabla 4.5: Valores principales del programa.....	85
Tabla 5.1: Comparación del Trabajo.....	99
Tabla 5.2: Mediciones corrección de coordenadas.....	100
Tabla 5.3: Errores de las mediciones.....	101
Tabla 5.4: Tiempo por pasada .....	103
Tabla 5.5: Evaluación del proyecto .....	104

## ÍNDICE DE FIGURAS

Figura 2.1: Descripción del Capítulo 2 .....	11
Figura 2.2: “Robot de Leonardo” .....	12
Figura 2.3: Clasificación de los Robots Bípedos .....	13
Figura 2.4: Robot bípedo con rodilla invertida .....	14
Figura 2.5: Robot bípedo con rodilla normal .....	15
Figura 2.6: Robot bípedo sin rodilla .....	15
Figura 2.7: Propósitos de la Robótica Humanoide .....	16
Figura 2.8: Clasificación de la Robótica Colectiva .....	22
Figura 2.9: Clasificación por el Nivel de Comunicación .....	22
Figura 2.10: Clasificación por la Arquitectura de Control .....	24
Figura 2.11: Sistema Multi-robot con Arquitectura Distribuida .....	24
Figura 2.12: Sistema Multi-robot con Arquitectura Centralizada .....	25
Figura 2.13: Sistema Multi-robot con Arquitectura Mixta .....	26
Figura 3.1: Descripción del Capítulo 3 .....	29
Figura 3.2: Modelo del Robot .....	30
Figura 3.3: Controlador CM-510 .....	31
Figura 3.4: Actuador AX-12A .....	33
Figura 3.5: GS-12 (izq) y DMS-80 (der) .....	34
Figura 3.6: Circuito del sensor TCRT 5000 .....	34
Figura 3.7: Zig100 (izq), Zig110A (centro) y Zig2Serial (der) .....	35
Figura 3.8: Cadena Cinemática del Robot .....	36
Figura 3.9: Puntos de referencia del robot .....	37

Figura 3.10: Matriz de transformación homogénea.....	38
Figura 3.11: Puntos de Referencia del torso.....	39
Figura 3.12: Puntos de referencia parte inferior.....	42
Figura 3.13: Análisis de la pieza FP04-F1 .....	46
Figura 3.14: Análisis de la pieza FP04-F2 .....	47
Figura 3.15: Análisis de la pieza FP04-F4 .....	48
Figura 3.16: Análisis de la pieza FP04-F5 .....	49
Figura 4.1: Organización del Capítulo 4.....	50
Figura 4.2: Formas de división del espacio.....	51
Figura 4.3: Espacios Divididos de Trabajo.....	52
Figura 4.4: Arquitectura general del sistema .....	57
Figura 4.5: Roboplus Motion .....	59
Figura 4.6: Información de las páginas de movimiento.....	60
Figura 4.7: Página de movimiento del robot .....	64
Figura 4.8: Funcionamiento Librería ADC.c.....	67
Figura 4.9: Funcionamiento Librería Serial.c .....	69
Figura 4.10: Conexión Zigbee.....	71
Figura 4.11: Funcionamiento comunicación desde la MPU .....	73
Figura 4.12: Subproceso de comunicación desde la MPU en Labview .....	74
Figura 4.13: Formato 4:3 y 16:9.....	75
Figura 4.14: Imagen original (izq) e imagen filtrada (der) .....	77
Figura 4.15: Patrón de la pelota.....	77
Figura 4.16: Patrón del robot en 0° .....	78

Figura 4.17: Diferentes ubicaciones robot .....	79
Figura 4.18: Zona de Influencia de cada robot .....	82
Figura 4.19: Orientación de la pelota con respecto al robot.....	82
Figura 4.20: Valores en espacio de trabajo .....	85
Figura 4.21: Diagrama de Flujo del Programa (1A) .....	86
Figura 4.22: Diagrama de Flujo del Programa (2A) .....	87
Figura 4.23: Diagrama de Flujo del Programa (2A1) .....	88
Figura 4.24: Diagrama de Flujo del Programa (2B) .....	89
Figura 4.25: Diagrama de Flujo del Programa (2C) .....	90
Figura 4.26: Diagrama de Flujo del Programa (3A) .....	91
Figura 4.27: Diagrama de Flujo del Programa (3A1) .....	92
Figura 4.28: Diagrama de Flujo del Programa (3B) .....	93
Figura 4.29: Diagrama de Flujo del Programa (3C) .....	94
Figura 5.1: Organización del capítulo 5 .....	95
Figura 5.2: Zonas de Influencia con (a) 3 robots y (b) 5 robots.....	96
Figura 5.3: Pruebas con 1 robot (izq) y dos robots (der).....	98

## **ANEXOS**

Anexo A Matrices Cinemáticas de las Piernas del Robot

Anexo B Programa Principal de la MPU

## RESUMEN

La robótica colectiva busca diseñar sistemas compuestos de varios robots capaces de resolver problemas conjuntamente. Actualmente los robots humanoides son estudiados en el mundo con dos propósitos, el primero de ámbito funcional y el otro con un objetivo experimental. En este documento se describe el trabajo hecho con dos robots humanoides de características similares para que trabajen colectivamente por medio de una unidad de procesamiento central que se basa en la visión artificial y datos de sensores de los robots. Con estos datos la unidad de procesamiento central puede dar órdenes a los robots. La meta final de este proyecto consiste en que dos robots (RTU) se pasen una pelota de un lado a otro lado de una superficie de trabajo usando una unidad central de procesamiento (MPU).

Palabras Clave:

- Robótica Colectiva
- Robótica Humanoide
- Unidad Central de Procesamiento
- Visión Artificial
- ROVITEL



## ABSTRACT

Collective robotics seeks to design systems composed of several robots that can solve problems together. Currently, humanoid robots are being studied around the world for two purposes, the first is a functional purpose and the other one is a experimental purpose. This document describes the work done with two humanoid robots with similar features that work collectively through a central processing unit which is based on machine vision and sensor data of the robot. With these data the central processing unit can command the robots. The ultimate goal of this project is that two robots (RTU) pass a ball across a work surface using a central processing unit (MPU).

Keywords;

- Collective Robotics
- Humanoid Robotics
- Central Processing Unit
- Machine Vision
- ROVITEL

# **CAPÍTULO 1**

## **DESARROLLO DE UN SISTEMA DE ROBÓTICA COLECTIVA CON PROCESAMIENTO CENTRALIZADO ENTRE DOS ROBOTS HUMANOIDES BIOLOID PREMIUM**

### **1.1. ANTECEDENTES**

El avance en el desarrollo de la robótica ha tenido un ritmo bastante acelerado en las últimas décadas, esto se ha dado gracias a los beneficios que los robots pueden aportar a la sociedad. Este avance ha hecho que los sistemas robóticos sean cada vez más complejos y por ende, más costosos, por esta razón se buscó una manera de simplificar sistemas robóticos y reducir costos, así se formuló la idea de la Robótica Colaborativa y Colectiva.

La robótica colectiva busca diseñar sistemas compuestos de varios robots capaces de resolver problemas conjuntamente. Los robots que forman parte de un sistema multi-robot son simples en términos de diseño y control, y menos costosos que los sistemas de un sólo robot especializado. Los sistemas multi-robot están orientados a resolver problemas en los cuales la participación de un solo robot no es suficiente o resulta ser muy costosa,

en términos de diseño y tiempo, como por ejemplo el transporte de objetos voluminosos, el manejo de material peligroso, la exploración y cobertura de terreno. Al dividir las tareas entre varios equipos, el trabajo se realiza de una manera más rápida y eficiente, y un fallo en el funcionamiento de un robot puede ser subsanado por el resto del equipo.

Alrededor del mundo se están desarrollando cada vez más aplicaciones de robótica colectiva, como es el caso de los robots de Kiva Systems, los cuales usan el concepto de robótica colectiva para poder organizar los productos que se encuentren en unas grandes bodegas de almacenamiento de productos. El propósito de estos robots es sustituir a los antiguos sistemas de manejo de productos. Al usar la robótica colectiva, estos robots se dividen el trabajo de transporte de productos dentro de la bodega. Así, gracias a la comunicación entre robots y a sus sensores pueden por ejemplo evitar obstáculos y seguir con el trabajo de un robot que haya sufrido un desperfecto.

Actualmente los robots humanoides son estudiados en el mundo con dos propósitos, el primero de ámbito funcional y el otro con un objetivo experimental. El propósito funcional es en el que los robots humanoides trabajan con herramientas diseñadas para los humanos o en ambientes humanos. En tanto que, el propósito experimental es poder estudiar diferentes aspectos del cuerpo humano como por ejemplo la locomoción bípeda.

El estudio de robótica colectiva con robots humanoides se está dando en varias partes del mundo. La Robocup es un evento anual que consiste en una copa de fútbol en la que participan diferentes tipos de robots. En esta copa existe una categoría de robots humanoides en la que se enfrentan dos equipos, los cuales son programados para colaborar entre sí para anotar el mayor número de puntos. El objetivo de la Robocup es promover la investigación de la robótica y la inteligencia artificial con el fin de que en el año 2050 se obtenga un equipo que pueda ganar un partido de fútbol en contra del equipo humano campeón de la copa mundial de ese año.

## **1.2. JUSTIFICACIÓN E IMPORTANCIA**

La robótica a nivel mundial tiene un gran auge debido a las diferentes aplicaciones que se van desarrollando alrededor del planeta. Existen varias áreas en las que los robots han tomado un papel protagónico como es el caso de la industria de manufactura. Sin embargo, existen otras áreas de la robótica en las que la investigación está en sus primeros pasos. Este es el caso de la robótica colaborativa la cual tiene poco tiempo de desarrollo con respecto a otras áreas de la robótica.

A continuación se enlistan algunos proyectos de robótica colectiva a nivel mundial:

En el Departamento de Ingeniería Eléctrica y Computación en la Universidad de Missouri junto a la Marina de los Estados Unidos se está desarrollando un robot humanoide que realice un trabajo colaborativo con

otros robots y con los humanos en misiones espaciales. Su objetivo es que los robots colaboren con los astronautas tan natural y eficientemente como lo harían con otros astronautas.

Por otra parte, en el Laboratorio de Sistemas Autónomos de la Universidad Drexel en Estados Unidos se está trabajando con robot humanoides del modelo HUBO. Este proyecto pretende alcanzar una colaboración entre humano y robot para hacer diferentes actividades cotidianas como por ejemplo cargar cosas pesadas.

Como último ejemplo, en el Centro de Robótica y Sistemas Inteligentes de la Universidad de Plymouth se está diseñando un sistema robótico colaborativo para el juego de fútbol. Este proyecto utiliza robots de la gama Bioloid y cuentan con un PDA como procesador.

Como se puede evidenciar, a través de la investigación en la universidad se genera conocimiento que fomenta el desarrollo de los países en varios ámbitos como la tecnología y la robótica. En algunos países del mundo, la investigación de la robótica colectiva y de los robots humanoides se está desarrollando paulatinamente, lo que ha generado nuevos conocimientos. Es así que en el Ecuador se debe investigar para alcanzar los conocimientos que tienen en otras partes del planeta.

Por todo esto el grupo de investigación "ROVITEL" de la ESPE empezó una investigación en cuanto a la robótica colaborativa basado en la plataforma de robots Fischertechnik que tienen un procesamiento no centralizado. A partir del desarrollo de este proyecto nació la necesidad de

incursionar en aplicaciones con procesamiento centralizado. Este proyecto contribuirá con nuevos conocimientos en esta área y en la de robots humanoides. Cabe destacar que en el país aún no se ha desarrollado proyectos en ésta área. Estos conocimientos serán aprovechados por la universidad. De esta manera docentes y estudiantes desarrollarán nuevos avances en robótica cooperativa humanoide para que en el futuro los robots ayuden a los humanos en tareas del diario vivir.

### **1.3. ALCANCE DEL PROYECTO**

En este proyecto se va a utilizar dos robots humanoides de la gama Bioloid Premium los cuales deberán realizar un trabajo cooperativo. Se contará con un sistema central de procesamiento que emitirá órdenes a los robots. Estas órdenes estarán basadas en datos que le proporcionaran los robots y un dispositivo de visión artificial. Como trabajo colaborativo se ejecutará una de las etapas en un juego de fútbol: los robots se pasarán una pelota. Este trabajo se lo realizará en un espacio rectangular de 2 metros por 1.5 metros. Al contar con dos robots colaborando entre sí, se va a realizar el trabajo de una manera más rápida de lo que se haría si se contara con un solo robot con lo que se verificará el nivel de cooperación alcanzado.

## **1.4. OBJETIVOS**

### **1.4.1. OBJETIVO GENERAL**

Implementar un sistema de robótica colectiva con el uso de dos robots humanoides de modelo Bioloid de características similares, que use un sistema centralizado de procesamiento.

### **1.4.2. OBJETIVOS ESPECÍFICOS**

Objetivos:

- Diseñar un modelo de programación para la robótica colectiva entre dos robots.
- Desarrollar un sistema de medición de distancias y ubicación confiable.
- Determinar el nivel cooperación alcanzado.

Metas:

- Manejar el software propietario de los kits de robótica Bioloid.
- Analizar las propiedades mecánicas que tienen los robots.
- Analizar los aspectos mecánicos de un robot humanoide Bioloid.
- Programar los movimientos del robot.
- Programar las comunicaciones del robot.

- Programar la computadora conectada a la cámara para visión artificial.
- Programar el control entre la cámara y los robots.
- Desarrollar las rutinas de software para crear un sistema de exploración del entorno, y distribuir la información.
- Realizar pruebas de funcionamiento.
- Realizar análisis de resultados.

## **1.5. DESCRIPCIÓN DEL PROYECTO**

### **1.5.1. SISTEMA MECÁNICO**

Se va a trabajar con dos robots que permiten configuraciones humanoides, de aproximadamente 400 mm de alto cada uno y que poseen hasta 18 grados de libertad. En el sistema mecánico se pretende analizar diferentes características que permiten realizar el movimiento de caminar en robots bípedos.

### **1.5.2. SISTEMAS DE CONTROL**

El sistema de control del proyecto estará basado en una arquitectura centralizada que consiste en unidad maestra de procesamiento (MPU) y en dos unidades remotas de procesamiento (RTU).



La MPU se encargará de recibir información y emitir órdenes hacia las RTU basados en los algoritmos de trabajo cooperativo y de visión artificial (reconocimiento de patrones y medición de distancias).

Las RTU estarán ubicadas en cada robot y se encargarán de los algoritmos de movimiento y estabilidad, comunicación con la MPU y adquisición de datos de sensores.

### **1.5.3. SISTEMA ELECTRÓNICO**

Las variables de los robots que se van a controlar a través del sistema electrónico en los robots son posición, orientación, estabilidad y distancia al objetivo.

Para el control de estas variables se usará:

- Una cámara web de alta definición que determinará la posición y orientación.
- Un sensor de estabilidad el cual ayudará en la caminata al robot y detectará si este cae al suelo.
- Dos sensores de detección de objetos de corto alcance (máximo 15 mm) que serán ubicados en los pies. Estos sensores en conjunto a la cámara web detectarán la distancia al objetivo.

- Un sensor de detección de objetos de largo alcance (800 mm) para evitar el choque contra grandes objetos. Este será colocado en la parte superior del robot.

La MPU estará basada en una computadora personal la cual contará con un módulo de comunicación inalámbrico ZigBee (estándar IEEE 802.15.4) y con la cámara web. La RTU consistirá en un controlador propio de los robots y cuenta cada una con un microcontrolador AVR, puertos para la conexión tipo “Daysi Chain” hacia los servomotores, conexiones individuales para cada sensor y un puerto para el módulo de conexión inalámbrica ZigBee.

#### **1.5.4. SOFTWARE PARA IMPLEMENTACIÓN Y ANÁLISIS**

El desarrollo del software para el controlador de los robots se lo va a realizar con el programador propio de los AVR que está basado en C.

Se va a usar LabVIEW para desarrollar el algoritmo de control de robótica colectiva, en tanto que para la visión artificial se va a usar los algoritmos de reconocimiento de patrones y medición de distancias en el mismo programa.

## **CAPÍTULO 2**

### **MARCO TEÓRICO**

En este capítulo se explicará la base teórica en la que se basa el proyecto descrito en este documento. Primero se revisará la teoría con respecto a los robots humanoides. Incluye una breve explicación del origen de esta rama, además se incluye la descripción y clasificación de los robots bípedos. Esta primera parte termina explicando los propósitos de la robótica humanoide.

La segunda parte del capítulo está dedicada a la teoría de la Robótica Colectiva y Cooperativa, en la que se explica el origen, características y clasificación de la misma.

La tercera y última parte describe proyectos relacionados al tema de este documento, de esta manera se puede analizar el avance al que han llegado en otras partes del mundo.

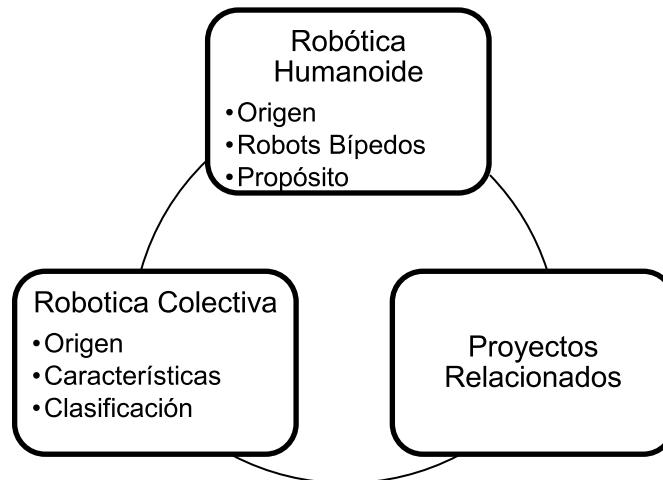


Figura 2.1: Descripción del Capítulo 2

## 2.1 ROBÓTICA HUMANOIDE

La robótica es una rama de la tecnología que combina varias ciencias como la mecánica, electrónica, inteligencia artificial, entre otras. El objetivo principal de la robótica es crear sistemas que puedan realizar trabajos independientemente. A estos sistemas los llamamos robots. El desarrollo de estos robots es más avanzado cada día y sus aplicaciones son variadas.

Dentro de la robótica existe una rama que se dedica al estudio de robots con formas humanas, llamada Robótica Humanoide. El objetivo final de la robótica humanoide es el diseño y construcción de un robot que sea lo más parecido a un ser humano.

### 2.1.1 ORIGEN

La historia de robots humanoides se remonta siglos atrás, los primeros que describieron una máquina con forma humana fueron los griegos, al cual denominaron “Autómata”<sup>1</sup> (Wadsworth & Few, 2010). Existen varios ejemplos de autómatas, uno de los más conocidos es el “Robot de Leonardo”, diseñado por Leonardo Da Vinci alrededor del año 1495 (Moran, 2006), el cual era capaz de hacer movimientos parecidos a los de los humanos. Una réplica es mostrada en la exhibición “Leonardo da Vinci. Mensch - Erfinder – Genie” en Berlín la cual se puede ver en la figura 2.2.



Figura 2.2: “Robot de Leonardo”

Pese a que los humanos han construido autómatas desde hace bastante tiempo, estos no eran capaces de la acción de caminar en dos pies tal y como lo hacen los humanos. Este avance se dio hace relativamente poco tiempo con el estudio de la locomoción bípeda y los robots bípedos.

---

<sup>1</sup> Del griego “automatos”: actuar por sí mismo

## 2.1.2 ROBOTS BÍPEDOS

Se denomina robots bípedos a los robots que son capaces de caminar en dos pies. El diseño de robots de este tipo no fue posible hasta el año 1968 en el que Miomir Vukobratović introdujo el concepto de “Punto de Momento Zero”. Este concepto especifica el punto en el que la fuerza de reacción dinámica al contacto del pie con el suelo no produce momento alguno en la dirección horizontal, es decir, el punto donde el total de inercia vertical y fuerzas de gravedad equivale a cero (Vukobratović, 2004). Este concepto ayudó a desarrollar robots bípedos que pudieran caminar por sí solos.

Los robots bípedos se clasifican en tres grupos dependiendo de la configuración de rodilla que tengan, los cuales son:

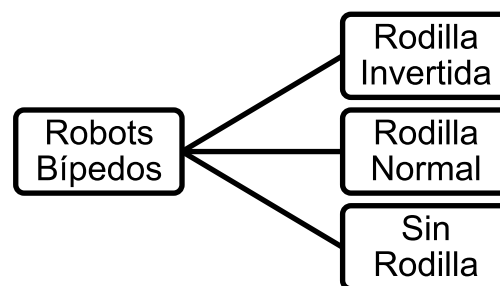


Figura 2.3: Clasificación de los Robots Bípedos

### 2.1.2.1 Rodilla Invertida

Los robots bípedos de rodilla invertida tienen, como su nombre lo indica, la articulación que corresponde a la rodilla dispuesta de modo que la pierna se flexiona de modo contrario que las piernas de los humanos. Esta

disposición de rodilla se basa en la aparente forma de las extremidades inferiores de las aves<sup>2</sup>. Se muestra en la figura 2.4 un claro ejemplo de este tipo de robot.



Figura 2.4: Robot bípedo con rodilla invertida

### 2.1.2.2 Rodilla Normal

Los robots bípedos con rodilla normal tienen la articulación correspondiente a la rodilla dispuesta de modo igual a la de los seres humanos. Esta configuración es la más usada ya que ayuda al estudio y entendimiento de la locomoción bípeda de los seres humanos. En la figura 2.5 se puede ver un ejemplo de un robot bípedo con rodilla normal.

---

<sup>2</sup> Las aves son animales digitígrados, es decir, caminan sobre los dedos de los pies. Lo que parece una rodilla invertida a simple vista en realidad es el tobillo.



Figura 2.5: Robot bípido con rodilla normal

### 2.1.2.3 Sin rodilla

Los robots bípedos sin rodilla no cuentan con la articulación correspondiente a la rodilla. Estos robots realizan el modo de caminado utilizando las articulaciones correspondientes a el tobillo y la cadera. En la figura 2.6 se puede notar la configuración que tiene un robot bípido sin rodilla.



Figura 2.6: Robot bípido sin rodilla



### 2.1.3 PROPÓSITO

La Robótica Humanoide en la actualidad no es un campo totalmente definido (Wadsworth & Few, 2010), en cambio es la colaboración de diferentes disciplinas y por lo tanto sus objetivos dependen de la ciencia que predomine diferentes investigaciones, así que se dividirá los propósitos de la robótica humanoide en dos grupos.

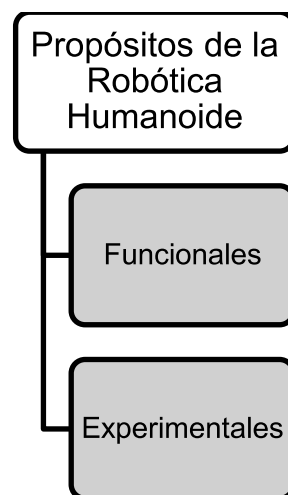


Figura 2.7: Propósitos de la Robótica Humanoide

#### 2.1.3.1 Propósitos Funcionales

Los propósitos funcionales de la robótica humanoide consisten en el desarrollo de robots humanoides con la finalidad de que estos sean usados por los humanos como herramientas.

Los humanos han desarrollado ambientes específicamente para que se desenvuelvan las personas, estos incluyen dispositivos y herramientas que solo pueden ser usados por la gente. Esta es la razón por la cual existen

diferentes investigaciones que tratan de adaptar robots al ambiente en el que desenvuelven los humanos, de esta manera los robots podrían ejercer trabajos que solo las personas pueden realizar con herramientas para humanos.

### **2.1.3.2 Propósitos Experimentales**

Los propósitos experimentales de la robótica humanoide se tratan acerca del estudio de diferentes características de los humanos. Estos estudios se centran en características como la locomoción bípeda, la manipulación de objetos con las manos, el aprendizaje y el comportamiento adaptativo. Algunos de los estudios con estos propósitos dan como resultado productos como las prótesis robóticas o exoesqueletos.

## **2.2 ROBÓTICA COLECTIVA Y COOPERATIVA**

Dentro del estudio de la robótica existen diferentes ramas, una de las cuales es la robótica colectiva y cooperativa. Esta rama de la robótica es relativamente nueva y se puede definir básicamente como el estudio, diseño e implementación de un conjunto de sistemas robóticos que interactúan entre sí o con las personas para cumplir un objetivo. Al conjunto de sistemas robóticos se les conoce como “Sistemas Multi-Robot”.

La diferenciación entre Robótica Colectiva y Robótica Cooperativa no está bien especificada todavía, ya que diferentes autores definen las dos de manera diferente.

Dos autores mexicanos definen la Robótica Colectiva como *“La robótica colectiva se interesa, de manera general, en el diseño y el control de sistemas multi-robot o multi-agente, grupos de robots o agentes trabajando de manera conjunta para lograr objetivos comunes.”* (León Fernández de Lara & Muñoz Meléndez, 2007).

En una de las exposiciones de la Conferencia IROS<sup>3</sup> del 2010 de la IEEE<sup>4</sup> se especifica que *“Uno de los objetivos a largo plazo de la robótica y los sistemas cognitivos artificiales es que los robots van a ser capaces de interactuar de una manera cooperativa y adaptativa con sus contra partes humanas en tareas que pueden cambiar en tiempo real”* (Lallee, y otros, 2010).

Un investigador brasileño define a la Robótica Colectiva como *“Un robot o conjunto de robots que trabajan con seres humanos para desarrollar una tarea”* y a la Robótica Cooperativa como *“Un grupo de robots que trabajan en conjunto para desarrollar una tarea”* (Montenegro Campos, 2012).

Se puede notar como las últimas dos definiciones difieren con las primera, y en la segunda no se da una definición exacta, en cambio menciona un objetivo en este campo. Por todo esto se puede asegurar que por el momento no hay la posibilidad de definir a la Robótica Colectiva y Robótica Cooperativa como términos aislados hasta que los investigadores se pongan de acuerdo en las definiciones correspondientes.

---

<sup>3</sup> Intelligent Robots and Systems (Robots y Sistemas Inteligentes)

<sup>4</sup> Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)

### 2.2.1 ORIGEN

Con el avance de la tecnología los robots se volvieron más complejos, lo que aumentó el tiempo de diseño y construcción además de los costos. La Robótica Colectiva se generó con la necesidad de crear sistemas multi-robot que puedan ejercer trabajos complicados de una manera eficiente y con bajo costo. Con la robótica colectiva se puede tener varios robots que ejecuten ciertas partes del trabajo en lugar de tener un robot especializado para ejercer una tarea complicada, de esta manera se aumenta la eficiencia y se disminuye la complejidad de cada robot.

### 2.2.2 CARACTERÍSTICAS

Un sistema multi-robot no siempre es una mejor opción frente a un solo robot, se debe analizar las características positivas y negativas de los sistemas multi-robot para decidir si es una buena opción para el trabajo que se quiera realizar.

<b>Características Positivas</b>	<b>Características Negativas</b>
Distribución de Tareas	Se necesita una comunicación robusta
Menor complejidad en diseño	Se necesita una mejor programación
Bajo costo	Ineficiente en espacios reducidos
Sistemas redundantes	
Menor tiempo de trabajo	

Tabla 2.1: Características de los sistemas multi-robot

A continuación se detallará en que consiste cada una de las características que constan en la Tabla 2.1.

### 2.2.2.1 Positivas

- **Distribución de Tareas:** Uno de los puntos fuertes de la robótica cooperativa es la posibilidad de que cada robot puede ejercer una tarea diferente que aportará al objetivo final del trabajo. También esta característica hace referencia a la distribución del espacio, se puede distribuir tareas iguales a cada robot que van a estar ubicados en determinados espacios dentro de un área más grande, con lo que se puede abarcar más terreno con más robots.
- **Menor complejidad de diseño:** Cada robot de un sistema multi-robot puede ser diseñado para ejecutar una parte de la tarea, por lo que cada robot solo tendrá que estar equipado con lo necesario para realizar su parte. Esto reduce la complejidad de diseño con respecto al diseño de un solo robot que necesita hacer todas las partes de la tarea.
- **Bajo costo:** Gracias a la menor complejidad del diseño de robots en los sistemas multi-robot, los costos de partes y producción de estos robots es menor a los de un robot complejo.
- **Sistemas redundantes:** Si en un sistema multi-robot se avería uno de los robots que están trabajando, se puede programar a otro para que termine el trabajo que estaba haciendo el robot averiado.
- **Menor tiempo de trabajo:** Al tener un número mayor de robots que están distribuidos en diferentes espacios y realizan partes de la

tarea, se puede reducir el tiempo final que tomará realizar la totalidad de la tarea.

### **2.2.2.2 Negativas**

- **Comunicación Robusta:** Cuando se tiene aplicaciones de robótica colaborativa que usen comunicación entre sus robots, es necesario que la comunicación sea confiable y robusta. Si la comunicación no cumple estos requisitos pueden existir problemas al momento de la ejecución de la tarea. Este requerimiento puede hacer que los costos y tiempos invertidos programando la comunicación aumenten.
- **Mejor Programación:** Pese a que los robots en un sistema multi-robot son menos complejos que en un sistema de robots individuales, se debe tener buenos conocimientos de programación ya que el programa que se haga va a ser usado en bastantes robots. Si existen errores en la programación se verían afectados todos los robots del sistema.
- **Ineficiente en espacios reducidos:** Para usar un sistema multi-robot se debe tomar en cuenta el espacio en donde se va a trabajar. Si se pone bastantes robots en espacios reducidos se pueden producir choques e interferencias.

### 2.2.3 CLASIFICACIÓN

La Robótica Colectiva se puede clasificar tomando en cuenta dos factores:



Figura 2.8: Clasificación de la Robótica Colectiva

#### 2.2.3.1 Por el Nivel de Comunicación

Un sistema multi-robot puede ser clasificado por su nivel de comunicación en:

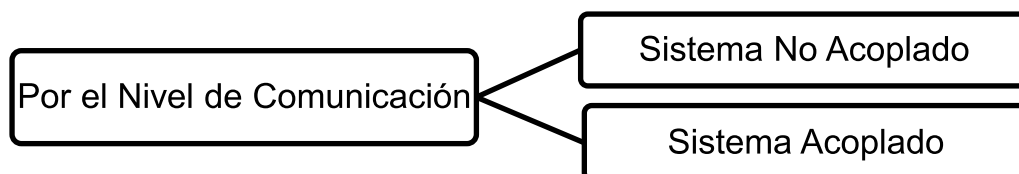


Figura 2.9: Clasificación por el Nivel de Comunicación

- **Sistema No Acoplado**

Cada robot perteneciente a este tipo de sistema sigue un conjunto de reglas por lo que el control es distribuido, pero no existe ningún tipo de

comunicación entre los robots o a una unidad central. Este tipo de robots no toma en cuenta los movimientos y acciones de los otros robots.

Como ejemplo de este tipo de sistema se puede mencionar a varios robots aspiradora distribuidos en diferentes lugares en una casa. Se puede decir que se trata de un sistema multi-robot que hace el trabajo más rápido pero que no necesita comunicación ya que cada robot está programado para ejercer su tarea.

- **Sistema Acoplado**

Un sistema acoplado es en el que los robots tienen la capacidad de comunicarse con los otros robots del sistema o con una unidad de procesamiento. Al tener cada robot una inteligencia se puede referir a la inteligencia colectiva como la suma de las inteligencias de los robots. Este tipo de sistema normalmente es usado cuando se necesita que los robots transfieran información.

Por ejemplo si tenemos varios robots en un campo minado en busca de minas personales, la comunicación ayuda para informar al resto de robots la localización de las minas que cada robot encuentra, además si un robot se daña puede solicitar que otro robot termine el trabajo.

### **2.2.3.2 Por la Arquitectura de Control**

Dependiendo de la manera en la que se vaya a programar el control, los sistemas multi-robot pueden ser de:



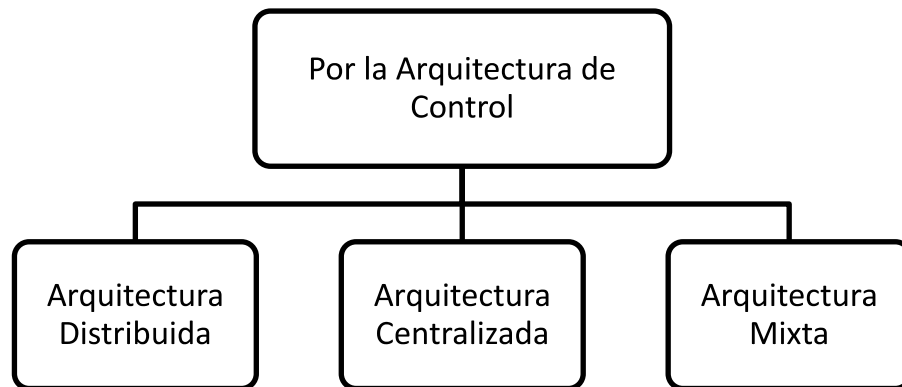


Figura 2.10: Clasificación por la Arquitectura de Control

- **Arquitectura Distribuida**

Los sistemas multi-robot con arquitectura distribuida cada robot recibe datos del mundo a través de sus sensores y los usa para tomar decisiones, los datos y las decisiones de cada robots pueden ser transmitidas a los otros robots los cuales pueden usar esta información para tomar sus propias decisiones.

Además la información compartida es útil cuando los robots deben sincronizarse para realizar un trabajo en conjunto, como por ejemplo para mover una caja que solo un robot no puede mover.

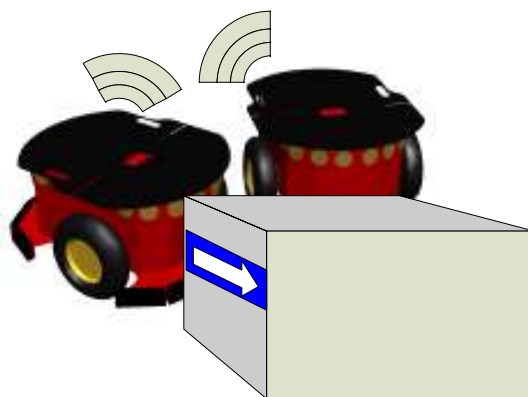


Figura 2.11: Sistema Multi-robot con Arquitectura Distribuida

- **Arquitectura Centralizada**

Los sistemas de robótica colectiva con arquitectura distribuida consisten varios robots, que denominaremos RTU<sup>5</sup>, los cuales se comunican con un sistema central de procesamiento, al cual llamaremos MPU<sup>6</sup>, el cual administra los datos que le proveen las distintas unidades RTU, analiza la información y envía órdenes a las RTU.

La MPU normalmente es estacionaria, por lo que se puede usar un equipo con más procesamiento y mejores características que las RTU.

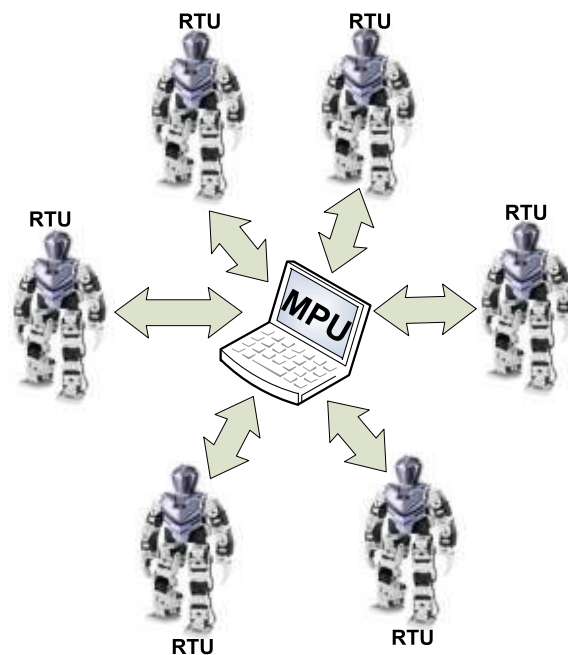


Figura 2.12: Sistema Multi-robot con Arquitectura Centralizada

- **Arquitectura Mixta**

Existe la posibilidad de que haya aplicaciones que usen una mezcla de las anteriores arquitecturas. Por ejemplo si se tiene dos grupos de robots en

---

<sup>5</sup> Remote Terminal Unit (Unidad Terminal Remota)

<sup>6</sup> Master Processing Unit (Unidad Maestra de Procesamiento)

lugares diferentes, cada grupo puede tener control distribuido en sus robots y el grupo en conjunto responde a una unidad central que maneja los grupos.

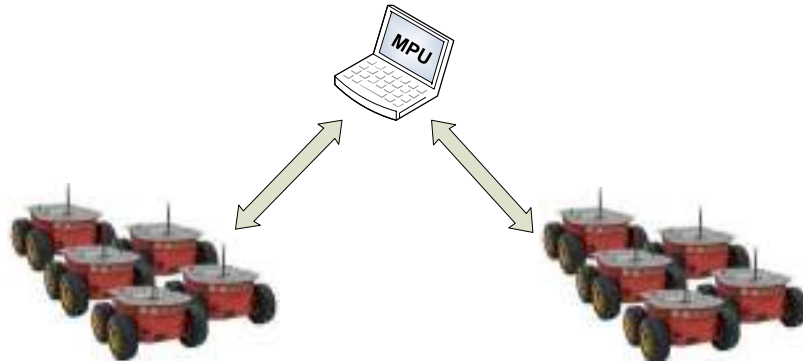


Figura 2.13: Sistema Multi-robot con Arquitectura Mixta

### 2.3 PROYECTOS RELACIONADOS

Existen diferentes proyectos que se han realizado o se están realizando en el área de Robótica Colectiva y Cooperativa en diferentes universidades y centros de investigación en el mundo.

Los temas en Robótica Colaborativa son bastante variados, esto debido a los diferentes tipos de robots que existen. En el año de 2007 en el Centro de Robótica y Sistemas Inteligentes en la Universidad de Plymouth se desarrolló proyecto cooperativo con robots Bioloid cuyo objetivo era jugar fútbol. Este proyecto usó un sistema no acoplado con una arquitectura distribuida para el control de robots. Cada robot fue modificado para que tuvieran una cámara cada uno que detectaba la pelota y el arco. Esta modificación supuso una carga de procesamiento para el controlador del

robot por lo que se adaptó un PDA<sup>7</sup> para el procesamiento de datos de la cámara (Wolf, Hall, Robinson, & Culverhouse, 2007).

En el año 2004 el MIT<sup>8</sup> Media Lab empezó un proyecto en colaboración con la NASA<sup>9</sup> para el desarrollo de la plataforma robótica conocida como *Robonaut*, cuyo propósito era trabajar en colaboración con astronautas humanos en operaciones de mantenimiento en misiones espaciales (Hoffmanand & Breazeal, 2004). En la actualidad ya existe el *Robonaut 2* que está siendo desarrollado por la NASA y General Motors.

En el año 2009 el Laboratorio de Robótica Distribuida del MIT empezó con el proyecto denominado *Materia Programable* cuyo objetivo es tener un grupo de pequeños robots cúbicos, con sistema acoplado y control distribuido, que cambien de posición y se armen de diferentes formas de manera autónoma según la forma deseada. El objetivo de este proyecto es crear sistemas que ayuden a soldados en combate a usar diferentes dispositivos que se puedan reconfigurar (antenas, trípodes, asientos, etc.) con los mismos materiales (Distributed Robotics Laboratory, 2009).

En el año 2012 el grupo de investigación ROVITEL empezó una investigación de robótica colaborativa con la plataforma de robots Fischertechnik. Este proyecto se basó en un sistema acoplado con arquitectura distribuida y usó robots móviles con ruedas de tipo oruga, los cuales contaban con dos servomotores, sensores ultrasónicos y sensores de

---

<sup>7</sup> Personal Digital Assistant (Asistente Digital Personal)

<sup>8</sup> Massachusetts Institute of Technology (Instituto de Tecnología de Massachusetts)

<sup>9</sup> National Aeronautics and Space Administration (Administración Nacional de la Aeronáutica y del Espacio)

presencia. El objetivo de este sistema era el transporte de cajas y usaba la robótica cooperativa cuando uno de los robots no era capaz de llevar una caja por sí solo debido al peso de la caja.

El presente proyecto es el siguiente paso de la investigación del proyecto que se mencionó anteriormente. Este proyecto será un sistema acoplado, que a diferencia del anterior, utilizará un sistema centralizado con robots humanoides. Las características de hardware y el desarrollo de software de este proyecto se detallarán en los siguientes capítulos.

## CAPÍTULO 3

### HARDWARE

En la primera parte de este capítulo se verá una descripción detallada de los componentes de hardware que se utilizan en el proyecto, principalmente enfocado hacia el robot Bioloid Premium y sus partes, las cuales incluyen controlador, actuadores, sensores y módulo de comunicación.

En la segunda parte se hará un análisis mecánico del robot. Este análisis incluye la cinemática del robot que se distribuye en estructura cinemática de las extremidades y el modelado cinemático final. Además se hace un estudio de deformación de las principales piezas que componen el armazón del robot y que están fabricadas de in plástico de ingeniería.

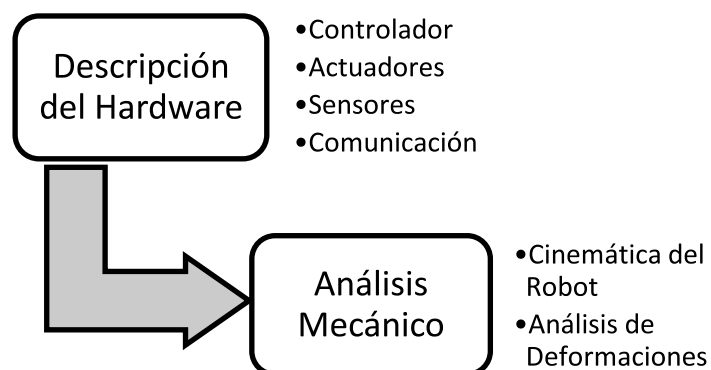


Figura 3.1: Descripción del Capítulo 3

### 3.1 DESCRIPCIÓN

Para el presente proyecto se utiliza dos robots de tipo “Bioloid Premium” fabricados por la compañía coreana “Robotis” los cuales pueden ser configurados como robots humanoides. El robot en configuración humanoide tiene una altura de 397 mm y un peso de 1.7 Kg. Todas las piezas del armazón del robot están hechas de plástico. Cada robot humanoide puede contener de 16 a 18 servomotores en cada articulación, dependiendo de la configuración que se elija. Además tiene un sensor de presencia de alcance medio, dos sensores de presencia de bajo alcance, un acelerómetro de dos ejes y un módulo de comunicación zigbee. En la figura 3.2 se puede observar un modelo de este robot desarrollado en el software SolidWorks de la compañía Dassault Systemes.



Figura 3.2: Modelo del Robot

Además de los robots, como parte del proyecto también se usa una cámara web de alta definición de modelo Logitech C310 que tiene una conexión USB 2.0 y un micrófono integrado. Esta cámara se usará para la detección de los robots y la pelota.

A continuación se dará una descripción detallada de los componentes del robot.

### 3.1.1 CONTROLADOR

El robot Bioloid Premium tiene equipado un controlador que se puede ver en la figura 3.3 que se denomina CM-510. Este controlador está basado en el microcontrolador AVR de 8 bits ATmega 2561 y cuenta con 6 puertos auxiliares de 5 pines en los que se puede conectar dispositivos periféricos tales como sensores, también posee 5 puertos de 3 pines que se usan para la conexión TTL con los servo motores, además tiene conectores para la comunicación con la computadora, conexión inalámbrica y un conector de poder. Este controlador también posee luces indicadoras y pulsadores que pueden ser programados para diferentes tareas.



Figura 3.3: Controlador CM-510



### 3.1.2 ACTUADORES

El robot Bioloid cuenta con 18 servo motores del modelo Dynamixel AX-12A de la compañía Robotis. Este actuador cuenta con un motor de precisión de corriente continua, un reductor de engranes y un circuito de control que incorpora un microcontrolador AVR de 8 bits ATmega8. A pesar de su reducido tamaño cuenta con un gran torque gracias al diseño de su reductor de engranes. Este actuador cuenta con la capacidad de detectar varios factores de operación como su temperatura, posición y voltaje de alimentación. La velocidad máxima de este motor es de 59 rpm y puede ser configurado para que funcione como motor normal o servo motor con una movilidad de 300°. En la figura 3.4 se tiene el modelo de este actuador. A continuación se detallará otras características del actuador.

<b>Peso:</b>	54.6 g
<b>Dimensiones:</b>	32mm * 50mm * 40mm
<b>Resolución:</b>	0.29°
<b>Razón de reducción de los engranes</b>	254 : 1
<b>Tórque Máximo</b>	1.5N.m (a 12.0V, 1.5A)
<b>Velocidad sin carga</b>	No load speed : 59rpm (at 12V)
<b>Ángulo de operación</b>	300°, giro completo
<b>Temperatura de operación</b>	-5°C a 85°C
<b>Señal de comunicación</b>	Paquete digital de datos
<b>Protocolo de comunicación</b>	Serial Half-duplex asíncrona
<b>Capa física de comunicación</b>	TTL (Conector tipo Daisy Chain)
<b>Velocidad máxima de comunicación</b>	1 Mbps
<b>Material</b>	Plástico de Ingeniería

Tabla 3.1: Características del actuador AX-12A



Figura 3.4: Actuador AX-12A

### 3.1.3 SENSORES

El robot Bioloid posee diferentes sensores que le ayudan a tomar datos del entorno que le rodea. El primero de los sensores es el GS-12 que es un sensor de velocidad angular que puede medir desde  $-300$   $^{\circ}/s$  a  $300$   $^{\circ}/s$  en dos ejes, este sensor se usa para poder balancear el robot y para detectar caídas imprevistas del robot.

El segundo sensor es el DMS-80, que es un sensor de distancia fabricado por Sharp y puede detectar objetos en una distancia desde los 100 mm hasta los 800 mm. Este sensor es usado por el robot para detectar obstáculos y evitarlos en el movimiento de caminado.

Estos dos sensores están ilustrados en las figura 3.5.



Figura 3.5: GS-12 (izq) y DMS-80 (der)

Además de estos sensores se añadieron dos sensores ópticos TCRT 5000 a los pies del robot para la detección de la pelota. Estos sensores fueron conectados a las entradas auxiliares 2 y 6 del controlador, se usó el circuito de la figura 3.6 para que los sensores funcionen. Este circuito fue usado debido al reducido tamaño que se tiene en los pies del robot.

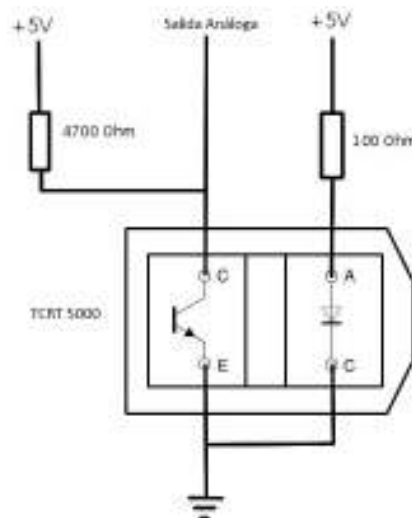


Figura 3.6: Circuito del sensor TCRT 5000

### 3.1.4 MÓDULO DE COMUNICACIÓN

El controlador CM-510 del robot tiene un puerto dedicado para la conexión de un módulo de comunicación inalámbrico, este módulo puede ser infrarrojo, bluetooth o Zigbee. Para este proyecto se escogió el módulo

Zigbee Zig 110A, el cual no necesita que esté en una posición predeterminada como el módulo infrarrojo y su costo de adquisición es menor que el módulo bluetooth.

Se denomina Zigbee a la especificación de un conjunto de protocolos de comunicación inalámbrica que está basado en el estándar IEEE 802.15.4 perteneciente a las redes inalámbricas de área personal.

Este módulo se comunica con un módulo Zig 100 que va a conectado a la computadora por medio de un adaptador llamado Zig2Serial. Este módulo convierte la señal del puerto serial de la computadora para transmitirla inalámbricamente, la cual es receptada por el módulo Zig 11A que está conectado al robot donde transforma la información y la pasa al controlador del robot. En la figura 3.7 se ilustran estos dispositivos.



Figura 3.7: Zig100 (izq), Zig110A (centro) y Zig2Serial (der)

## 3.2 ANÁLISIS MECÁNICO

### 3.2.1 CINEMÁTICA DEL ROBOT

El robot Bioloid Premium configurado como humanoide “Tipo A” tiene 18 GDL<sup>10</sup> los cuales corresponden a los 18 servo motores AX-12A. Cada brazo tiene 3 GDL y cada pierna tiene 6 GDL. La cadena cinemática del robot se muestra en la figura 3.8, donde todos los motores tienen la posición de 0°. Los ángulos de cada articulación están nombrados desde  $\theta_1$  hasta  $\theta_{18}$ . Esta nomenclatura se basa en los identificadores predeterminados de los actuadores en la configuración humanoide. El lado derecho tiene las variables con números impares y el izquierdo tiene números pares.

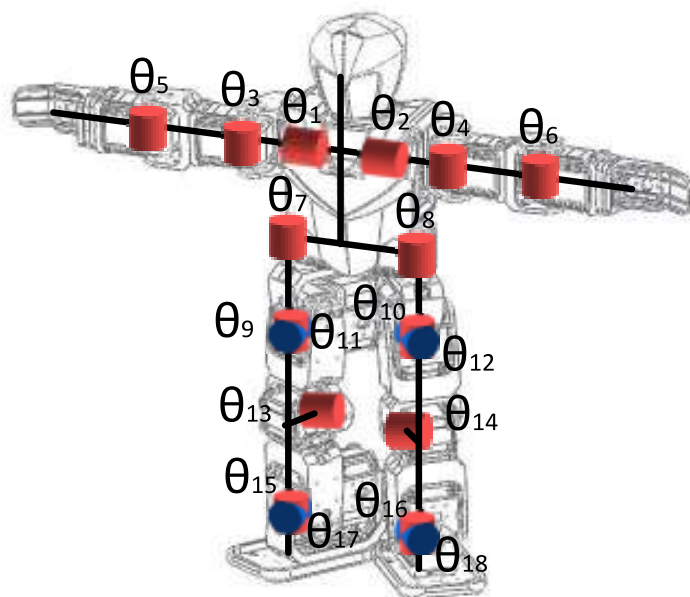


Figura 3.8: Cadena Cinemática del Robot

Para el modelamiento del robot se necesitan agregar dos puntos de referencia, uno en el pecho y uno en la cadera del robot que serán

<sup>10</sup> Grado de Libertad

denominados  $\Omega_P$  y  $\Omega_C$  respectivamente. Para el cálculo cinemático también es necesario tomar en cuenta los puntos de referencia de las manos y pies los cuales estarán denominados de la siguiente forma: mano derecha ( $\Omega_{MD}$ ), mano izquierda ( $\Omega_{MI}$ ), pie derecho ( $\Omega_{PD}$ ) y pie izquierdo ( $\Omega_{PI}$ ). Estos puntos de referencia se pueden observar en la figura 3.9.

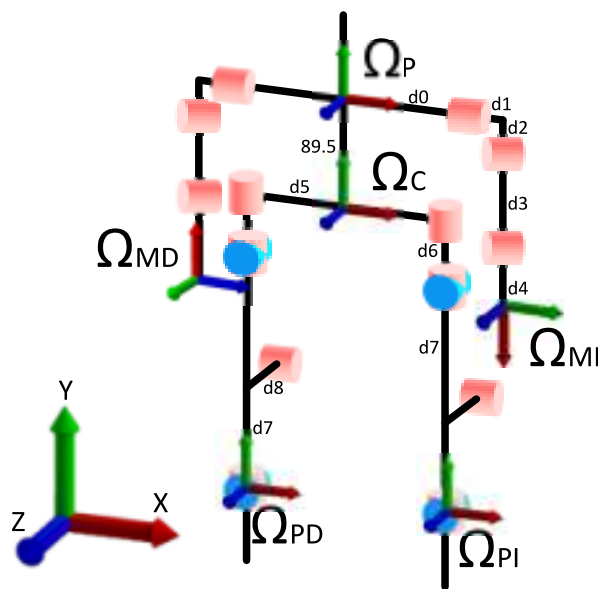


Figura 3.9: Puntos de referencia del robot

La figura 3.9 muestra la estructura del robot en la posición de parado, donde la mayoría de los actuadores tienen la posición de  $0^\circ$ , excepto por  $\theta_1=-90^\circ$ ,  $\theta_2=90^\circ$ ,  $\theta_3=-90^\circ$ ,  $\theta_4=90^\circ$ ,  $\theta_7=-45^\circ$  y  $\theta_8=45^\circ$ . La distancia entre puntos de referencia y servo motores está marcada desde d1 hasta d8, la distancia entre  $\Omega_P$  y  $\Omega_C$  es de 89.5 mm.

Como referencia para el cálculo de se va a usar un método conocido como Matriz de Transformación Homogénea usado en el proyecto llamado “*Kinematic Analysis for Robot Arm*” de Sirma Yavuz. Este método es

explicado en el libro “*Planning Algorithms*” de Steven LaValle. Una matriz de transformación homogénea está compuesta de una matriz 3x3 de rotación y una matriz 3x1 de translación (Yavuz, 2009), como se ve en la figura 3.10.

$$\begin{array}{ccc|c}
 & r_1 & r_2 & r_3 & \Delta x \\
 \text{3x3 matriz de} & & & & \\
 \text{rotación} & r_4 & r_5 & r_6 & \Delta y \quad \text{3x1 traslación} \\
 & r_7 & r_8 & r_9 & \Delta z \\
 \text{1x3 perspectiva} & \underline{0} & \underline{0} & \underline{0} & \underline{1} \quad \text{Escala global}
 \end{array}$$

Figura 3.10: Matriz de transformación homogénea

Para determinar la localización exacta de cada articulación se usará matrices de transformación homogénea que describen la posición y orientación, usando ángulos de Euler, de los puntos de referencia de los actuadores en cada articulación con respecto al anterior punto en la cadena cinemática. La forma de la matriz homogénea de acuerdo a LaValle (LaValle, 2006) es la siguiente:

$${}^{i-1}T_i = \begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & x_t \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & y_t \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma & z_t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Donde usando ángulos de Euler,  $\alpha$  (ángulo de yaw) es el ángulo entre el eje x y la línea de nodos,  $\beta$  (ángulo de pitch) es el ángulo de rotación entre el eje z y el siguiente eje Z, y  $\gamma$  (ángulo de roll) es el ángulo entre la línea de nodos y el eje X (Weisstein, 2010). Además  $x_t$ ,  $y_t$  y  $z_t$  es la distancia entre dos puntos contiguos en los diferentes ejes (LaValle, 2006).

La localización de la cadera con respecto al pecho está dada por la siguiente matriz homogénea:

$${}^P T_C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -89.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Las matrices presentadas a continuación serán la base para la cinemática directa del robot.

### 3.2.1.1 Brazo Derecho

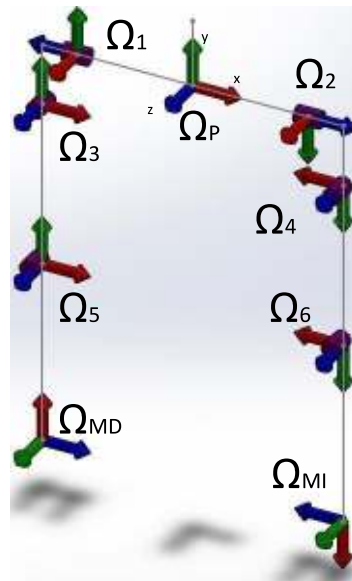


Figura 3.11: Puntos de Referencia del torso

En la figura 3.11 se puede observar los puntos de referencia en las articulaciones del brazo derecho e izquierdo que poseen actuadores, así como también los puntos de referencia del pecho, mano derecha y mano



izquierda. Los puntos de referencia de las articulaciones se nombran de igual manera que los mencionados anteriormente.

Para los brazos el punto de referencia del pecho es considerado como el punto base. Como ya se mencionó anteriormente,  $\beta$  es el ángulo de rotación, por lo tanto lo sustituiremos por nuestro ángulo  $\theta_i$ . A continuación se mostrará el cálculo de la matriz de transformación homogénea del punto de referencia  $\Omega_1$  que corresponde al servo motor  $\theta_1$  con respecto al punto  $\Omega_P$ , que corresponde al pecho.

$${}^P T_1 = \begin{pmatrix} \cos \alpha \cos \theta_1 & \cos \alpha \sin \theta_1 \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \theta_1 \cos \gamma + \sin \alpha \sin \gamma & x_t \\ \sin \alpha \cos \theta_1 & \sin \alpha \sin \theta_1 \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \theta_1 \cos \gamma - \cos \alpha \sin \gamma & y_t \\ -\sin \theta_1 & \cos \theta_1 \sin \gamma & \cos \theta_1 \cos \gamma & z_t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^P T_1 = \begin{pmatrix} \cos 90 \cos \theta_1 & \cos 90 \sin \theta_1 \sin(-90) - \sin 90 \cos(-90) & \cos 90 \sin 90 \cos(-90) + \sin 90 \sin(-90) & -d0 \\ \sin 90 \cos \theta_1 & \sin 90 \sin \theta_1 \sin(-90) + \cos 90 \cos(-90) & \sin 90 \sin \theta_1 \cos(-90) - \cos 90 \sin(-90) & 0 \\ -\sin \theta_1 & \cos \theta_1 \sin(-90) & \cos \theta_1 \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

El resultado sería:

$${}^P T_1 = \begin{pmatrix} 0 & 0 & -1 & -d0 \\ \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Las matrices homogéneas que definen la localización de las otras articulaciones del brazo derecho en la cadena cinemática son las siguientes.

$${}^1T_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -\cos(\theta_3) & -\text{sen}(\theta_3) & 0 & -d1 \\ \text{sen}(\theta_3) & \cos(\theta_3) & 0 & -d2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^3T_5 = \begin{pmatrix} \cos(\theta_5) & \text{sen}(\theta_5) & 0 & 0 \\ -\text{sen}(\theta_5) & \cos(\theta_5) & 0 & -d3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^5T_{MD} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -d4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### 3.2.1.2 Brazo Izquierdo

Para el brazo izquierdo, la cadena cinemática está definida por las siguientes matrices de transformación homogéneas.

$${}^P T_2 = \begin{pmatrix} 0 & 0 & 1 & d0 \\ \cos(\theta_2) & \text{sen}(\theta_2) & 0 & 0 \\ \text{sen}(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^2T_4 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ -\cos(\theta_4) & -\text{sen}(\theta_4) & 0 & d1 \\ -\text{sen}(\theta_4) & \cos(\theta_4) & 0 & -d2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^4T_6 = \begin{pmatrix} -\cos(\theta_6) & -\text{sen}(\theta_6) & 0 & 0 \\ \text{sen}(\theta_6) & \cos(\theta_6) & 0 & -d3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^6T_{MI} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -d4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### 3.2.1.3 Pierna Derecha

Para las piernas, el punto de referencia base es  $\Omega_C$  en la cadera como se muestra en la figura 3.12.

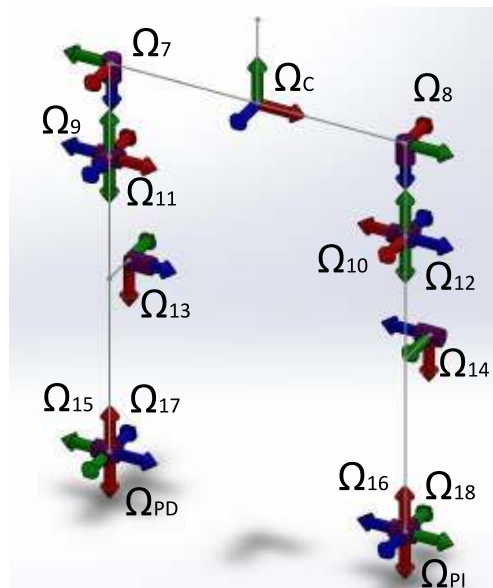


Figura 3.12: Puntos de referencia parte inferior

Cada pierna tiene seis puntos de referencia por lo que van a haber matrices de transformación homogénea para especificar la cadena cinemática de cada pierna. Para la pierna derecha las matrices son las siguientes.

$${}^c T_7 = \begin{pmatrix} -\cos(\theta_7) & \text{sen}(\theta_7) & 1 & -d5 \\ 0 & 0 & -1 & 0 \\ -\text{sen}(\theta_7) & -\cos(\theta_7) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^7 T_9 = \begin{pmatrix} \frac{\sqrt{2}}{2} \cos(\theta_9) & \frac{\sqrt{2}}{2} \text{sen}(\theta_9) & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} \cos(\theta_9) & \frac{\sqrt{2}}{2} \text{sen}(\theta_9) & \frac{\sqrt{2}}{2} & 0 \\ \text{sen}(\theta_9) & -\cos(\theta_9) & 1 & d6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^9T_{11} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ \cos(\theta_{11}) & -\text{sen}(\theta_{11}) & 0 & 0 \\ -\text{sen}(\theta_{11}) & -\cos(\theta_{11}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^{11}T_{13} = \begin{pmatrix} -\cos(\theta_{13}) & -\text{sen}(\theta_{13}) & 0 & -d7 \\ \text{sen}(\theta_{13}) & -\cos(\theta_{13}) & 0 & -d8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{13}T_{15} = \begin{pmatrix} -\text{sen}(\theta_{15}) & \cos(\theta_{15}) & 0 & d7 \\ -\cos(\theta_{15}) & -\text{sen}(\theta_{15}) & 0 & -d8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^{15}T_{17} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ \cos(\theta_{17}) & -\text{sen}(\theta_{17}) & 0 & 0 \\ -\text{sen}(\theta_{17}) & -\cos(\theta_{17}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{17}T_{PD} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### 3.2.1.4 Pierna Izquierda

Para la pierna izquierda las matrices de transformación homogénea son las siguientes.

$${}^cT_8 = \begin{pmatrix} \cos(\theta_8) & -\text{sen}(\theta_8) & 0 & d5 \\ 0 & 0 & -1 & 0 \\ \text{sen}(\theta_8) & \cos(\theta_8) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^8T_{10} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \cos(\theta_{10}) & -\frac{\sqrt{2}}{2} \text{sen}(\theta_{10}) & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} \cos(\theta_{10}) & \frac{\sqrt{2}}{2} \text{sen}(\theta_{10}) & -\frac{\sqrt{2}}{2} & 0 \\ \text{sen}(\theta_{10}) & -\cos(\theta_{10}) & 1 & -d6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{10}T_{12} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \cos(\theta_{12}) & -\sin(\theta_{12}) & 0 & 0 \\ \sin(\theta_{12}) & \cos(\theta_{12}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^{12}T_{14} = \begin{pmatrix} \cos(\theta_{14}) & \sin(\theta_{14}) & 0 & -d7 \\ -\sin(\theta_{13}) & \cos(\theta_{13}) & 0 & d8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{14}T_{16} = \begin{pmatrix} \sin(\theta_{16}) & -\cos(\theta_{16}) & 0 & -d7 \\ \cos(\theta_{16}) & \sin(\theta_{15}) & 0 & -d8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^{16}T_{18} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ \cos(\theta_{18}) & -\sin(\theta_{18}) & 0 & 0 \\ \sin(\theta_{18}) & \cos(\theta_{18}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{18}T_{PI} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### 3.2.1.5 Modelado Cinemático

Para obtener el modelado directo de las extremidades del robot, se necesita multiplicar las matrices de la siguiente manera:

$${}^P T_{MD} = {}^P T_1 \cdot {}^1 T_3 \cdot {}^3 T_5 \cdot {}^5 T_{MD}$$

$${}^P T_{MI} = {}^P T_2 \cdot {}^2 T_4 \cdot {}^4 T_6 \cdot {}^6 T_{MI}$$

$${}^C T_{PD} = {}^C T_7 \cdot {}^7 T_9 \cdot {}^9 T_{11} \cdot {}^{11} T_{13} \cdot {}^{13} T_{15} \cdot {}^{15} T_{17} \cdot {}^{17} T_{PD}$$

$${}^C T_{PI} = {}^C T_8 \cdot {}^8 T_{10} \cdot {}^{10} T_{12} \cdot {}^{12} T_{14} \cdot {}^{14} T_{16} \cdot {}^{16} T_{18} \cdot {}^{18} T_{PI}$$

Donde  ${}^P T_{MD}$ ,  ${}^P T_{MI}$ ,  ${}^C T_{PD}$  y  ${}^C T_{PI}$  son las matrices de transformación homogénea del brazo derecho, brazo izquierdo, pierna derecha y pierna izquierda respectivamente.

El cálculo de estas multiplicaciones se realizó con la ayuda del software *Mathematica 9* de *Wolfram Alpha* con los siguientes valores de distancias:  $d_0=47$  mm,  $d_1=14.5$  mm,  $d_2=25$  mm,  $d_3=67.5$  mm,  $d_4=74.5$  mm,  $d_5=38.5$  mm,  $d_6=31$  mm,  $d_7=75$  mm y  $d_8=14.5$  mm. El resultado para el brazo derecho es el siguiente:

$${}^{P_T}_{MD} = \begin{pmatrix} C_3C_5 - S_3S_5 & 0 & -C_5S_3 - C_3S_5 & -67.5C_3 - 74.5(C_3C_5 - S_3S_5) - 22 \\ C_3S_1S_3 + C_3S_1S_5 & C_1 & C_3C_5S_1 - S_3S_5S_1 & -67.5S_3S_1 + 14.5S_1 - 74.5(C_3S_1S_3 + C_3S_1S_5) \\ C_1C_5S_3 + C_1C_3S_5 & -S_1 & C_1C_3C_5 - C_1S_3S_5 & -67.5S_3C_1 + 14.5C_1 - 74.5(C_1C_5S_3 + C_1C_3S_5) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Donde  $S_i$  representa el seno de  $\theta_i$  y  $C_i$  representa el coseno de  $\theta_i$ . Para el brazo izquierdo la matriz es la siguiente:

$${}^{P_T}_{MI} = \begin{pmatrix} -C_4C_6 + S_4S_6 & 0 & C_6S_4 - C_4S_6 & 67.5C_4 - 74.5(-C_4C_6 + S_4S_6) + 22 \\ -C_6S_2S_4 + C_4S_2S_6 & -C_2 & C_4C_6S_2 - S_4S_6S_2 & 67.5S_4S_2 + 14.5S_2 - 74.5(-C_6S_2S_4 + C_4S_2S_6) \\ -C_2C_6S_4 + C_2C_4S_6 & -S_2 & C_2C_4C_6 - C_2S_4S_6 & 67.5S_4C_2 + 14.5C_2 - 74.5(-C_2C_6S_4 + C_2C_4S_6) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En el caso de las piernas las matrices son demasiado grandes, por lo que se adjuntarán en el Anexo A en forma de texto.

### 3.2.2 ANÁLISIS DE DEFORMACIONES

Con la ayuda del software Solidworks se va a ejecutar un análisis de deformaciones en 4 de las principales tipos de pieza que conforma el robot. Se eligió estas 4 piezas ya que son las que generalmente sostienen los actuadores en la parte en la que estos se mueven.

### 3.2.2.1 Pieza FP04-F1

Esta pieza es la que sostiene cada brazo. Se configuró el software para que la carga sea de 0.6 Kg que se puede considerar como el peso de cada brazo. El resultado del análisis se ve en la figura 3.13.

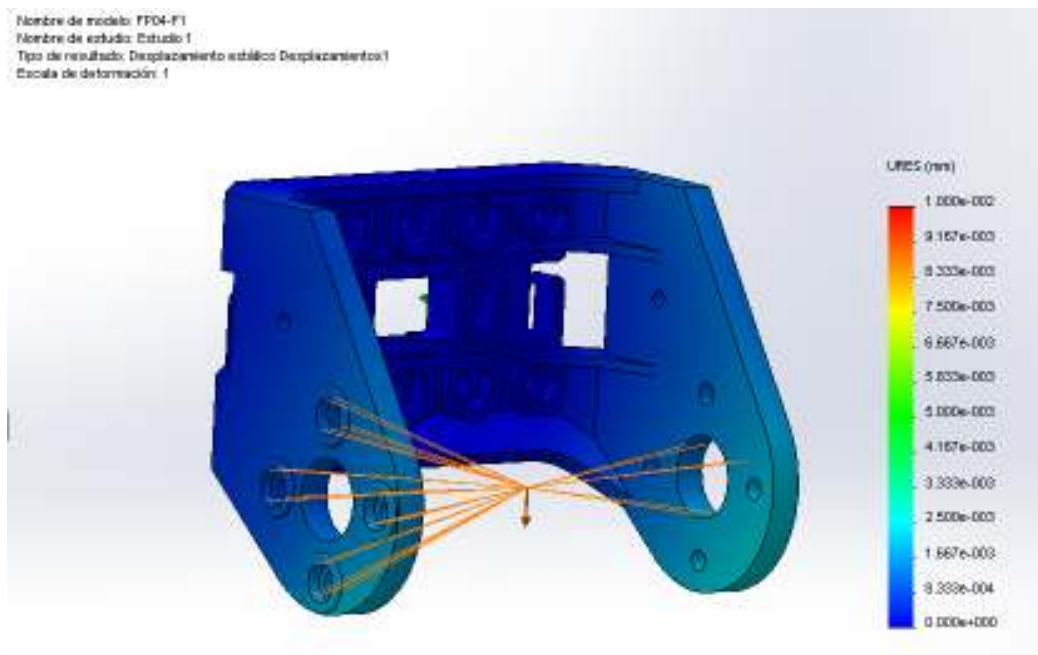


Figura 3.13: Análisis de la pieza FP04-F1

La figura 3.13 tiene una deformación máxima que es  $1.681 \times 10^{-3}$  mm o  $1.681 \mu\text{m}$ , la cual es una deformación despreciable con respecto a la pieza.

### 3.2.2.2 Pieza FP04-F2

La pieza FP04-F2 está colocada en los brazos del robot, en la figura 3.14 se ve la deformación de la pieza a una escala de 20609.2 sometida a una carga de 0.6 Kg. La deformación máxima es de  $2.37 \times 10^{-4}$  mm o  $0.237 \mu\text{m}$  que es despreciable con respecto a la pieza.

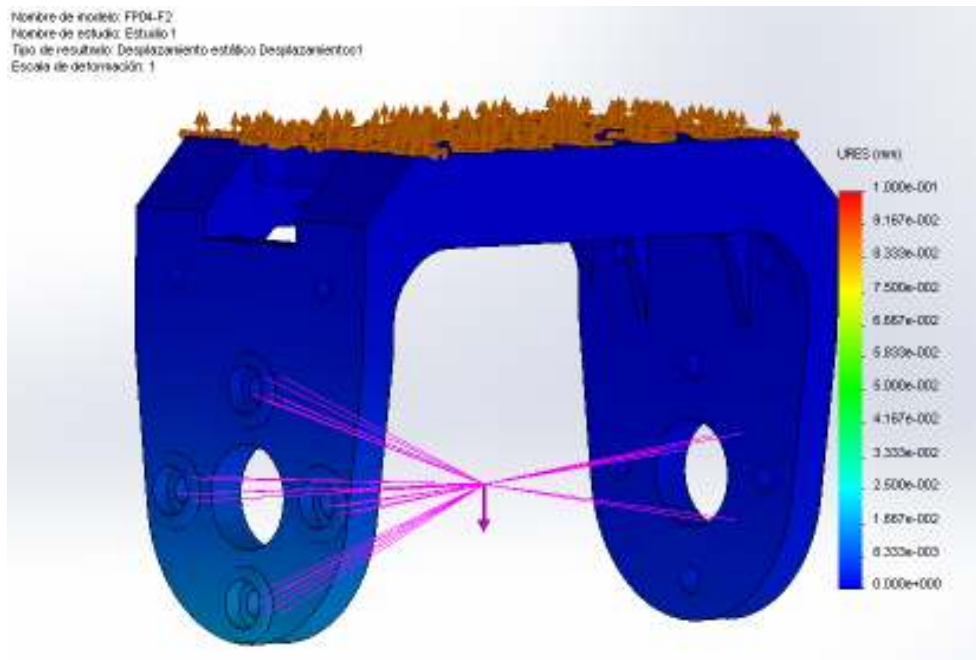


Figura 3.14: Análisis de la pieza FP04-F2

### 3.2.2.3 Pieza FP04-F4

En la figura 3.15 se tiene la deformación de la pieza denominada FP04-F4 a una escala de 7209.48. Esta pieza va colocada en las piernas del robot. Para el estudio se usó una carga de 1.7 Kg que es el peso total del robot, pese a esto la deformación máxima de la pieza es de  $1.066 \times 10^{-3}$  mm o 1.066  $\mu\text{m}$  q es despreciable con respecto a la pieza.



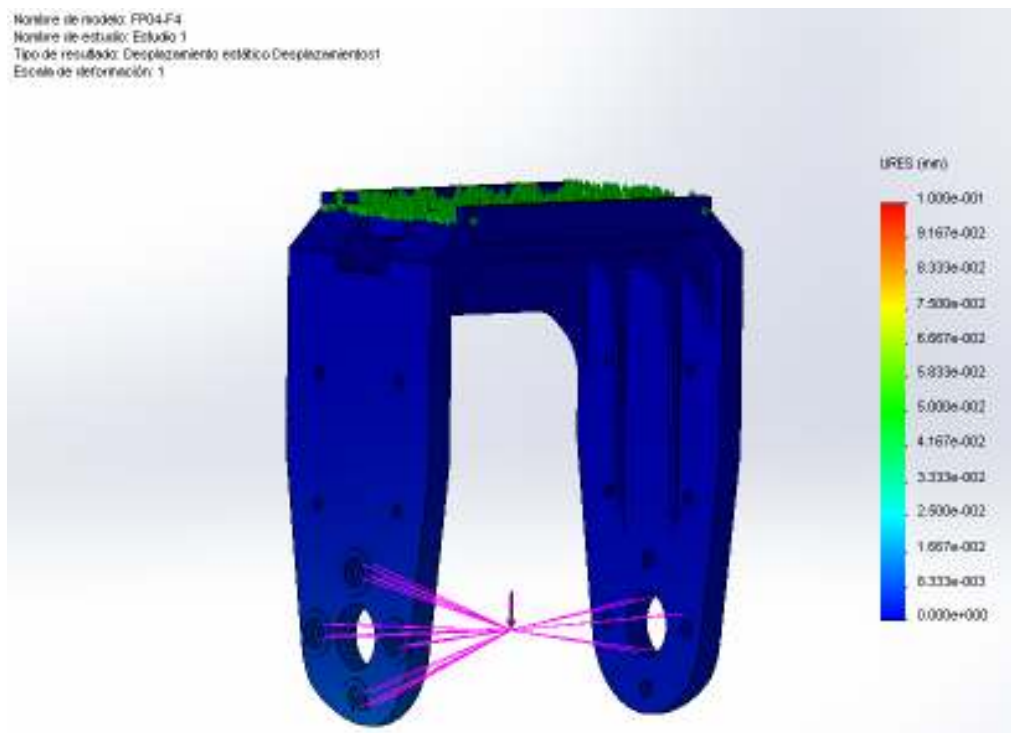


Figura 3.15: Análisis de la pieza FP04-F4

#### 3.2.2.4 Pieza FP04-F5

Por último se realizó el análisis de la pieza FP04-F5, de la cual van colocadas 2 en cada pierna. Se usó una carga de 0.85 Kg para el estudio, cuyo resultado se puede ver en la figura 3.16 a una escala de 142.446. La máxima deformación en esta pieza es de  $5.711 \times 10^{-2}$  mm o 57.11  $\mu$ m. Pese a que su deformación con respecto a la pieza es despreciable, se puede notar que se deforma más que las otras piezas que se analizó, esto es debido a que su longitud es mayor a las otras piezas.

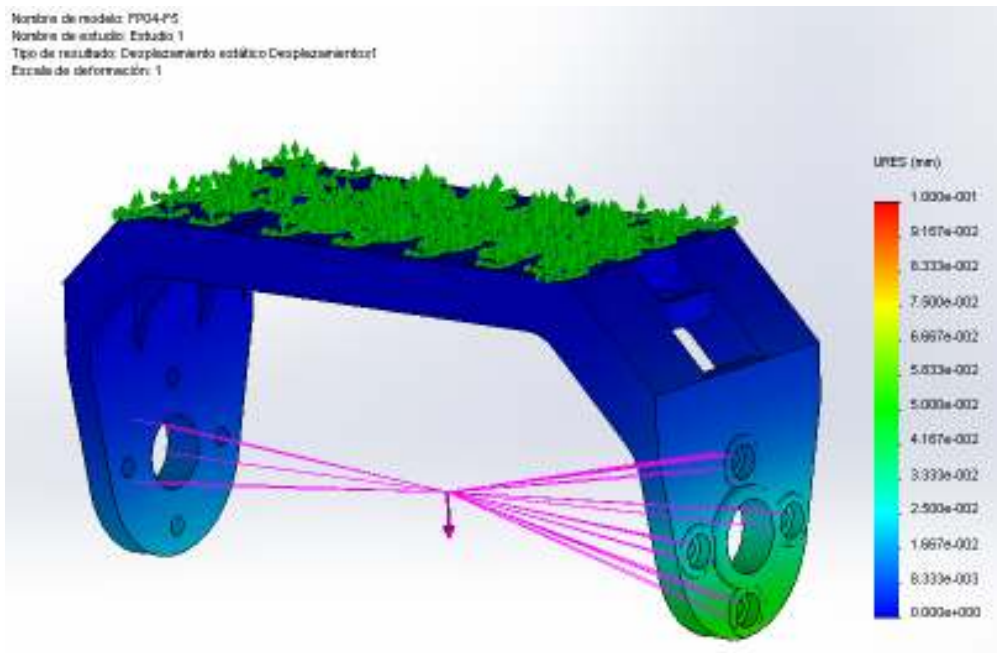


Figura 3.16: Análisis de la pieza FP04-F5

## CAPÍTULO 4

### CONTROL DEL SISTEMA

En este capítulo se explicará el diseño e implementación de la programación las cuales están basadas en el trabajo requerido. En la parte de diseño se especifica el trabajo requerido, además se tiene los tipos de control de robótica colaborativa que se utiliza en el proyecto y la arquitectura que este utiliza. En la implementación se tiene la programación embebida de cada robot y la programación de la unidad central de procesamiento que da las órdenes a los robots.

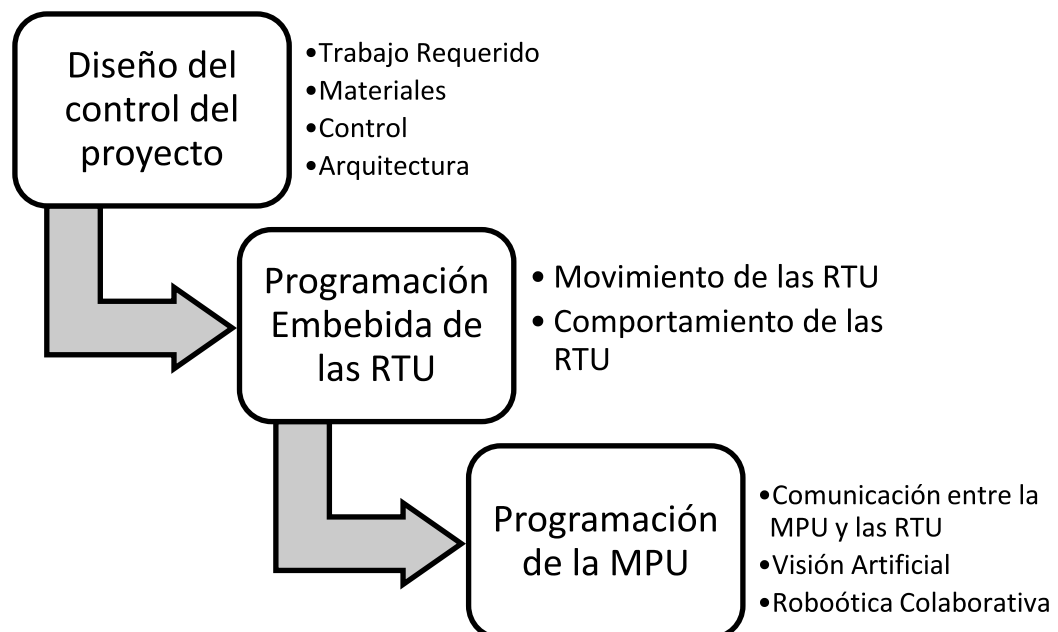


Figura 4.1: Organización del Capítulo 4

## 4.1 DISEÑO

Es necesario primero establecer un diseño para el programa que se usará en las RTU y en la MPU. Para poder diseñar un sistema que satisfaga el trabajo que se quiere lograr con el proyecto es necesario tomar en cuenta los materiales con los que se dispone para el proyecto, así como también las limitaciones que nos da estos materiales.

### 4.1.1 TRABAJO REQUERIDO

La meta final de este proyecto consiste en el uso de dos robots (RTU) que se pasen una pelota de un lado a otro lado de una superficie de trabajo, de 2 m por 1.5 m, usando una unidad central de procesamiento (MPU).

Es necesario definir los lados de la superficie de trabajo en las que se pueda considerar que los robots están cumpliendo su objetivo al pasar la pelota. Las formas de dividir el campo de trabajo se pueden ver en la Figura 4.2.

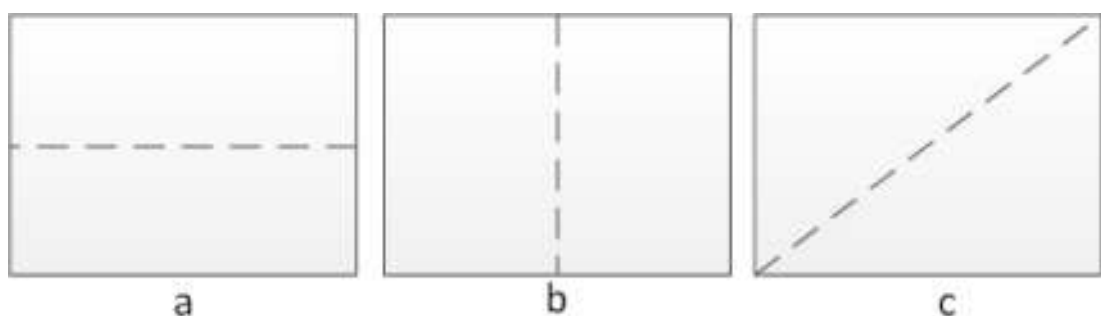


Figura 4.2: Formas de división del espacio

Existen 3 formas de dividir el espacio de trabajo de manera que las partes divididas tengan la misma área. Para seleccionar una de las tres tomaremos en cuenta la máxima distancia que tendría que recorrer el robot en cada parte. Para la forma (a) se tiene que la mayor distancia es la diagonal de la parte dividida, en este caso es de 2.13 m. Para la forma (b) la mayor distancia también la forma la diagonal de la parte dividida, en este caso es de 1.80 metros. Para la forma (c) se tiene que la mayor distancia sería la misma división, en este caso sería de 2.50 m. Por esta razón se va a usar la forma (b) para la división del espacio de trabajo entre la cual debe pasar la pelota. A cada uno de estos espacios se le denominará como “Espacio Dividido de Trabajo” como se ve en la Figura 4.3.



Figura 4.3: Espacios Divididos de Trabajo

#### 4.1.2 MATERIALES

Los materiales con los que se cuenta para el proyecto son::

- Dos kits de Robots Humanoides con 18 grados de libertad cada uno y diferentes sensores.
- Computadora Personal.
- Módulos de comunicación Zigbee.
- Cámara web de alta definición con conexión USB.
- Espacio de 2 metros por 1.5 metros para trabajar.
- Pelota.

#### **4.1.2.1 Limitaciones de los Materiales**

Cada uno de los materiales, excepto la pelota, tienen limitaciones que pueden reducir la capacidad del trabajo final si no son bien identificadas. Las limitaciones de cada material son:

##### **▪ Robots**

La principal limitación de los robots humanoides Bioloid Premium es la falta de sensores que capturen datos del medio en el que están. Con tan solo sensores básicos, estos robots no pueden captar datos necesarios para ejercer el trabajo requerido, por sí solos. Es necesario el uso de un elemento externo a los robots para poseer más datos y ejecutar el trabajo deseado.

##### **▪ Computadora Personal**

La computadora personal para el proyecto posee características normales que se pueden encontrar en una computadora doméstica con sus respectivas limitaciones de procesamiento. Esta computadora cuenta con

puertos USB para manejar la cámara web y con dos puertos seriales para manejar las comunicaciones con los robots.

- **Módulos de Comunicación Zigbee**

Los módulos de comunicación inalámbrica Zigbee están basados en el estándar de comunicación IEEE 802.15.4 de redes de área personal de baja tasa de transmisión de datos. Este tipo de comunicación posee niveles bajos de alcance y la tasa de transmisión de datos máxima es de 200 kbps cuando la conexión es óptima. Debido a la necesidad de que se transmitan y reciban datos continuamente a través de estos módulos, es necesario que cada módulo se conecte a un puerto serial en la computadora.

- **Cámara web**

La cámara que se posee tiene una resolución máxima de 1280 píxeles por 960 píxeles. Tiene un puerto USB 2.0 que sólo puede ser conectada a la computadora personal ya que los controladores de los robots poseen limitaciones de hardware y procesamiento lo cual no permite una conexión directa con esta cámara.

- **Espacio**

Los dos robots tienen que trabajar en un espacio de 2 metros por 1.5 metros para realizar el espacio requerido. Tomando en cuenta el ancho de los robots de 21 cm aproximadamente se tiene que un robot si se lo toma como circunferencia tiene un área de  $0.139 \text{ m}^2$ . El área del espacio de trabajo es de  $3 \text{ m}^2$ . Con esto se tiene que la relación del robot con respecto

al área de trabajo es de 1 a 21 aproximadamente. Esto quiere decir que el espacio de trabajo daría cabida a máximo 21 robots.

### **4.1.3 CONTROL**

#### **4.1.3.1 Visión en Robótica Colectiva en Enjambre**

Uno de los principales estudios de la robótica colectiva son los sistemas de robots en enjambre. Este modo de control de robots utiliza varios robots simples para ejecutar una acción deseada. Una manera de controlar los robots en enjambre consiste en usar una cámara sobre el sistema de robots como es planteado en (Martinoli, 1999). En este documento se plantea que los puntos principales de esta manera de control son las siguientes:

- Los robots no necesitan muchos sensores para captar su entorno.
- Se puede usar cámaras que son de bajo costo.
- Sólo es necesario 1 marcador por robot u objeto que se esté controlando.
- Se tiene buena precisión.
- Se debe tener cuidado con la iluminación.
- No deben existir obstáculos entre la cámara y los marcadores que esta identifica.



En esta investigación se plantea el uso de un enjambre de robots por todo el espacio de trabajo. En cambio en (Kopacek, 2008) ya se plantea el uso de equipos para el juego de fútbol con el uso de microrobots.

Con este tipo de control se resuelve la limitación que se tiene de poseer una cámara que se puede conectar a la MPU, en este caso la computadora personal.

#### **4.1.3.2 Zona de Influencia**

En la Robótica Colectiva se puede hacer divisiones de espacio en donde trabajará un robot. Este espacio es conocido también como Zona de Influencia (Rocher & Duhaut, 1998) del robot que el espacio en el que el robot se mueve y ejecuta sus acciones. La zona de influencia de cada robot para este proyecto es el espacio determinado por la división del campo de trabajo ya que sólo se cuentan con dos robots. Si se contara con más robots, la zona de influencia de cada robot fuera una fracción del “Espacio Dividido de Trabajo”.

#### **4.1.4 ARQUITECTURA**

Tomando en cuenta las limitaciones, se elabora la arquitectura que va a usar este proyecto en base al control con una cámara sobre el campo de trabajo usada en los sistemas de robótica colaborativa en enjambres y a la Zona de Influencia de cada robot.

Los elementos que conforman la arquitectura para este proyecto son un computador de control o MPU que controla a los robots o RTU y maneja la visión artificial que es suministrada por una cámara web a través de una conexión USB. La comunicación de la computadora hacia a las unidades robóticas se la hace a través de los puertos RS232 de la computadora por medio de comunicación inalámbrica Zigbee a los controladores de los robots. Los controladores de los robots a través de su programación embebida manejan los actuadores para ejecutar los movimientos deseados y los sensores de distancia y velocidad angular, para detectar caídas y levantarse. Los datos de los sensores de detección de pelota pasan a través del controlador directo a la MPU donde ésta puede tomar decisiones de acuerdo al sistema de visión artificial. La arquitectura del proyecto se puede ver en la Figura 4.4.

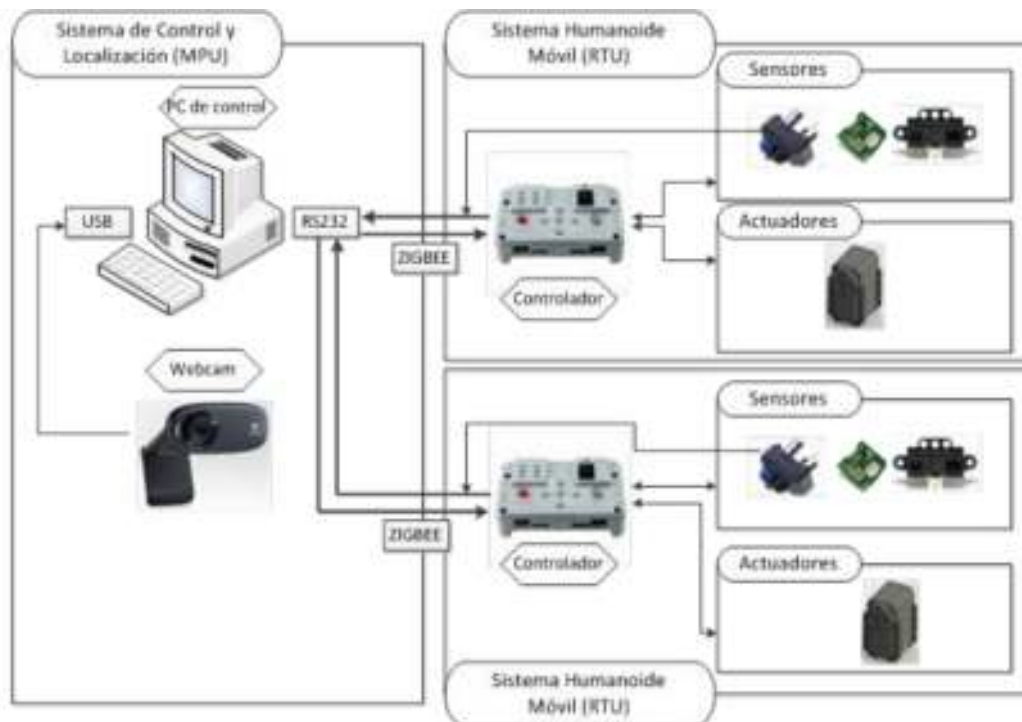


Figura 4.4: Arquitectura general del sistema

## **4.2 PROGRAMACIÓN EMBEBIDA DE LAS RTU**

### **4.2.1 MOVIMIENTO DE LAS RTU**

Para que un robot Bioloid pueda hacer ejercer un movimiento se necesitan dos tipos de programación. En la primera se programan las posiciones y velocidades de los actuadores de todas las poses que hace el robot. En la segunda se programa el comportamiento que tiene el robot con respecto a los datos que recibe del usuario y a los datos que recibe del ambiente que le rodea a través de sus sensores, con estos datos el robot llama a diferentes partes del primer programa y el robot ejecuta un movimiento. Para programar el comportamiento se necesita saber que movimientos tiene cada robot, es por esto que se programa primero estos movimientos.

Este proyecto va a usar código abierto para la programación de los robots, se usará el software Atmel Studio 6 de la compañía Atmel Corporation el cual es gratis y está orientado a la programación de diferentes microcontroladores de esta compañía, en el caso de los robots Bioloid, el ATmega 2561. Como la programación de los movimientos del robot es bastante larga, se decidió usar el software propietario RoboPlus Motion de la compañía que fabrica los robots, Robotis, para luego transformar el programa a código que se pueda usar en Atmel Studio. Este proceso se lo detallará en el siguiente apartado de “Comportamiento de las RTU” de este capítulo.

### 4.2.1.1 Roboplus Motion

El programa Roboplus Motion es desarrollado por la compañía Robotis y es gratis a través del portal web de dicha compañía. Este software permite la programación de los movimientos del robot, la ventana principal se puede ver en la Figura 4.5.

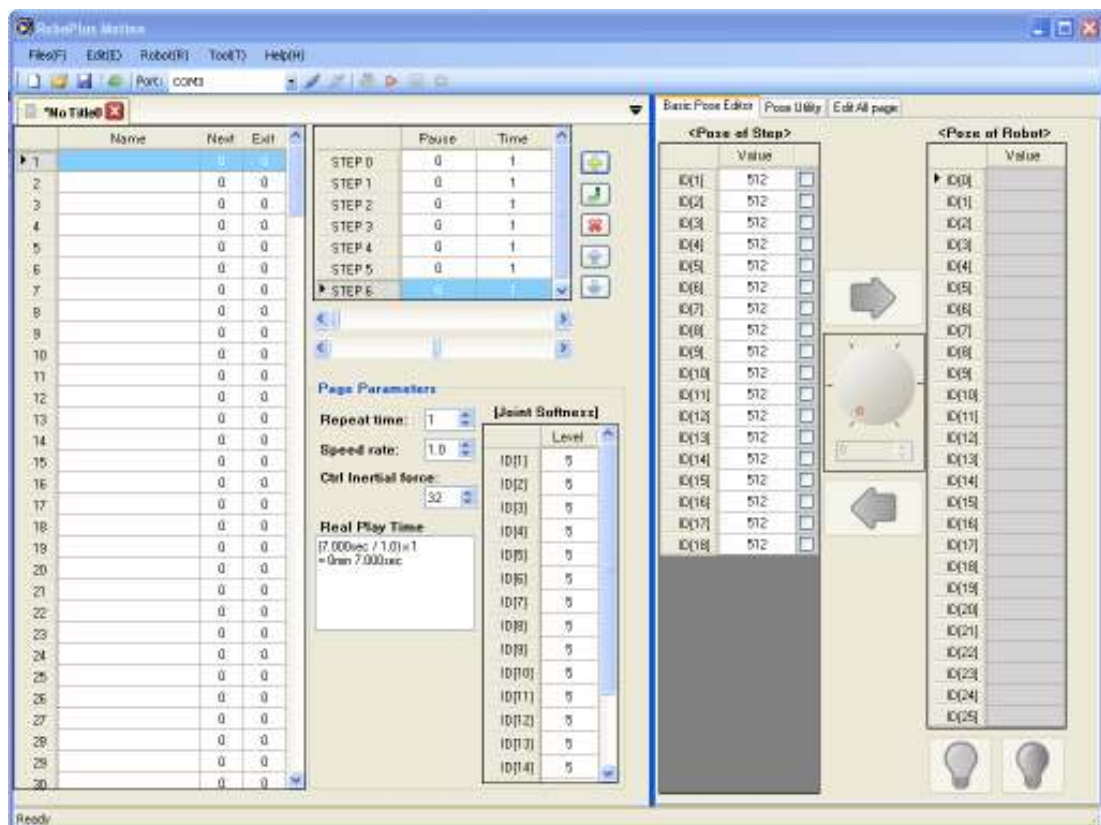


Figura 4.5: Roboplus Motion

En este programa se pueden hacer hasta 255 páginas de movimiento en el controlador CM-510. Esta información se puede guardar en una computadora en un archivo con la extensión “.mtn”. La información que contiene cada página de movimiento se ve en la Figura 4.6.

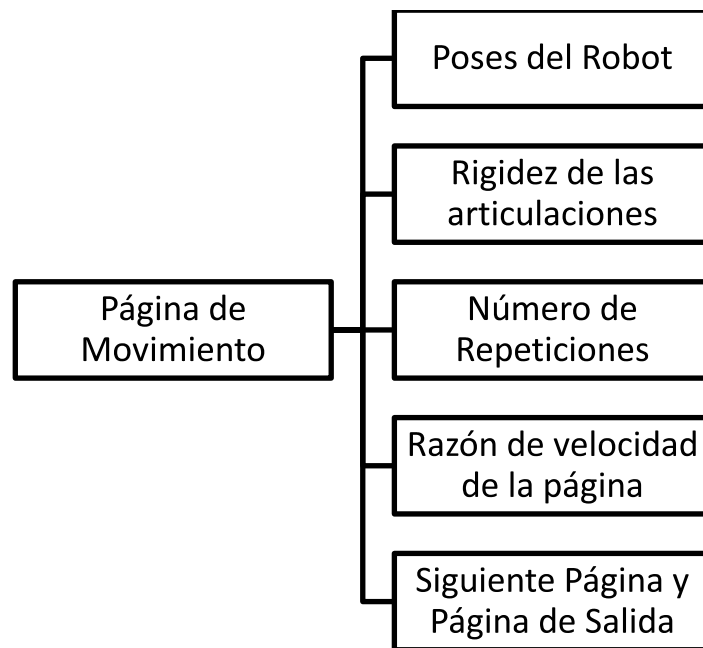


Figura 4.6: Información de las páginas de movimiento

A continuación se detallará cada uno de estos elementos.

- **Poses del Robot**

En cada página de movimiento del robot puede haber hasta 7 poses del robot. Cada pose contiene la información de la posición de cada actuador en dicha pose, además tiene el valor del tiempo total de la pose y la pausa antes que se ejecute la siguiente pose, estos tiempos están en segundos. Las poses en una misma página son consecutivas, es decir, se ejecutan una después de otra. Para poder crear cada pose existe la posibilidad de conectar el robot y cambiar las posiciones de los actuadores para de esta manera grabar estas posiciones. Este es un método que está presente en la programación de diferentes robots educativos o industriales, además ahorra

tiempo al momento de crear movimientos y la programación se vuelve más fácil cuando se tiene muchos actuadores.

- **Rigidez de las Articulaciones**

Este parámetro es usado para determinar cuan rígido se comportan los actuadores en el movimiento. Puede ser ajustado individualmente a cada actuador y tiene valores que van desde el 1 hasta el 7, siendo 1 el más rígido y el 7 el menos rígido. Las ventajas y desventajas de diferentes niveles de rigidez se pueden ver en la Tabla 4.1.

<b>Nivel</b>	<b>Ventaja</b>	<b>Desventaja</b>
<b>Poco Rígido</b>	El movimiento es fluido	Puede ser malo si es que las piernas necesitan bastante soporte.
<b>Muy Rígido</b>	El movimiento es estable. Se usa para movimientos que necesitan ser estables.	El movimiento luce rígido si es que se hacen movimientos fluidos.

Tabla 4.1: Ventajas y desventajas del nivel de rigidez.

- **Número de Repeticiones**

Se puede configurar que una página se repita más de una vez al llamar al movimiento. Cuando se ajusta un número de movimientos mayor a 1, después de la ejecución de la última pose de la página se ejecuta la primera pose otra vez.

- **Razón de velocidad de la página**

La velocidad de la página es un divisor del tiempo total de ejecución de una página. Con este valor se puede modificar el tiempo real de ejecución de una página cuya fórmula es:

$$\textit{Tiempo real de ejecución} = \frac{\textit{Tiempos + Pausas}}{\textit{Razón de Velocidad}} * \textit{Repeticiones}$$

Donde *Tiempos + Pausas* representa la sumatoria de tiempos y pausas de las poses en la página.

- **Siguiente página y página de salida**

Ya que cada página solo puede contener 7 poses, se puede concatenar diferentes páginas para tener más poses en un movimiento continuo. Esto se lo hace configurando el número de página siguiente que uno desee. Además se puede configurar una página que, antes de detenerse, ejecute un movimiento específico cuando se le da un comando de parada al robot, a la que se le conoce como página de salida.

#### 4.2.1.2 Programación de los Movimientos

Conocidas las características del software de programación de movimientos, se procedió a programar el robot Bioloid. Esta programación se la hizo con el método de la grabación de posiciones, es decir que se ajustó el robot a diferentes posiciones y se las guardó para crear diferentes páginas de movimiento. El resultado final de esta programación produjo 231 páginas

de movimiento. Esta programación sirve para los dos robots ya que tienen que ejecutar el mismo tipo de movimientos.

#### **4.2.2 COMPORTAMIENTO DE LAS RTU**

La programación del comportamiento de los robots Bioloid normalmente se lo hace con el software RoboPlus Task, el cual es un software basado en la programación por línea de comandos, pero de una manera bastante simplificada. Este programa es usado normalmente por personas que tienen poca experiencia en la programación. Para los que tienen más experiencia la compañía creadora de los robots distribuye librerías de código abierto que pueden usarse para programar los robots en entornos de programación más avanzados, tales como el software Atmel Studio. Este software permite la programación de microcontroladores de la compañía del mismo nombre, tales como el ATmega 2561 que está dentro del controlador CM-510 del robot, en los lenguajes de programación C, C++ y ensamblador.

Para la programación del robot se usará las librerías conocidas como "Bioloid Control" que se encuentran bajo la licencia de código abierto Commons Creative BY-SA 3.0, la cual permiten el uso, modificación y distribución de las librerías del robot.



#### 4.2.2.1 Movimientos del Robot

Como se mencionó en el primer apartado del capítulo se debe realizar una conversión del archivo de movimientos que tiene la extensión “.mtn”, para esto se usa una herramienta incluida en las librerías que se van a usar, la cual usa un script escrito en el lenguaje de programación Perl para extraer la información del archivo de movimiento y crea un archivo llamado “motion.h” que es una cabecera que se puede incluir a las librerías para llamar a los movimientos. De esta manera los movimientos quedan en forma de estructura como se puede ver en la Figura 4.7.

```
const struct // L kick
{
    const uint8 JointFlexibility[18];
    const uint8 NextPage;
    const uint8 ExitPage;
    const uint8 RepeatTime;
    const uint8 SpeedRate10;
    const uint8 InertialForce;
    const uint8 Steps;
    const uint32 StepValues[6][6];
    const uint16 PauseTime[6];
    const uint16 PlayTime[6];
} MotionPage18 PROGMEM = {
{5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5},
0, 0, 1, 15, 4, 6,
{{987275543, -1173458383, 1486146077, -1853164903, 1205244528, 1691484741},
{987275543, -1173458383, 1486154240, -1769278777, 1205508720, 1439828551},
{987275543, -1173458383, 1486154240, -1769176394, 1204822640, -1773008313},
{987275543, -1173458383, 1486154240, -1769278715, 1205682800, 1255279175},
{987275543, -1173458383, 1486146067, -1853164903, 1205244528, 1691484741},
{987275543, -1173458383, 1498728955, -2130007382, 1008237191, 1578097156}},
{0,0,400,0,0,0}, {1000,400,200,600,1000,1000}
};
```

Figura 4.7: Página de movimiento del robot

Donde se puede notar que está la diferente información que hay en una página de movimiento.

#### 4.2.2.2 Bioloid Control

Esta serie de librerías ayudan a la programación de los robots en código abierto, su objetivo es ofrecer una alternativa a la programación con el software proporcionado por el fabricante de los robots, a continuación se va dar una descripción de las principales librerías para los robots.

- **Global**

En la librería Global.h se ponen las definiciones básicas de hardware para el controlador CM-510, así como también las variables básicas que van a ser usadas por otras librerías.

- **ADC**

La librería “ADC.c” se usa para la lectura de los puertos analógicos en el microprocesador ATmega2561 los cuales van desde el puerto ADC0 hasta el ADC7. El puerto ADC0 se usa para medir el nivel de batería, los puertos ADC1 hasta el ADC6 están asignados a las entradas auxiliares del controlador CM-510 y el puerto ADC7 no se usa. El conversor analógico/digital del microprocesador tiene una resolución de 10 bits, en sistema decimal van desde el 0 hasta el 1023. Esta librería contiene operaciones para la lectura y conversión, cuando se requiera, de los datos en las entradas analógicas y les asigna espacios de memoria para que se puedan usar en otras partes del programa.

Usando esta librería como base se hicieron las siguientes implementaciones:

- Habilitación de los puertos designados a los sensores del robot.
- Ejecución de levantado del robot si se detecta una caída.
- Asignación de valores que proporcionan los sensores a variables en el programa.
- Impresión de los valores de los sensores de los pies para la transmisión que se necesita en la MPU.

La operación básica de esta librería se puede ver en la Figura 4.8.

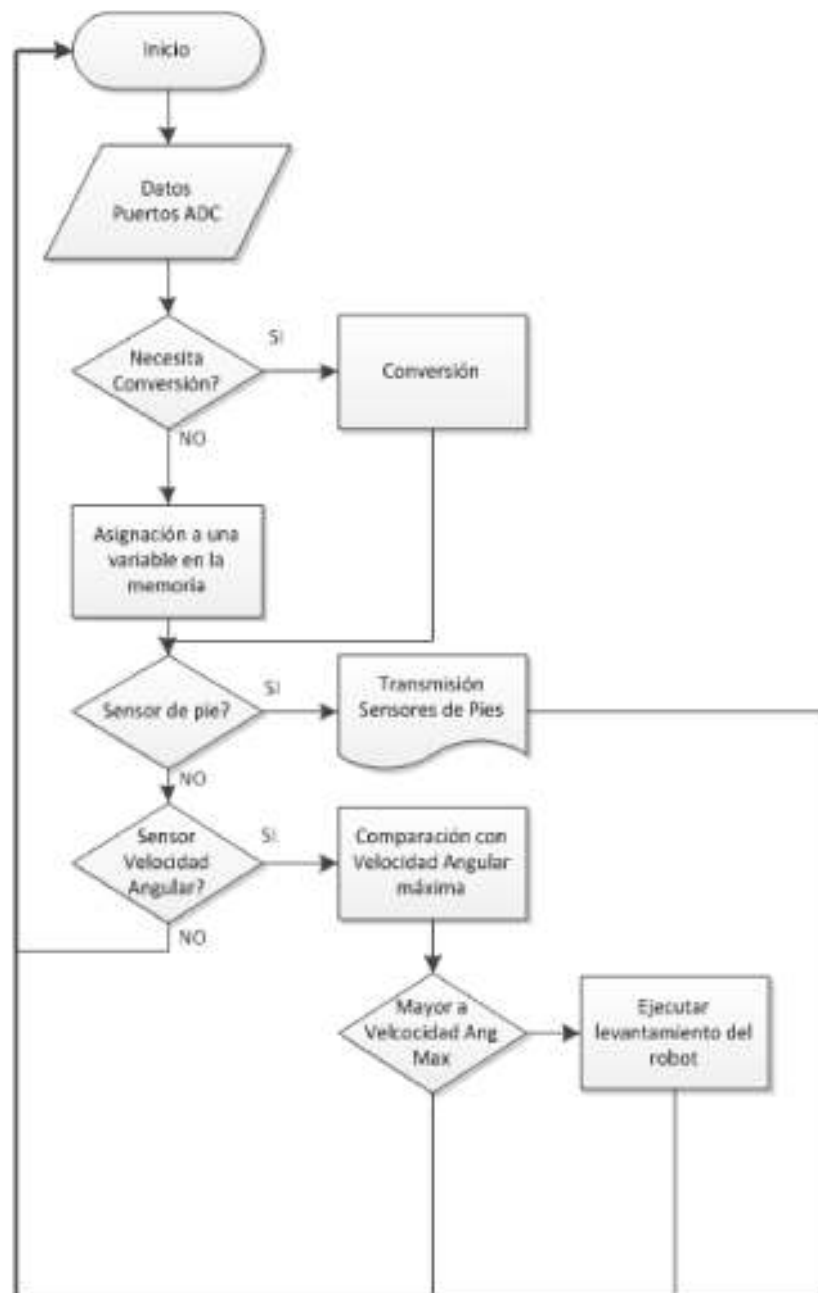


Figura 4.8: Funcionamiento Librería ADC.c

- **Dynamixel**

La librería “Dynamixel.c” contiene las funciones para la transmisión y recepción de datos a los actuadores AX-12A. Esta librería recibe todos los

datos que proveen los sensores como ID, posición, temperatura, entre otras; y envía la posición requerida para que el actuador la ejecute. Los paquetes digitales con los que se comunican los actuadores están conformados como se ve en la Tabla 4.2.

<b>Paquete de Instrucción (hacia el actuador)</b>	
2 Bytes de Inicio	FF FF
ID del actuador	Id (1 byte, FE para broadcast)
Tamaño del Paquete	Número de parámetros + 2
Instrucción	Instrucción (1 byte)
Parámetros	Depende de la instrucción
Suma de verificación	ID+Tamaño+Instr+Parám
<b>Paquete de Estatus (desde el actuador)</b>	
2 Bytes de Inicio	FF FF
Tamaño del Paquete	Número de parámetros + 2
Código de Error	Desde 0 hasta 6
Parámetros	Depende de la instrucción
Suma de verificación	ID+Tamaño+Error+Parámetros

Tabla 4.2: Paquetes de comunicación del AX-12A

- **Motion**

La librería “Motion.c” contiene el código necesario para cambiar la información contenida en la cabecera “Motion.h” en información que pueda ser transmitida a los actuadores a través de la librería “Dynamixel.c”. Además con esta librería se crean definiciones para poder llamar a diferentes movimientos desde otras librerías del programa.

- **Serial**

La librería denominada Serial.c tiene la función de manejar las comunicaciones que entran y salen del controlador por medio del módulo de comunicación Zigbee. El funcionamiento básico de esta librería se puede ver en la Figura 4.9.

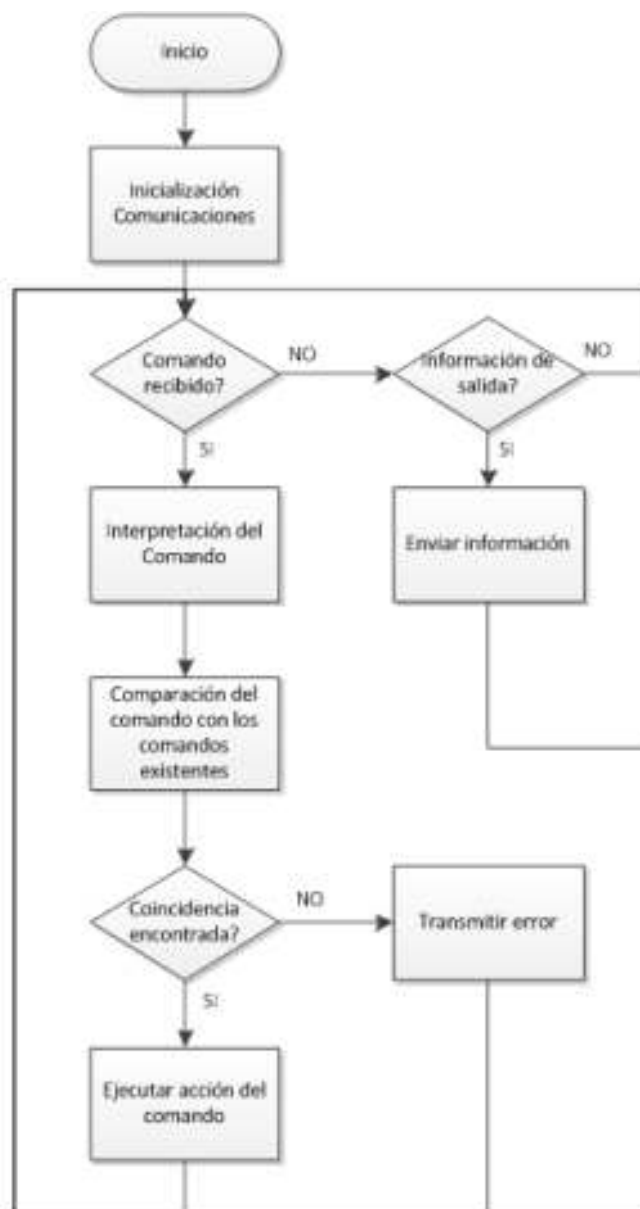


Figura 4.9: Funcionamiento Librería Serial.c

En esta librería se definió una serie de comandos que fueron asociados a diferentes movimientos del robot, para que de esta manera cuando el robot reciba un comando desde la MPU, este ejecute un movimiento. De esta manera es posible el control de las RTU desde la MPU. Los comandos están en la Tabla 4.3.

<b>Comandos</b>	<b>Acción</b>
<b>PARO</b>	Detiene el movimiento
<b>CLST</b>	Inicializa el robot para caminar
<b>CADL</b>	Caminar Adelante
<b>CGIZ</b>	Girar a la izquierda
<b>CGDR</b>	Girar a la derecha
<b>FLVT</b>	Levantarse si está caído de frente
<b>ELVT</b>	Levantarse si está caído de espalda

Tabla 4.3: Comandos de orden al robot

Además se puede enviar comandos para llamar a páginas específicas de movimiento almacenadas en el robot, para esto se debe escribir el comando como por ejemplo “M001”, el cual llama a la página 1 de movimiento del robot. La estructura de este comando es la misma para llamar a cualquier página de movimiento, solo se debe sustituir el número “001” por la página de movimiento que se necesite.

Por último estas librerías principales y otras dentro del conjunto de librerías de “Bioloid Control” se compilan y se construye un archivo en

lenguaje de máquina con la extensión “.hex” la cual debe ser transferida a los robots por medio del programa “Roboplus Terminal”.

### 4.3 PROGRAMACIÓN DE LA MPU

#### 4.3.1 COMUNICACIÓN ENTRE LA MPU Y LAS RTU

Como ya se estableció en el anterior apartado del capítulo, el robot tiene que recibir diferentes comandos para ejecutar las acciones de movimiento. Entonces es necesario que la MPU, que en este caso es una computadora personal, le envíe dichos comandos.

La parte del control de la MPU se la va a trabajar en el software Labview 8.5 de la compañía National Instruments. Este programa va a ser encargado de manejar las comunicaciones, visión artificial y el control de la robótica cooperativa.

##### 4.3.1.1 Conexión Física

Como ya se estableció anteriormente, para la comunicación inalámbrica se usa módulos Zigbee. Estos módulos de comunicación establecen un puente donde se usa la comunicación serial como se muestra en la Figura 4.10.

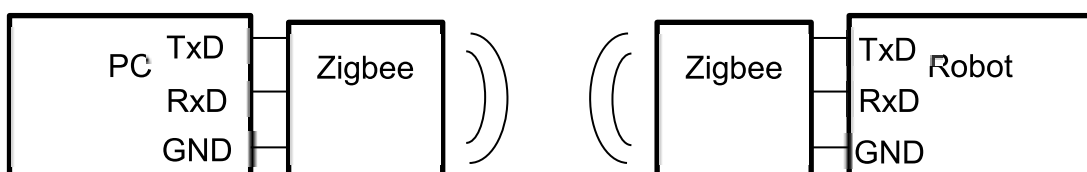


Figura 4.10: Conexión Zigbee



#### 4.3.1.2 Programación de las Comunicaciones

Para empezar la programación se debe tomar en cuenta las diferentes configuraciones que debe tener el puerto de comunicación serial hacia el robot. Estas configuraciones están determinadas por el fabricante y se muestran en la Tabla 4.4

<b>Configuración</b>	<b>Valor</b>
Bits por Segundo	57600
Bits de datos	8
Paridad	Ninguno
Bits de Parada	1
Control de flujo	Ninguno

Tabla 4.4: Configuración del puerto de comunicación

Se usa el módulo VISA<sup>11</sup> Serial del software Labview. Este conjunto de módulos tienen diferentes herramientas para tratar con comunicaciones de tipo Serial. La programación de la comunicación queda determinada por el siguiente diagrama de flujo en la Figura 4.11.

<sup>11</sup> Virtual Instrument Software Architecture

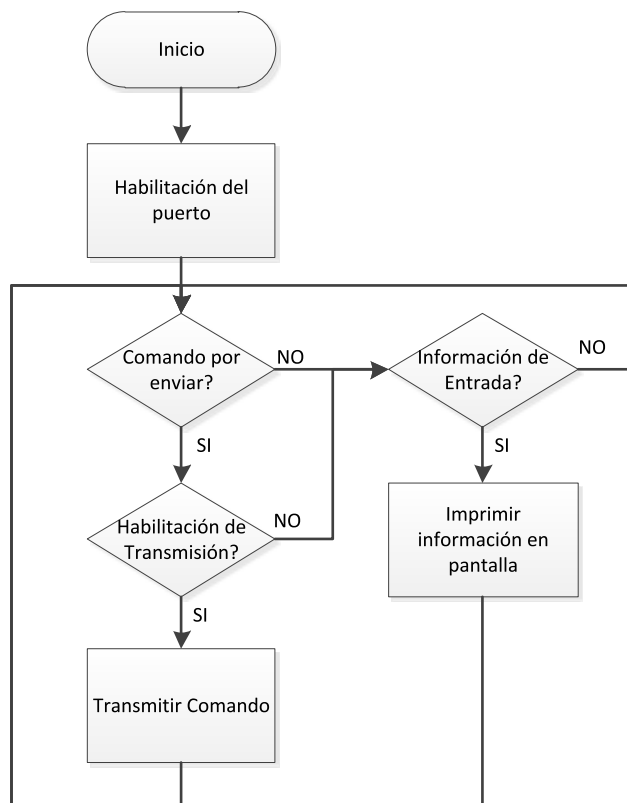


Figura 4.11: Funcionamiento comunicación desde la MPU

Se programó las comunicaciones como un subproceso del programa principal y puede ser llamado cuantas veces se desee siempre y cuando no se use el mismo puerto serial. Esto se hizo debido a que se tiene 2 RTU, las cuales van a ser controladas por diferentes puertos serial debido a que cada robot envía continuamente la información relacionada a sensores hacia la MPU. La programación final del módulo de comunicaciones puede verse en la Figura 4.12.

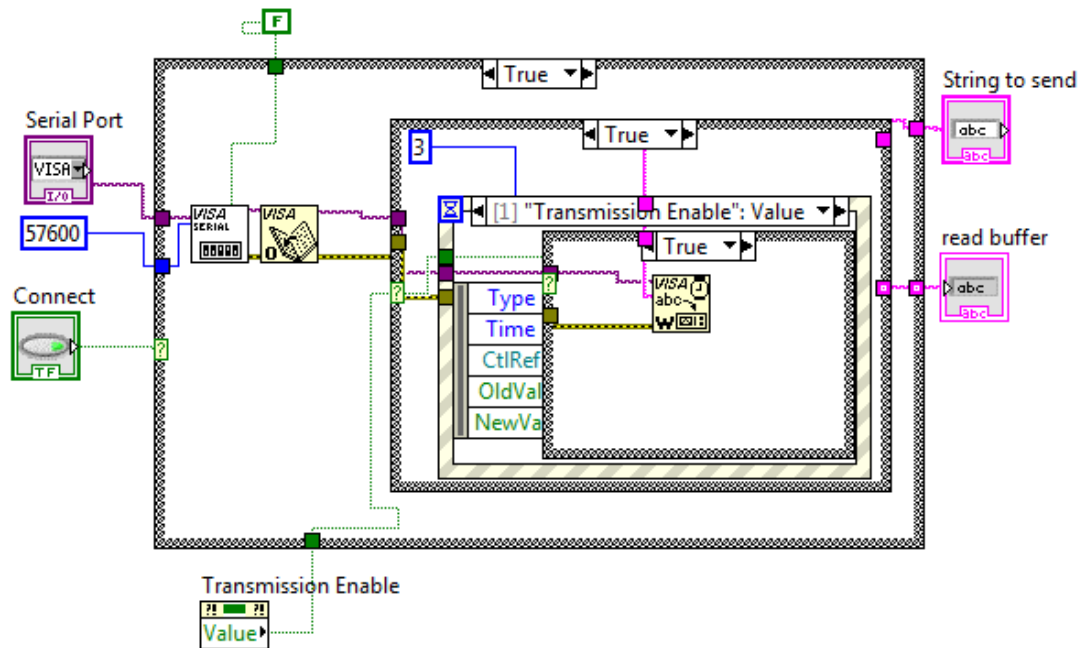


Figura 4.12: Subproceso de comunicación desde la MPU en Labview

Donde “String to Send” es la orden que se le enviará a cada robot, “Read Buffer” es la información que transmite cada robot y que será decodificada después para sacar el valor de los sensores del robot, “Serial Port” sirve para escoger el puerto con el que se va a trabajar, “Connect” sirve para abrir el puerto serial y establecer la comunicación, y “Transmission Enable” sirve para transmitir la orden por el puerto serial al robot y estará dada por el programa que controle a los robots.

### 4.3.2 VISIÓN ARTIFICIAL

Ya que para poder guiar a un robot móvil se requiere conocer su localización en un ambiente determinado, es necesario que la MPU use algoritmos de visión artificial para analizar el área y detectar a los robots así

como también a la pelota. Estos algoritmos necesitan poder detectar la localización y la orientación de los objetos con los que se va a trabajar por lo que la visión artificial es la mejor alternativa si se requiere monitorear estos parámetros en tiempo real.

#### 4.3.2.1 Resolución

Como se mencionó en el anterior capítulo, para la visión artificial se dispone de una cámara web de alta definición. Esta cámara trabaja en diferentes definiciones las cuales están en dos diferentes formatos: el formato 4:3 y el formato 16:9. Estos formatos se refieren a la escala que tiene el largo de la imagen con respecto al alto de la imagen. Dichos formatos se pueden ver en la Figura 4.13.

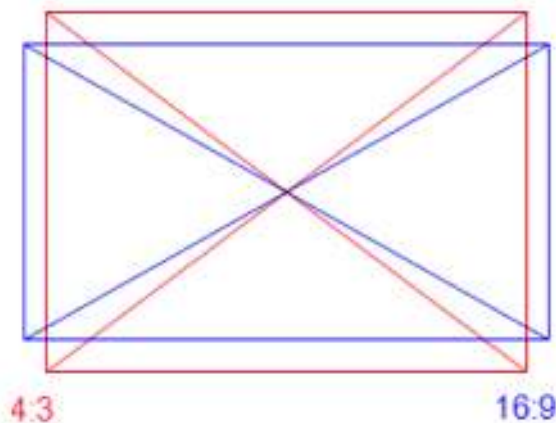


Figura 4.13: Formato 4:3 y 16:9

Para el proyecto se usa el formato 4:3 ya que después de realizar algunas pruebas se determinó que con este formato la cámara capta una mayor área de trabajo que con el formato 16:9. Establecido el formato se procede a seleccionar la resolución con la que se va a trabajar. Para esto se

usó diferentes resoluciones en el módulo de “Vision Acquisition” de Labview para captar diferentes objetos en movimiento. Se usa la resolución máxima de la cámara la cual es de 1280 píxeles de largo por 960 píxeles de alto para tener una mejor resolución en el reconocimiento de los objetos que se van a utilizar.

El origen de coordenadas en una imagen de cualquier cámara conectada a una computadora es la esquina superior izquierda del cuadro. Para este trabajo se cambia este origen para ubicarlo en la esquina inferior izquierda para tener un formato más familiar al momento de la programación.

#### **4.3.2.2 Reconocimiento de Imágenes**

Labview posee diferentes algoritmos que se pueden usar para el reconocimiento de objetos. Para este proyecto se usa el algoritmo de reconocimiento de formas geométricas. Este algoritmo permite que el programa detecte diferentes formas geométricas que se le enseñen por medio de la cámara. Además con este algoritmo se puede obtener información necesaria como la ubicación y orientación de la figura geométrica. Para poder aplicar este algoritmo a la imagen es necesario que esta sea sometida a un proceso de filtrado para quitar la información que no es necesaria, en este caso se quita los colores como se muestra en la Figura 4.14. Para esto se aplicó el algoritmo llamado “Color plane extraction” y con esto se puede aplicar el algoritmo de reconocimiento.



Figura 4.14: Imagen original (izq) e imagen filtrada (der)

Después de tratada la imagen se aplica el reconocimiento de formas geométricas a la pelota y se genera la imagen patrón que se ve en la Figura 4.15.

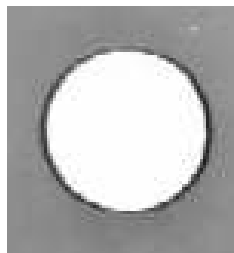


Figura 4.15: Patrón de la pelota

Para cada robot se debe elaborar una forma que servirá como patrón y se la adhiere a la cabeza de cada robot. Estas formas son las mismas para cada robot, para poder trabajar con cada robot la cámara solo reconoce cada robot en su área de trabajo la que se detallará más adelante. Para que el proceso de programación sea fácil, se debe hacer que el programa aprenda el patrón en una orientación que sea la más cercana a  $0^\circ$  como se lo puede ver en la Figura 4.16.



Figura 4.16: Patrón del robot en 0°

Como se pudo apreciar en las figuras de los patrones las imágenes son en escala de grises debido al filtro aplicado. Por esta razón no importa el color de los patrones, en cambio es necesario que estos tengan un buen contraste con respecto a donde están para que la cámara pueda captarlos.

La iluminación de la cámara se ajusta automáticamente a la luz ambiental para que se puede captar una imagen correcta la mayoría del tiempo. Pero si la iluminación ambiental es muy alta o muy baja el sistema de visión no funcionará adecuadamente.

Después de adquiridos los datos de coordenadas y orientación de los patrones de visión artificial, se somete estos datos a dos diferentes algoritmos de Labview que mejoran la operación general del programa final. El primer algoritmo calcula la mediana de estos datos con un factor de reducción de 3, es decir, toma 3 puntos y obtiene la mediana. El segundo algoritmo recolecta los puntos tomados y los muestra de manera que solo sean mostrados los valores si el valor del dato no es 0. Cuando un valor da 0 es porque la cámara perdió la posición del patrón por un momento de manera que el algoritmo muestra el último punto recolectado antes de la pérdida de imagen. De esta manera se ayuda cuando por ejemplo el robot

obstruye la vista de la pelota por parte de la cámara, esta situación se da cuando la pelota se encuentra en los extremos del campo y el robot se dirige a ella.

#### 4.3.2.3 Corrección de Coordenadas

Después de que el programa aprendió los patrones se puede sacar los datos de ubicación y orientación. Debido a la diferencia de alturas entre la ubicación de los patrones de los robot y la pelota, se debe corregir los datos de ubicación de los patrones del robot con respecto a la pelota, ya que como se puede ver en la Figura 4.17, estos se distorsionan si es que se encuentran en diferentes partes con respecto a la cámara.

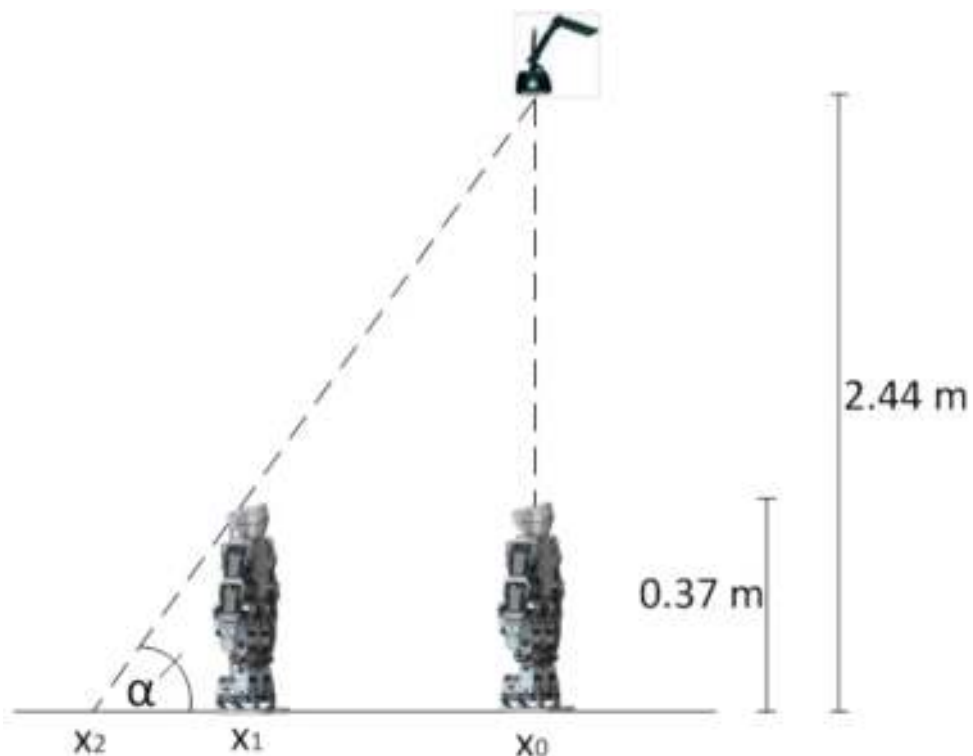


Figura 4.17: Diferentes ubicaciones robot



Se puede apreciar que la cámara ve al objeto proyectado en la posición  $x_2$  pero la verdadera ubicación es  $x_1$ , por esta razón se debe corregir la posición vista por la cámara para que sea la verdadera ubicación del robot

Para poder corregir la ubicación del robot se necesita el valor de la distancia entre  $x_2$  y  $x_1$ , para poder restar de  $x_2$  y obtener la ubicación real. A esta distancia la llamaremos  $x_3$ . Como se puede apreciar en la Figura 4.17 se tiene 2 triángulos semejantes, las alturas fueron acercadas a un valor simple para simplificar el resultado. Usando una de las propiedades de los triángulos semejantes obtenemos que:

$$\tan \alpha = \frac{2.44}{x_2} = \frac{0.37}{x_3}$$

$$x_3 = 0.152x_2$$

Por lo tanto,

$$x_1 = x_2 - x_3 = x_2 - 0.152x_2 = 0.848x_2$$

Con esto se obtiene un resultado aproximado de la ubicación real del robot con respecto a la ubicación que ve la cámara, pero como la ubicación debajo de la cámara no es el centro de coordenadas se escribe un script para que la ubicación sea corregida con respecto al centro de la cámara en los ejes  $x$  y  $y$ .

Con todos estos datos se procede a la última etapa de programación en la que se integra los anteriores pasos.

### 4.3.3 ROBÓTICA COLABORATIVA

En esta etapa además de integrar los anteriores pasos, se crea los algoritmos para que los dos robots trabajen colaborativamente. Primero se debe establecer el modo en el que los robots van a trabajar. Para esto se deberá hacer ciertas consideraciones.

#### 4.3.3.1 Definición de la Zona de Influencia

El trabajo de los robots consiste en pasarse una pelota en el campo de trabajo, este campo está monitoreado por una cámara ubicada a 2 metros sobre este. El campo de trabajo tiene 2 metros de largo por 1.5 metros los cuales estarán representados 1280 píxeles de largo por 960 píxeles de alto. Se delimitó el espacio en el que se mueven los robots a 1.80 m por 1.30 metros ya que la cámara no puede detectar la parte superior de los robots cuando estos están fuera de estos límites debido al ángulo de visión de la cámara.

Debido a la división del área de trabajo mencionada la parte de “Trabajo Requerido” mencionada en este capítulo se divide el espacio en dos sectores para que los robots trabajen, esto se ve en la Figura 4.18.

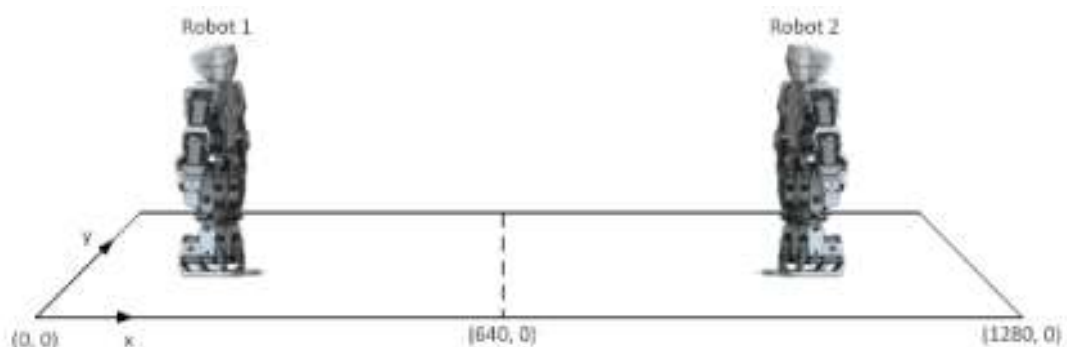


Figura 4.18: Zona de Influencia de cada robot

### 4.3.3.2 Orientación de la Pelota

El algoritmo de reconocimiento de patrones geométricos nos da la posición de la pelota además de las posiciones y orientaciones de los robots, pero este algoritmo no nos da la orientación de la pelota con respecto a los robots, es por esto que se debe escribir un algoritmo que haga este cálculo. Para escribir este algoritmo es necesario saber el ángulo en el que está ubicada la pelota tomando como punto de referencia la ubicación de un robot como se ve en la Figura 4.19.

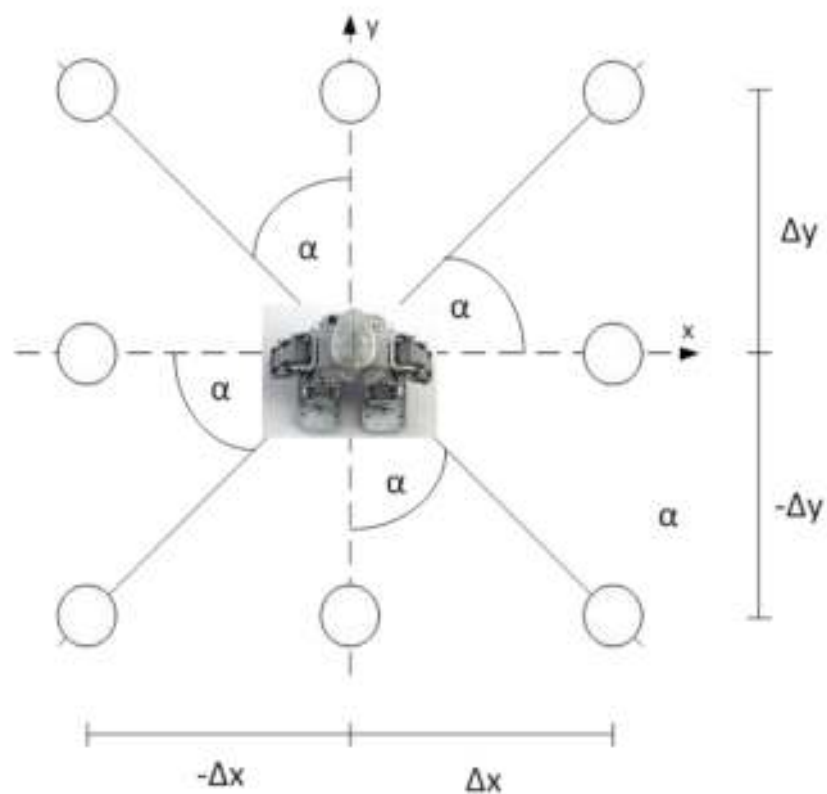


Figura 4.19: Orientación de la pelota con respecto al robot

Para hacer obtener el ángulo necesario se hace el cálculo en cada uno de los cuadrantes con lo que se obtienen las siguientes ecuaciones de cada cuadrante:

Primer cuadrante	$\alpha = \tan^{-1} \frac{\Delta y}{\Delta x}$
Segundo cuadrante	$\alpha = \tan^{-1} \frac{\Delta x}{\Delta y} + 90$
Tercer cuadrante	$\alpha = \tan^{-1} \frac{\Delta y}{\Delta x} + 180$
Cuarto cuadrante	$\alpha = \tan^{-1} \frac{\Delta x}{\Delta y} + 270$

Teniendo estas ecuaciones se crea un subproceso en labview que ejecute el algoritmo de cálculo de la orientación de la pelota con respecto al robot. Además se hace el cálculo de la diferencia de ángulos para poder establecer por cual lado debe girar el robot para ponerse de frente a la pelota de una manera más rápida.

#### 4.3.3.3 Movimientos del Robot

Es necesario incorporar al programa de Labview los comandos de orden que se van a enviar a los robots que se pueden ver en la Tabla 4.3, para esto se crea un subproceso que podrá ser llamado al momento de programar el comportamiento final de los robots. Este subproceso utiliza variables auxiliares que activan un temporizador de tiempo real, el cual envía un impulso de corta duración hacia el subproceso de comunicación con el robot, específicamente, activa la variable "Transmission Enable". Esta programación se debe a que las órdenes de programa deben ser enviadas

solo una vez para que el robot empiece a hacer un movimiento, si el robot recibe muchas órdenes iguales durante un tiempo prolongado, el movimiento no es ejecutado de una manera apropiada.

#### 4.3.3.4 Algoritmo de Movimiento y Acciones

El algoritmo general de movimiento y acciones de los robots tiene que ser programado dependiendo de varios factores que deben ser considerados. Estos factores se los puede representar con diferentes valores en el programa los cuales ayudarán a determinar las distintas acciones del robot. En la Tabla 4.5 se presentan los principales valores junto a una descripción.

Tabla 4.5: Valores principales del programa

Valores	Descripción
<b>xr1, yr1</b>	Coordenadas de la posición del robot 1
<b>xr2, yr2</b>	Coordenadas de la posición del robot 2
<b>xb, yb</b>	Coordenadas de la posición de la pelota
<b>Angle r1</b>	Ángulo de orientación del robot 1
<b>Angle r2</b>	Ángulo de orientación del robot 2
<b>Ang Dif 1</b>	Diferencia entre los ángulos de la pelota y el robot 1
<b>Ang Dif 2</b>	Diferencia entre los ángulos de la pelota y el robot 2
<b>Left r1, Right r1</b>	Valores de los sensores de los pies del robot 1
<b>Left r2, Right r2</b>	Valores de los sensores de los pies del robot 2
<b>Sens</b>	Valor al cual se detecta un objeto en los pies del robot
<b>Half</b>	Valor en X del centro del campo, en este caso Half = 480

<b>MinYb, MaxYb</b>	Valores máximo y mínimo en los que se puede encontrar la pelota en Y para que el robot pueda patear en la dirección deseada
<b>MinXb, MaxXb</b>	Valores máximo y mínimo en los que se puede encontrar la pelota en X para que el robot pueda patear en la dirección deseada
<b>MinDistRB</b>	Mínima distancia a la cual los robots empezarán a moverse diferente para acercarse a la pelota

Tabla 4.5: Valores principales del programa

Algunos valores de la Tabla 4.5 están relacionados al espacio de trabajo como se puede ver en la Figura 4.20.

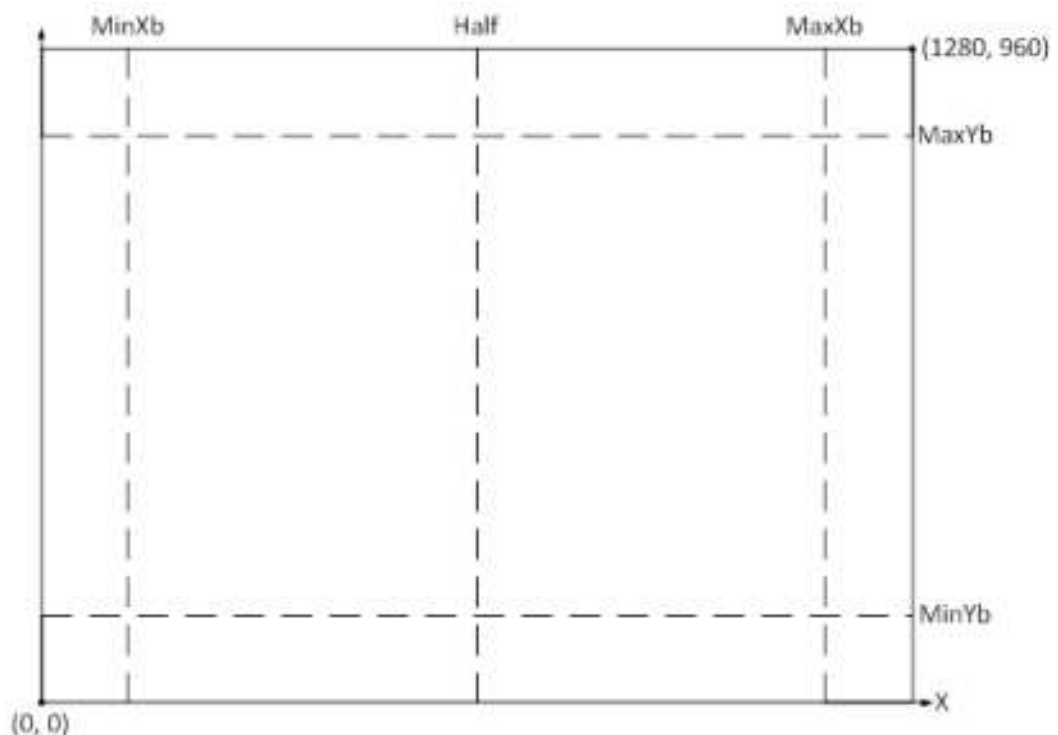


Figura 4.20: Valores en espacio de trabajo

Con estos valores se elabora el programa obedeciendo a los diagramas de flujo de la Figura 4.21 hasta la Figura 4.29.

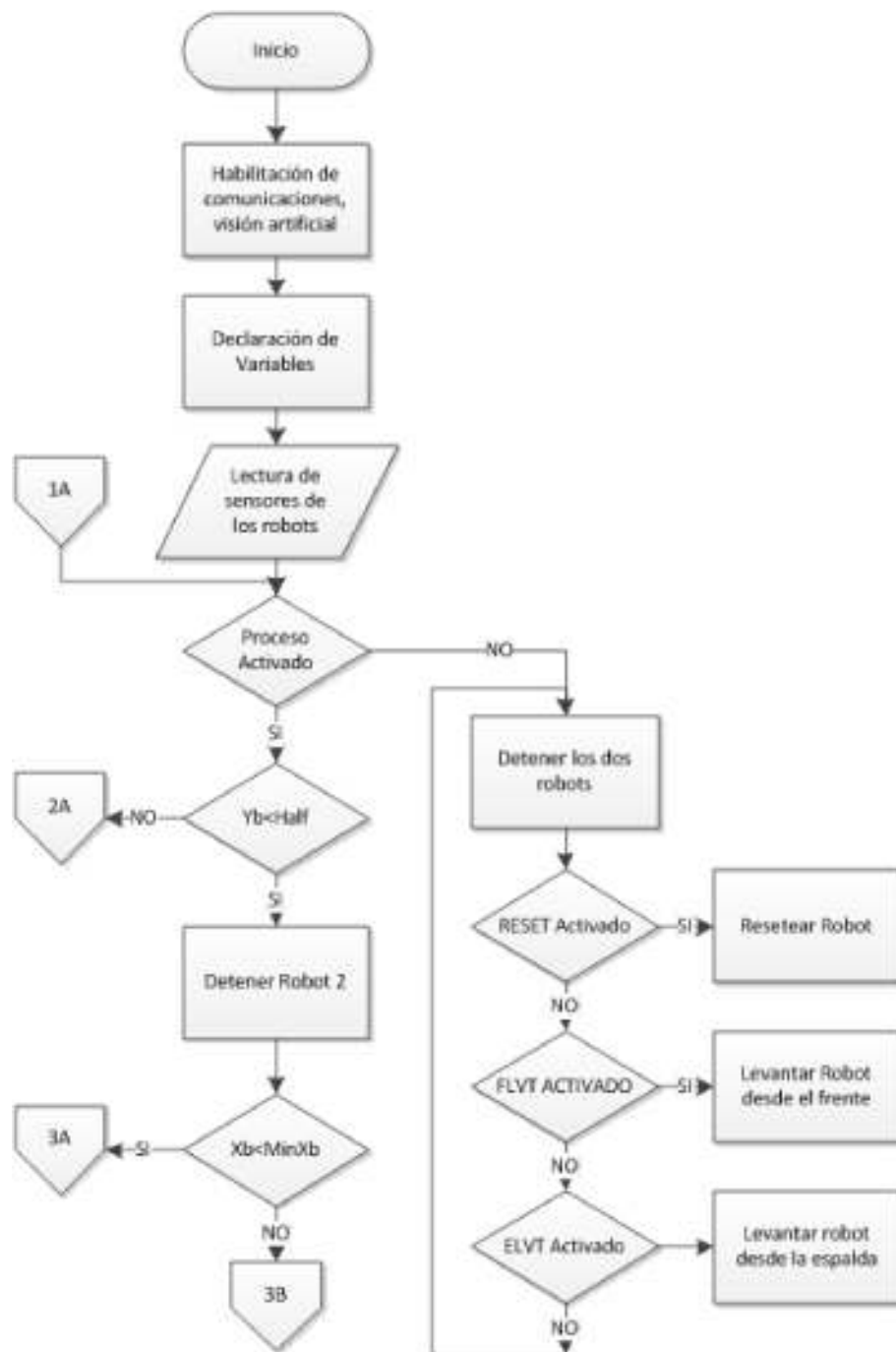


Figura 4.21: Diagrama de Flujo del Programa (1A)



Figura 4.22: Diagrama de Flujo del Programa (2A)



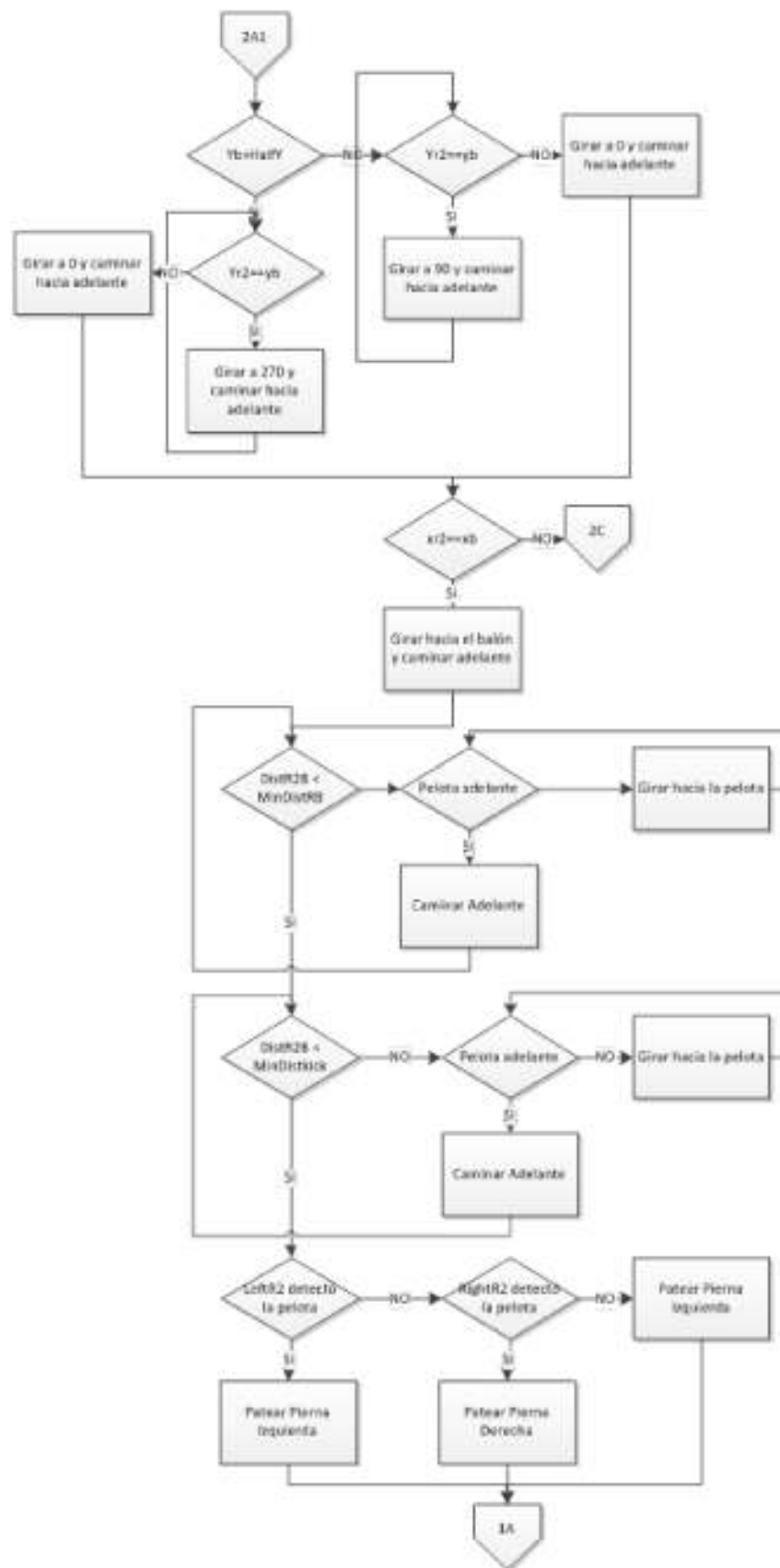


Figura 4.23: Diagrama de Flujo del Programa (2A1)



Figura 4.24: Diagrama de Flujo del Programa (2B)



Figura 4.25: Diagrama de Flujo del Programa (2C)



Figura 4.26: Diagrama de Flujo del Programa (3A)

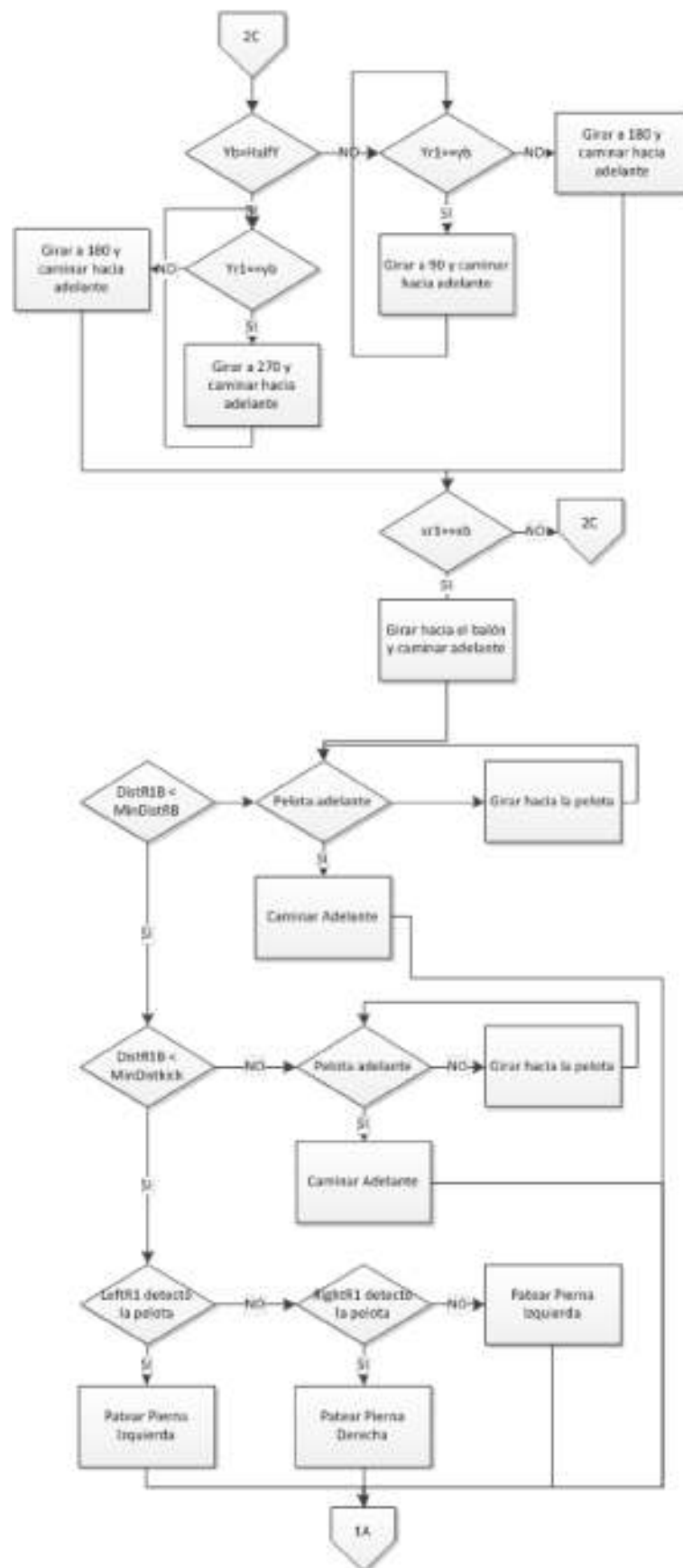


Figura 4.27: Diagrama de Flujo del Programa (3A1)

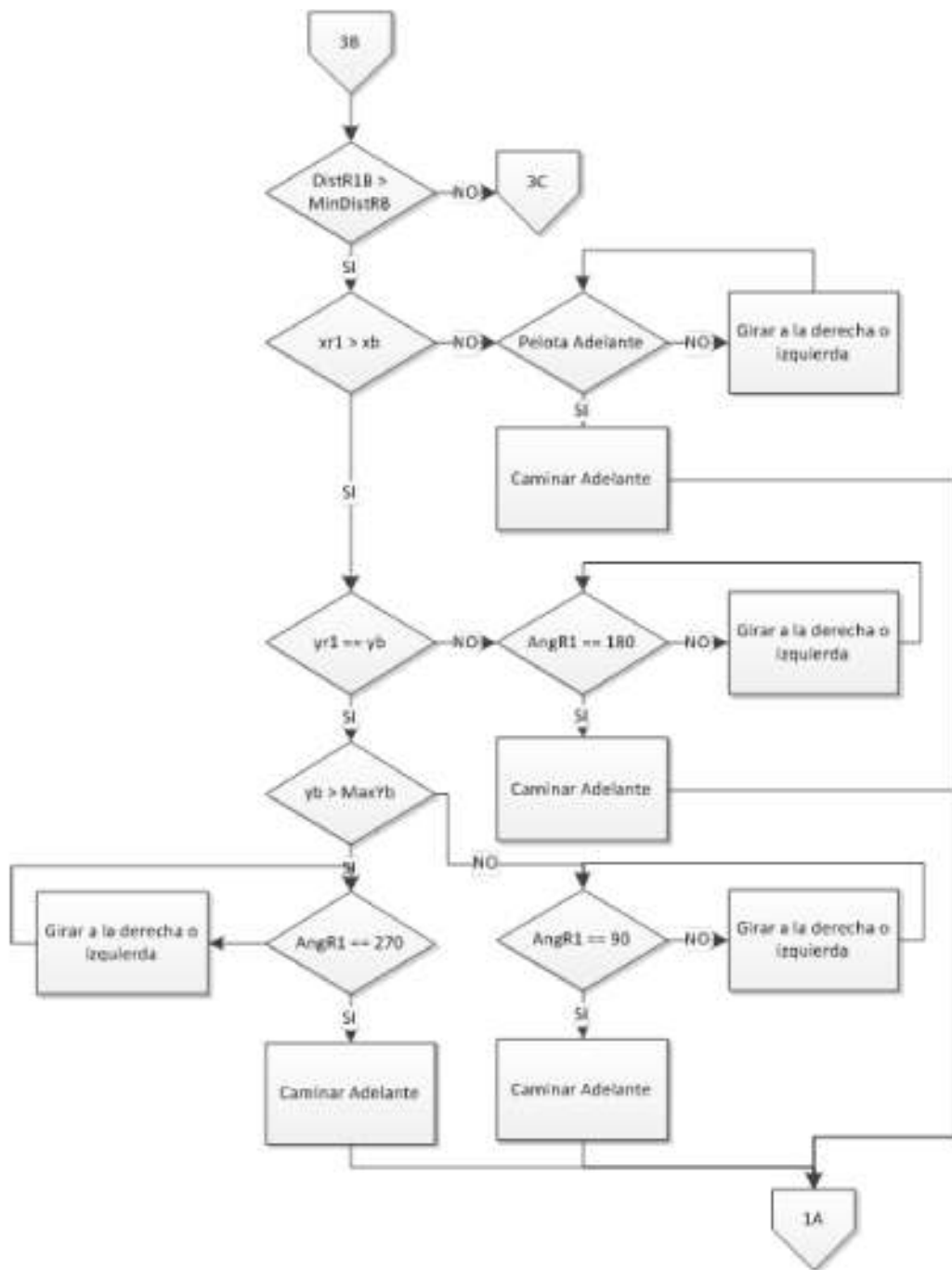


Figura 4.28: Diagrama de Flujo del Programa (3B)

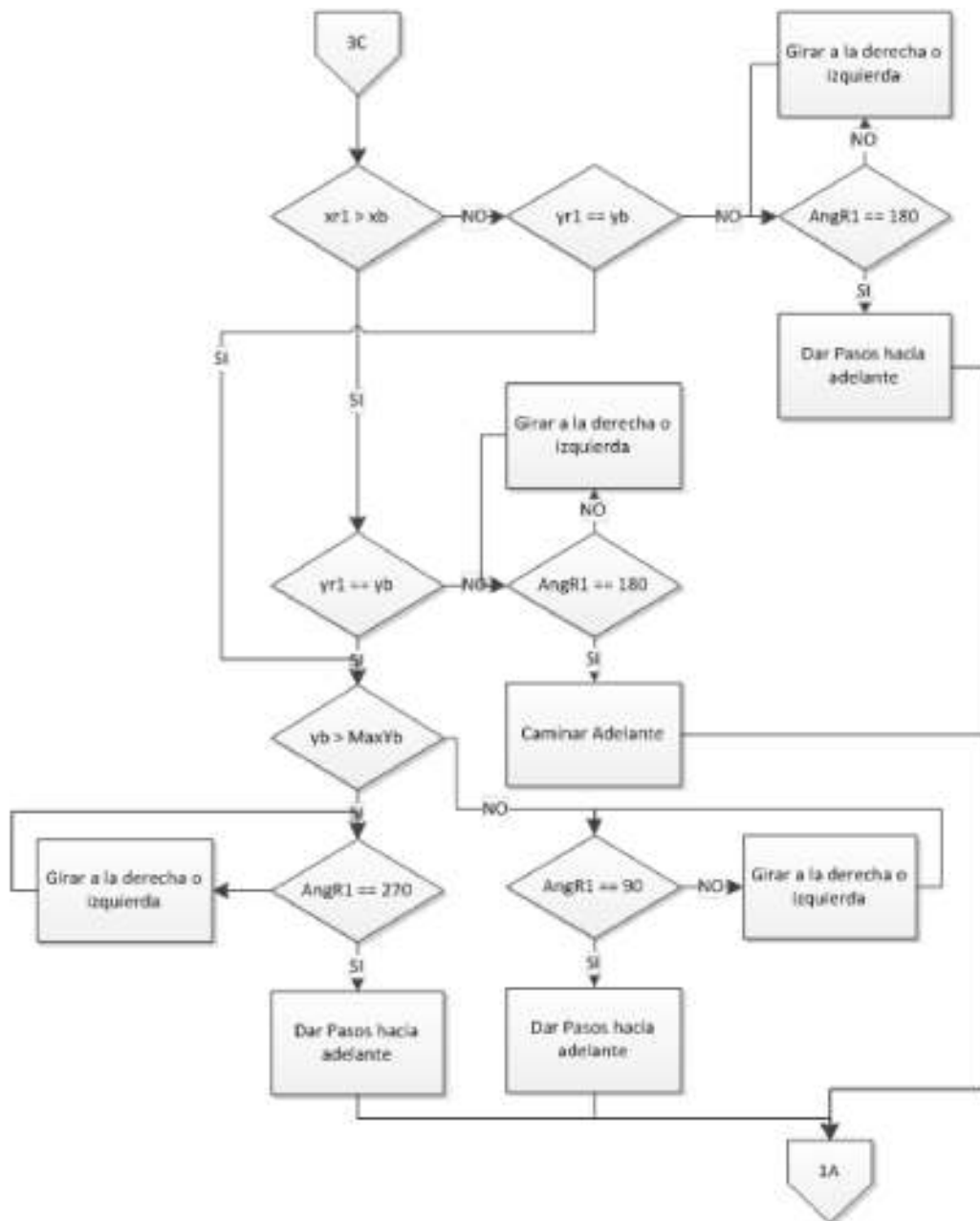


Figura 4.29: Diagrama de Flujo del Programa (3C)

El programa principal se lo puede ver en el anexo B.

## CAPÍTULO 5

### RESULTADOS

En el presente capítulo se expondrá diferentes pruebas realizadas en partes del proyecto y sus respectivos resultados. Además de los resultados en cuanto a las posibilidades de expansión a más robots y cambios en la división del campo de trabajo. La organización de este capítulo se lo puede observar en la Figura 5.1.

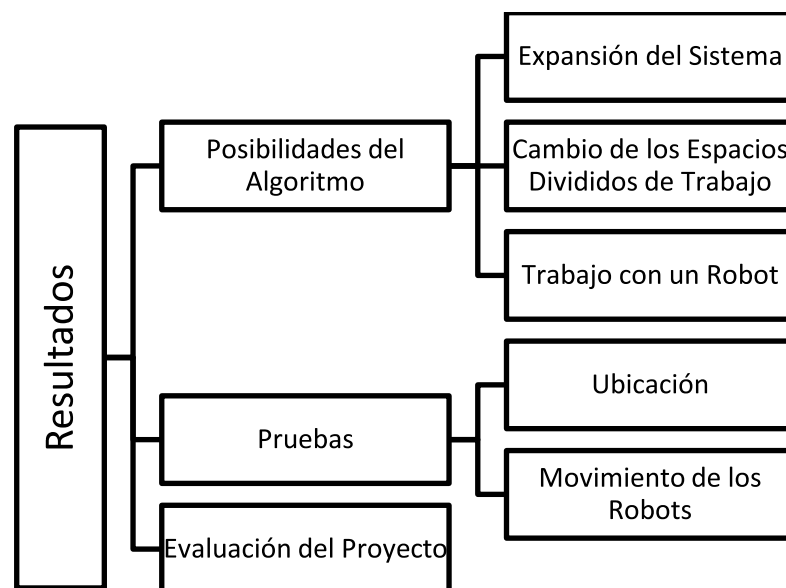


Figura 5.1: Organización del capítulo 5



## 5.1 POSIBILIDADES DEL ALGORITMO

### 5.1.1 EXPANSIÓN DEL SISTEMA

El software de control del proyecto está programado para ser usado con dos robots humanoides Bioloid Premium cuya zona de influencia es cada Espacio Dividido de Trabajo mencionado en el anterior capítulo.

Cada Espacio de Trabajo Dividido corresponde a la mitad del campo total de trabajo y con estos espacios se comprueba que los robots ejerzan su trabajo cuando la pelota es pasada de un espacio dividido hacia el otro. Ya que existen únicamente dos robots, la zona de influencia de cada robot corresponde a un espacio de trabajo dividido. Pero en este algoritmo existe la posibilidad de aumentar el número de robots al dividir cada Espacio de trabajo Dividido en diferentes zonas de influencia dependiendo del número de robots que haya. Un ejemplo se puede ver en la Figura 5.2 con 3 y 5 robots en total.

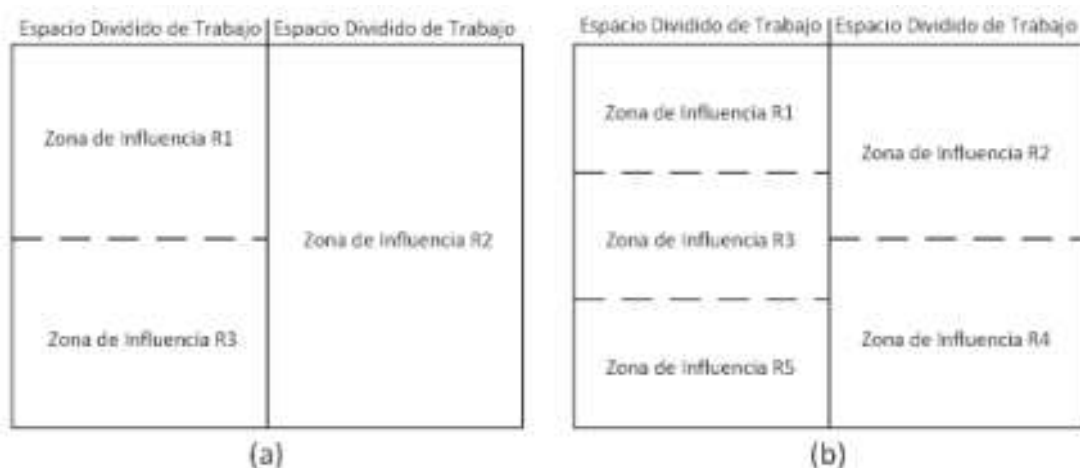


Figura 5.2: Zonas de Influencia con (a) 3 robots y (b) 5 robots

Si se tuviera más de un robot en cada Espacio Dividido de Trabajo cada uno sólo actuaría en su zona de influencia y su objetivo fuera el mismo que cuando se tiene un robot, es decir, pasar la pelota hacia el otro espacio dividido de trabajo. El cambio necesario en el algoritmo consistiría en delimitar la Zona de Influencia de cada robot y enviar las órdenes pertenecientes al robot en cada Espacio Dividido de Trabajo cuando la pelota esté en la zona de influencia de un robot.

Para hacer esta extensión se necesitaría además de otros robots, más módulos de comunicación y una computadora con más puertos seriales ya que cada robot usa un canal de comunicación exclusivo para transmitir y recibir datos.

### **5.1.2 CAMBIO DE LOS ESPACIOS DIVIDIDOS DE TRABAJO**

En la Figura 4.2 del capítulo 4 se puede ver las formas de división que se pueden hacer en el campo de trabajo de las cuales se decidió trabajar con la forma (b). Ya que esta división fue en la etapa inicial del proceso, no es posible usar el algoritmo de la MPU de este proyecto para trabajar con una forma diferente de división del campo de trabajo.

### **5.1.3 TRABAJO CON UN ROBOT**

De la misma manera que hay la posibilidad de trabajar con más de dos robots, también existe la opción de trabajar con un robot haciendo unos

cambios en el programa que consisten en enviar las órdenes para los robots en cada Espacio de Trabajo Dividido a un solo robot.

Se hizo una comparación del trabajo realizado cuando sólo un robot realiza la tarea y cuando los dos robots la hacen conjuntamente. Por lo que se hará comparaciones teniendo los dos robots y solo un robot como se muestra en la Figura 5.3.



Figura 5.3: Pruebas con 1 robot (izq) y dos robots (der)

Esta prueba consistió en una serie de ejecuciones del programa durante 15 minutos cada ejecución. En cada ejecución se contará el número de veces efectiva que la pelota pasa uno a otro Espacio Dividido de Trabajo los cuales están divididos por una línea negra. No se hará el conteo si es que la pelota regresa a la misma área desde la que fue lanzada. Los resultados se pueden ver en la Tabla 5.1.

<b>Prueba</b>	<b>1 robot (pasadas)</b>	<b>2 robots (pasadas)</b>
<b>1</b>	8	15
<b>2</b>	9	17
<b>3</b>	10	19
<b>4</b>	7	20
<b>5</b>	8	14

Tabla 5.1: Comparación del Trabajo

Como se esperaba existe un aumento de pases cuando hay más robots, y se podría esperar lo mismo si se aumentan los robots hasta un número en el que no se interrumpan entre ellos.

## 5.2 PRUEBAS

A los resultados de cada una de las pruebas realizadas se les asignará un puntaje que va desde 1 punto hasta los 5 puntos. Estos puntajes servirán como punto de referencia para dar una calificación al sistema proyecto presentado en este documento.

### 5.2.1 UBICACIÓN

En esta prueba se dará la calificación del sistema de ubicación de los robots y la pelota en el espacio designado. Esta calificación será asignada con 1 punto cuando la ubicación de los objetos es nula y con 5 puntos cuando la ubicación es totalmente confiable.

Como se detalló en el anterior capítulo, se hizo una corrección de las coordenadas proporcionadas por el algoritmo de visión artificial en el reconocimiento de los robots debido a que estos tienen una diferencia de altura con respecto al campo de trabajo. Para comprobar que esta corrección este dando datos precisos se hizo una prueba y se tomó diferentes medidas.

La prueba que se realizó consiste en usar marcadores idénticos a los que llevan los robots sobre sus cabezas para tomar la posición de estos sin corrección cuando están sobre el campo de trabajo. Después se tomó los robots, se los ubicó sobre los marcadores en el campo de trabajo y se tomó las coordenadas con corrección para hacer una comparación con las medidas sin corrección. Los resultados de estas medidas se los puede ver en la Tabla 5.2.

<b>Medida</b>	<b>Campo</b>		<b>Robots</b>	
	x	y	x	y
<b>1</b>	209	377	214	376
<b>2</b>	240	371	248	665
<b>3</b>	364	412	360	410
<b>4</b>	490	317	490	308
<b>5</b>	533	649	533	645
<b>6</b>	856	492	859	495
<b>7</b>	893	725	892	722
<b>8</b>	957	235	968	224
<b>9</b>	1034	344	1048	346
<b>10</b>	1039	642	1046	638

Tabla 5.2: Mediciones corrección de coordenadas

Con estas mediciones se procede al cálculo de los errores absoluto y relativo para poder determinar si la corrección de las coordenadas es confiable. Las fórmulas para el cálculo de los errores son las siguientes:

$$\text{Error absoluto} \quad E_a = x_i - x$$

$$\text{Error relativo porcentual} \quad E_r = \frac{E_a}{x} \times 100\%$$

En la Tabla 5.3 están los valores absolutos y relativos de las mediciones realizadas junto al error relativo promedio.

	<b>x</b>			<b>y</b>		
	Abs.	Rel.	Rel. %	Abs.	Rel.	Rel. %
<b>1</b>	5,00	0,02	2,39	1,00	0,00	0,48
<b>2</b>	8,00	0,03	3,33	6,00	0,03	2,50
<b>3</b>	4,00	0,01	1,10	2,00	0,01	0,55
<b>4</b>	0,00	0,00	0,00	9,00	0,02	1,84
<b>5</b>	0,00	0,00	0,00	4,00	0,01	0,75
<b>6</b>	3,00	0,00	0,35	3,00	0,00	0,35
<b>7</b>	1,00	0,00	0,11	3,00	0,00	0,34
<b>8</b>	11,00	0,01	1,15	11,00	0,01	1,15
<b>9</b>	14,00	0,01	1,35	2,00	0,00	0,19
<b>10</b>	7,00	0,01	0,67	4,00	0,00	0,38
<b>Prom.</b>			1,05			0,85

Tabla 5.3: Errores de las mediciones

De esta manera se puede ver que el error relativo promedio de la corrección de coordenadas en el eje x es del 1.05 % y en el eje y es del 0.85 %.

Además de los datos de corrección de coordenadas se debe tomar en cuenta que hay una cámara sobre los robots y el campo sobre el que estos trabajan está limitado se puede monitorear de una buena forma las posiciones de los objetos que intervienen en el proyecto. Pero en ciertas ocasiones, debido al ángulo de visión de la cámara, los robots obstaculizan la detección de la pelota, aunque el último valor en el que estaba la pelota es mostrado, puede que este valor no refleje cambios que se pueden dar en el lapso de interrupción de la visión. Por esta razón el puntaje es de 4.

### **5.2.2 MOVIMIENTO DE LOS ROBOTS**

Con esta prueba se evaluará que tan rápido logra cada robot pasar la pelota hacia el espacio del otro robot. Se asignará 1 punto si los robots no logran pasar la pelota y 5 puntos si el tiempo promedio es similar a un tiempo de ejecución de pateado de pelota por parte de un robot en el equipo ganador de la *Robocup 2013* categoría robots humanoides tamaño "kids".

Como se detalló en el capítulo anterior los robots poseen un factor configurable de rigidez en sus actuadores. Dependiendo de este factor los movimientos son rígidos o fluidos. Debido a este factor en las pruebas realizadas se da un movimiento de oscilación del robot cuando está en movimiento el cual es más pronunciado a en la parte superior del robot. Debido a que el marcador del robot se encuentra sobre la cabeza este movimiento de oscilación hace que la ubicación del robot cambie con este movimiento. Se hizo diferentes ajustes al factor de oscilación de los 18

actuadores y esta oscilación bajó aunque no fue eliminada debido a las limitaciones físicas del robot.

Para la prueba se toma promedio de tiempo que tomaron los robots en pasar la pelota en las pruebas de Robótica Colaborativa con dos robots detalladas anteriormente. Estos valores se pueden ver en la Tabla 5.4.

<b>Prueba</b>	<b>Pasadas</b>	<b>Tiempo</b>	<b>Tiempo por pasada</b>
<b>1</b>	15	0:15:00	0:01:00
<b>2</b>	17	0:15:00	0:00:53
<b>3</b>	19	0:15:00	0:00:47
<b>4</b>	20	0:15:00	0:00:45
<b>5</b>	14	0:15:00	0:01:04
<b>Promedio</b>			0:00:54

Tabla 5.4: Tiempo por pasada

En la rapidez se tiene que el tiempo promedio de ejecución de los movimientos para pasar la pelota hacia el espacio de trabajo del otro robot es de 54 segundos. Esto es debido a que los robots no usan un movimiento continuo para movilizarse ya que se pretende la mayor estabilidad y control posible en los movimientos. El tiempo normal que usan los robots “Darwin-OP”, del equipo “DARwin” que ganó la *Robocup 2013* en la categoría de tamaño “kid”, es menor a 30 segundos ya que sus movimientos son fluidos. Se asignó la calificación de 3 ya que los robots logran pasar la pelota pero el tiempo en el que lo hacen no es muy rápido.



### 5.3 EVALUACIÓN DEL PROYECTO

Además de los puntajes obtenidos en las distintas pruebas se asignará un puntaje de 3 a las Posibilidades del Algoritmo mencionadas al principio de este capítulo. Esta calificación se debe a que no es posible cambiar el modo de división del campo de trabajo.

En la Tabla 5.5 se puede ver los valores obtenidos por en el presente proyecto.

<b>Indicador de Evaluación</b>	<b>Calificación</b>
<b>Ubicación Confiable</b>	4
<b>Rapidez</b>	3
<b>Posibilidades del Algoritmo</b>	3
<b>Promedio</b>	3,33

Tabla 5.5: Evaluación del proyecto

La calificación final es de 3.3 sobre 5 lo que nos da una pauta para hacer mejoras las cuales estarán listadas en el capítulo 6 de Conclusiones y Recomendaciones.

## CAPÍTULO 6

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 CONCLUSIONES

- La Robótica Colectiva con robots humanoides es un campo multidisciplinario y reciente.
- En este proyecto se usan dos robots humanoides de características similares para que trabajen colectivamente por medio de una unidad de procesamiento central que se basa en la visión artificial y datos de sensores de los robots. Con estos datos la unidad de procesamiento central puede dar órdenes a los robots.
- Cada robot tiene un programa embebido en su controlador para poder interpretar y ejecutar las órdenes que reciben de la unidad central de procesamiento central.
- A través de la visión artificial se logró determinar un sistema de distancias y ubicación que satisfizo las necesidades del proyecto. Este sistema determinó la posición de los robots y la pelota además de la distancia entre estos.

- El elemento más simple del proyecto es la pelota que no cuenta con ningún sistema sobre el cual se pueda hacer control y es parte del medio en el que actúan los robots. Con este elemento se realiza el trabajo de los robots cuando cambian una de sus características, particularmente su posición.
- El algoritmo de Robótica Colectiva del proyecto permite la incorporación de más robots para que realicen el mismo trabajo, pero existen limitaciones físicas que deben ser solventadas primero para poder ejecutar esta implementación.
- Se determinó una calificación de 3.3 sobre 5 en base a tres factores. Con estos factores se pueden hacer mejoras a las continuaciones de este proyecto. Las mejoras deben ser con respecto a la rapidez de desplazamiento de los robots, la flexibilidad del algoritmo y mejoras con respecto al sistema de ubicación.
- Uno de los principales objetivos que se deben trazar en un sistema de robótica colectiva es la disminución del tiempo en la ejecución de un trabajo.
- Al trabajar con robots humanoides se puede ver claramente lo avanzada que es la fisionomía humana ya que diferentes tipos de acciones que realiza fácilmente un humano, son difíciles para un robot.

## 6.2 RECOMENDACIONES

- Se recomienda usar el sistema de visión artificial sobre un conjunto de robots cuando estos no tengan mucha altura con respecto al nivel de la superficie sobre la que se encuentran.
- Se recomienda el uso de un sistema de comunicación que tenga mejores características que el sistema Zigbee.
- Para un mejor control es recomendable usar robots humanoides con cámaras montadas en cada robot, de esta manera se puede analizar mejor el entorno y el objetivo.
- Se recomienda la adquisición de unidades robóticas humanoides más avanzadas para continuar la investigación en el campo de Robótica Humanoide.
- Se recomienda la adquisición de unidades robóticas avanzadas de otros tipos, como por ejemplo drones voladores, para la continuación en la investigación de Robótica Colaborativa.
- Con unidades de robots humanoides más avanzados que los de este proyecto, es recomendable usar una arquitectura de control distribuida que base su toma de decisiones en los datos de todo el grupo. Para esto se puede seguir el lineamiento propuesto en la categoría Humanoide Standard de la competencia Robocup.
- Es necesario que la Universidad de las Fuerzas Armadas apoye más a las unidades y grupos de investigación, como el grupo

“Rovitel”, para de esta manera ser capaces de participar en competiciones internacionales como la Robocup que es celebrada cada año.

## REFERENCIAS BIBLIOGRÁFICAS

- Distributed Robotics Laboratory. (2009). *Programmable Matter*. Recuperado el 13 de Agosto de 2013
- Hackel, M. (2007). *Humanoid Robots: Human Like Machines*. Vienna: I-Tech Educational and Publishing.
- Hoffmanand, G., & Breazeal, C. (2004). Collaboration in Human-Robot Teams.
- Kopacek, P. (2008). ROBOTSOC CER. *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, (págs. 9-11). Seoul.
- Lallee, S., Lemaignan, S., Lenz, A., Melhuish, C., Natale, L., Skachek, S., y otros. (2010). Towards a platform-independent cooperative human-robot interaction system: I. Perception. *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (págs. 4444 - 4451). Taipei.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge: Cambridge University Press.
- León Fernández de Lara, Y., & Muñoz Meléndez, A. (2007). Investigación sobre los Requisitos del Movimiento. *Revista Iberoamericana de Inteligencia Artificial*, 83-103.

- Martinoli, A. (1999). *Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies*. Lausanne: EPFL.
- Montenegro Campos, M. (2012). Cooperative and Collaborative Robotics. *Latin American Robotics Symposium / Brazilian Robotics Symposium*. Fortaleza.
- Moran, M. (2006). The da Vinci robot. *Journal of Endourology*, 986-90.
- Rocher, S., & Duhaut, D. (1998). MARCH: A Flexible Multi-Agent Architecture, Applied to Autonomous Robots Playing Football. En A. Drogoul, M. Tambe, & T. Fukuda, *Collective Robotics: First International Workshop* (págs. 98-109). París: Springer.
- Vukobratović, M. (2004). Zero-moment point—Thirty five years of its life. *International Journal of Humanoid Robotics*, 157–173.
- Wadsworth, D., & Few, D. (27 de Diciembre de 2010). *Humanoid Robotics*. Recuperado el 13 de Agosto de 2013, de <https://inlportal.inl.gov/portal/server.pt/community/introduction/536>
- Weisstein, E. W. (2010). *Euler Angles*. Recuperado el 15 de Agosto de 2013, de MathWorld--A Wolfram Web Resource: <http://mathworld.wolfram.com/EulerAngles.html>
- Wolf, J. C., Hall, P., Robinson, P., & Culverhouse, P. (2007). Bioid based Humanoid Soccer Robot Design.

Yavuz, S. Ç. (2009). *Kinematic Analysis for Robot Arm*. Estambul: Yildiz Technical University.