REPÚBLICA DEL ECUADOR
Escuela Politécnica Nacional
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.

- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.

- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

Aníbal Roberto Pérez Checa

Junio 2016

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

APLICACION DE ALGORITMOS DE VISION ARTIFICIAL PARA EL CONTROL DE ROBOTS COOPERATIVOS MOVILES DENTRO DE ESPACIOS SEMI ESTRUCTURADOS

TESIS PREVIA A LA OBTENCIÓN DEL GRADO DE MAGÍSTER EN AUTOMATIZACION Y CONTROL ELECTRÓNICO INDUSTRIAL

ANÍBAL ROBERTO PÉREZ CHECA
robertoperez32@hotmail.com

Director: Dr. Jorge Andrés Rosales Acosta.
andres.rosales@epn.edu.ec

Quito, junio 2016

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

APLICACION DE ALGORITMOS DE VISION ARTIFICIAL PARA EL CONTROL DE ROBOTS COOPERATIVOS MOVILES DENTRO DE ESPACIOS SEMI ESTRUCTURADOS

TESIS PREVIA A LA OBTENCIÓN DEL GRADO DE MAGÍSTER EN AUTOMATIZACION Y CONTROL ELECTRÓNICO INDUSTRIAL

ANÍBAL ROBERTO PÉREZ CHECA
robertoperez32@hotmail.com

Director: Dr. Jorge Andrés Rosales Acosta.
andres.rosales@epn.edu.ec

Quito, junio 2016

DECLARACIÓN

Yo, Aníbal Roberto Pérez Checa, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Aníbal Roberto Pérez Checa

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Aníbal Roberto Pérez Checa, bajo mi supervisión.

_____

Dr. Jorge Andrés Rosales Acosta.

DIRECTOR

AGRADECIMIENTO

Primero agradezco a **Dios**, que con su eterna gracia me ha dado la fortaleza suficiente para concluir con este trabajo de grado que ha demandado un gran esfuerzo y perseverancia.

A mi familia, a mi esposa Glenda y a mis hijos por haber tenido la paciencia y comprensión ante mi abandono durante muchas horas en búsqueda de la conclusión de este trabajo.

A Andrés, mi director de tesis, por la dedicación prestada a este trabajo a pesar del corto tiempo y la premura del proceso, debido a los cambios en las leyes nacionales.

A la Escuela Politécnica Nacional y todo su cuerpo docente, por la gran cantidad de conocimientos brindados a lo largo de todo mi tiempo de estudio.

Roberto.

DEDICATORIA

Dedico este trabajo
a mis hijos y sobrinas,
para que entiendan que los
sueños se cumplen con
esfuerzo y dedicación.

Sobre todo, para que se den cuenta
que los sueños tienen
como límite el que cada
quien le ponga.

Roberto…

CONTENIDO

DECLARACIÓN i

CERTIFICACIÓN ii

AGRADECIMIENTO iii

DEDICATORIA iv

CONTENIDO v

ÍNDICE DE FIGURAS x

ÍNDICE DE TABLAS xvi

RESUMEN xvii

PRESENTACIÓN xviii

1. CAPITULO 1 1

1.1. Estado del arte 1

1.2. Robótica cooperativa 2

1.2.1. Control de formaciones. 3

1.2.2. Transporte de objetos. 3

1.3. Planificación de trayectorias 8

2. CAPÍTULO 2 10

2.1. Marco teórico 10

2.2. Robótica 10

2.2.1. Clasificación generacional de la robótica 10

2.2.1.1. Primera generación. 10

2.2.1.2. Segunda generación. 10

2.2.1.3. Tercera generación. 11

2.2.1.4. Cuarta generación. 11

2.2.1.5. Quinta generación. 11

2.2.2. Clasificación en base de la arquitectura de la robótica 11

2.2.2.1. Poli articulados. 11

2.2.2.2. Móviles. 12
2.2.2.3. Androides. 12
2.2.2.4. Zoomórficos. 12
2.2.2.5. Híbridos. 13
2.2.3. Clasificación en base de la ubicación de la inteligencia 13
2.3. Estructuras robóticas móviles con ruedas 13
2.3.1. Componentes 13
2.3.2. Estructura. 14
2.3.3. Actuadores. 14
2.3.4. Sensores. 14
2.3.5. Inteligencia. 15
2.4. Robótica Cooperativa. 15
2.4.1. Control Centralizado 15
2.4.2. Control descentralizado. 16
2.4.3. Muy Acoplados o colectivos. 17
2.4.4. Poco acopladas o manadas 17
2.5. Inteligenica artificial 18
2.5.1. Visión artificial 18
2.5.2. Cámara 19
2.5.3. Tipos de imágenes. 19
2.5.3.1. Imagen RGB. 19
2.5.3.2. Imagen CMYK. 20
2.5.3.3. Imagen Gris. 20
2.5.3.4. Imagen Binaria. 20
2.5.4. Iluminación 21
2.5.4.1. Iluminación direccional. 21
2.5.4.2. Iluminación difusa. 22

2.6. Operaciones con imágenes 23
2.6.1. Conversión RGB a Gris 23
2.6.2. Conversión RGB a binaria. 23
2.6.3. Etiquetado de imágenes 23
2.6.4. Filtrado de imágenes 24
2.6.5. Filtro mediana 26
2.6.6. Segmentación de imágenes por color. 26
2.6.7. Búsqueda de imágenes o patrones 28
2.6.8. Reducción de imagen 29
2.6.9. Dilatación 29
2.7. Algoritmo A* 30
2.8. Reconstrucción de ruta 33
2.9. Lógica Difusa 34
2.9.1. Conceptos básicos 35
2.9.1.1. Universo discurso. 35
2.9.1.2. Función de membresía. 35
2.9.2. Razonamiento Difuso 36
2.9.3. Reglas Si-Entonces 38
2.10. Control difuso 38
3. CAPÍTULO 3 45
3.1. Visión general del sistema 45
3.2. Diseño del hardware 45
3.2.1. Tarjeta de Desarrollo 46
3.2.2. Motores de tracción 47
3.2.3. Rueda loca 48
3.2.4. Motor de horquilla 49
3.2.5. Comunicación Infrarroja 51
3.2.6. Fuente de alimentación 51

3.3. Diseño del software 57
3.4. Diseño de software para el maestro 57
3.4.1. Bloque SETEO 58
3.4.2. Bloque PATRONES 58
3.4.3. Bloque ENTORNO PUNTO FINAL 60
3.4.4. Bloque OBTENER PUNTOS 61
3.4.5. Bloque REDUCCIÓN 62
3.4.6. Bloque RUTA ROJO 64
3.4.7. Bloque MOVIMIENTO ROJO 65
3.4.8. SISTEMA DIFUSO 67
3.4.9. Bloque ELEVACIÓN DE OBJETO 72
3.4.10. Bloque RUTA COOPERATIVA 72
3.4.11. Bloque MOVIMIENTO COOPERATIVO 73
3.4.12. Bloque FINALIZACIÓN 76
3.5. Diseño de software de los esclavos 76
4. CAPÍTULO 4 79
4.1. Pruebas y resultados 79
4.2. Prueba 01: MOVIMIENTO LIBRE. 79
4.2.1. Ruta horizontal Robot_A. 80
4.2.2. Ruta vertical Robot_A. 89
4.2.3. Rutas a múltiplos de 45° Robot_A 90
4.2.4. Rutas curvas Robot_A. 95
4.2.5. Ruta horizontal Robot_B. 97
4.2.6. Ruta vertical Robot_B. 97
4.2.7. Rutas a múltiplos de 45° del Robot_B. 98
4.2.8. Rutas curvas Robot_B. 100
4.3. Prueba 02: fuente. 102
4.4. Prueba 03: Movimiento cooperativo 1. 106

4.4.1. Fases comunes. 106
4.4.2. Obtención de puntos. 107
4.4.3. Desplazamiento de los robots individuales. 110
4.4.4. Desplazamiento cooperativo. 113
4.5. Prueba 04: movimiento cooprativo 2 118
4.5.1. Obtención de puntos. 118
4.5.2. Desplazamiento de los robots individuales 119
4.5.3. Desplazamiento cooperativo. 123
4.6. Prueba 05: movimiento cooprativo 3 127
4.6.1. Desplazamiento de los robots individuales. 128
4.6.2. Desplazamiento cooperativo. 131
4.7. Prueba 06: movimiento cooprativo 4 134
4.7.1. Desplazamiento de los robots individuales. 134
4.7.2. Desplazamiento de los robots individuales. 135
4.7.3. Desplazamiento cooperativo. 137
5. CAPÍTULO 5 140
5.1. Conclusiones 140
5.2. Recomendaciones 144
6. REFERENCIAS BIBLIOGRÁFICAS. 145
6.1. Bibliografía 145
7. ANEXOS 150
7.1. Anexo 1: Manual de Usuario. 150
7.2. Anexo 2: Estudio Paralaje. 158
7.3. Anexo 3: Datos Técnicos Placa Arduino Mega. 165
7.4. Anexo 4: Datos Técnicos Driver de Horquilla. 173
7.5. Anexo 5: Datos Técnicos Driver de Motores. 179
7.6. Anexo 6: Datos Técnicos Módulo Bluetooth. 182

ÍNDICE DE FIGURAS

Figura 2-1 Robots poli articulados 11
Figura 2-3 Robot móvil 12
Figura 2-4 Robot humanoide Nao 12
Figura 2-5 Big Dog. robot zoomórfico. 12
Figura 2-6 iRobot 510 PackBot, robot híbrido 13
Figura 2-7 Chasis comercial de robot móvil. 14
Figura 2-8 Motores para robots móviles. 14
Figura 2-9 Algunos sensores utilizados en robots móviles. 14
Figura 2-10 Algunos controladores para robots 15
Figura 2-11 Esquema control centralizado. 16
Figura 2-12 Esquema de control descentralizado. 16
Figura 2-13 Sistema muy acoplado o colectivo. 17
Figura 2-14 Sistema poco acoplado o manada. 17
Figura 2-15 Esquema de captura de una cámara 19
Figura 2-16 Componentes RGB y CMYK 20
Figura 2-17 Imagen escala de grises 20
Figura 2-18 Imagen Binaria 20
Figura 2-19 Descripción de las coordenadas de las imágenes 21
Figura 2-20 Sistema de iluminación directa 22
Figura 2-21 Sistema de Iluminación difusa. 22
Figura 2-22 Imagen binarizada e imagen segmentada por área 24
Figura 2-23 Forma de ejecución de la convolución. 25
Figura 2-24 Componentes de color de la imagen RGB 26
Figura 2-25 Segmentado color rojo 27
Figura 2-26 Desplazamiento de la plantilla 28
Figura 2-27 Compresión por áreas 29
Figura 2-28 Imagen dilatada 30
Figura 2-29 Entorno de trabajo 30
Figura 2-30 Imagen binarizada del entorno 31
Figura 2-31 Matriz de entorno indexado 31
Figura 2-32 Distancias entre vecinos 32

Figura 2-33 Camino encontrado 32
Figura 2-34 Trazado de curvas de Bézier 34
Figura 2-35 Partes de una función de pertenencia 36
Figura 2-36 Ejemplo variable lingüística 37
Figura 2-37 Control realimentado 38
Figura 2-38 Controlador difuso 39
Figura 2-39 Funciones que intervienen en la fusificación. 40
Figura 2-40 Resultado de la fusificación. 40
Figura 2-41 Primera parte de inferencia 42
Figura 2-42 Segunda parte de inferencia 42
Figura 2-43 Métodos de defusificación. 43
Figura 3-1 Estructura general del sistema 45
Figura 3-2 Estado inicial de los agentes como juguetes 46
Figura 3-3 Placa Arduino Mega 2560 47
Figura 3-4 Driver dual de motores 47
Figura 3-5 Rueda loca de los agentes. 48
Figura 3-6 Driver TA7257P. 49
Figura 3-7 Sistema regla sensor. 50
Figura 3-8 Acondicionador de sensor. 50
Figura 3-9 Sistema de comunicación infrarroja. 51
Figura 3-10 Circuitos de alimentación de fuente. 52
Figura 3-11 Prestaciones adicionales de la placa. 53
Figura 3-12 Forma para el circuito impreso. 53
Figura 3-13 Distribución elementos en placa de robots. 54
Figura 3-14 Placas con el ruteo de los circuitos. 54
Figura 3-15 Circuitos en tres dimensiones. 55
Figura 3-16 Marcas y pesos en los agentes. 55
Figura 3-17 Vistas de los agentes automatizados. 56
Figura 3-18 Vistas laterales de los agentes automatizados. 56
Figura 3-19 Objeto y Pallete. 56
Figura 3-20 Esquema general del sistema. 57
Figura 3-21 Bloque Seteo. 58
Figura 3-22 Bloque patrones. 60

Figura 3-23 Bloque entorno punto final. 61
Figura 3-24 Algoritmo obtener puntos del entorno. 62
Figura 3-25 Algoritmo reducción. 63
Figura 3-26 Ruta Robot_A. 64
Figura 3-27 Ruta Robot_B. 65
Figura 3-28 Movimiento del Robot. 66
Figura 3-29 Entorno dividido. 67
Figura 3-30 Estructura de control del sistema. 68
Figura 3-31 Controlador difuso MatLab. 68
Figura 3-32 Variable de entrada error. 69
Figura 3-33 Variable de salida motor derecho. 70
Figura 3-34 Variable salida motor izquierdo. 71
Figura 3-35 Elevación del objeto. 72
Figura 3-36 Rutas cooperativas. 73
Figura 3-37 Imagen robots en punto final. 74
Figura 3-38 Algoritmo de movimiento cooperativo. 75
Figura 3-39 Forma de vector recibido. 77
Figura 3-40 Algoritmo de robots esclavos. 78
Figura 4-1 Toma a diferente posición. 79
Figura 4-2 Imagen fusionada. 80
Figura 4-3 Modificación a la altura de marcas. 80
Figura 4-4 Ruta virtual. 80
Figura 4-5 Ruta inicial del robot rojo. 81
Figura 4-6 Ángulos generados robot rojo. 81
Figura 4-7 Error generado entre los ángulos. 82
Figura 4-8 Ruta corregida del controlador. 83
Figura 4-9 Ángulos de movimiento. 84
Figura 4-10 Error entre ángulos. 84
Figura 4-11 Gráfica relación tiempo – distancia Robot_A. 85
Figura 4-12 Gráfica relación tiempo – distancia Robot_B. 86
Figura 4-13 Trama de comunicación enviada a los esclavos. 87
Figura 4-14 Ruta horizontal Robot_A. 87
Figura 4-15 Error de ángulo. 88

Figura 4-16 Ruta vertical Robot_A. 89
Figura 4-17 Error entre ángulos ruta vertical Robot_A. 89
Figura 4-18 Ruta virtual a 135°. 90
Figura 4-19 Posiciones con ruta a 135°. 90
Figura 4-20 Error de ángulo con ruta a 135°. 91
Figura 4-21 Ruta virtual a 45°. 91
Figura 4-22 Posiciones con ruta a 45°. 92
Figura 4-23 Error de ángulo con ruta a 45°. 92
Figura 4-24 Ruta virtual -45°. 93
Figura 4-25 Posición y error en la ruta -45°. 93
Figura 4-26 Ruta virtual a -135°. 94
Figura 4-27 Posición y error ruta -135°. 94
Figura 4-28 Movimiento primer cuadrante Robot_A. 95
Figura 4-29 Movimiento segundo cuadrante Robot_A. 95
Figura 4-30 Movimiento tercer cuadrante Robot_A. 96
Figura 4-31 Movimiento cuarto cuadrante Robot_A. 96
Figura 4-32 Ruta horizontal Robot_B. 97
Figura 4-33 Ruta vertical del Robot_B. 98
Figura 4-34 Ruta 135° Robot_B. 98
Figura 4-35 Ruta 45° Robot_B. 99
Figura 4-36 Ruta a -45° Robot_B. 99
Figura 4-37 Ruta -135° Robot_B. 100
Figura 4-38 Movimiento primer cuadrante Robot_B. 100
Figura 4-39 Movimiento segundo cuadrante Robot_B. 101
Figura 4-40 Movimiento tercer cuadrante Robot_B. 101
Figura 4-41 Movimiento cuarto cuadrante Robot_B. 102
Figura 4-42 Señal PWM. 102
Figura 4-43 Ruta con diferentes valores de fuente. 104
Figura 4-44 Falso positivo color azul. 104
Figura 4-45 Robot con cubierta. 105
Figura 4-46 Fases Comunes. 106
Figura 4-47 Puntos de Movimiento prueba 03. 107
Figura 4-48 Compresión del Entorno prueba 03. 108

Figura 4-49 Ruta Robot_A prueba 03. 109
Figura 4-50 Ruta Robot_B prueba 03. 109
Figura 4-51 Movimientos independientes de robots prueba 03. 110
Figura 4-52 Distancias generadas por robots individuales prueba 03. 110
Figura 4-53 Errores de distancia de los robots individuales prueba 03. 111
Figura 4-54 Ángulos del Robot_A, prueba 03. 112
Figura 4-55 Ángulos del Robot_B, prueba 03. 112
Figura 4-56 Cambios en el Objeto. 113
Figura 4-57 Rutas cooperativas prueba 03. 114
Figura 4-58 Movimiento Cooperativo prueba 03. 114
Figura 4-59 Distancias Cooperativas de los Robots prueba 03. 115
Figura 4-60 Ángulos de los robots en el sistema cooperativo prueba 03. 116
Figura 4-61 Distancias ejecutadas por objeto prueba 03. 117
Figura 4-62 Puntos obtenidos prueba 04. 118
Figura 4-63 Operaciones de compresión y ubicación de puntos prueba 04. 118
Figura 4-64 Ruta Robot_A prueba 04. 119
Figura 4-65 Ruta Robot_B prueba 04. 119
Figura 4-66 Rutas individuales de los robots prueba 04. 120
Figura 4-67 Gráficas de distancias generadas movimiento individual prueba 04. 120
Figura 4-68 Gráfico de los errores de distancia movimiento solos prueba 04. 121
Figura 4-69 Ángulos del Robot_A prueba 04. 122
Figura 4-70 Ángulos del Robot_B prueba 04. 122
Figura 4-71 Rutas cooperativas prueba 04. 123
Figura 4-72 Rutas cooperativas prueba 04. 123
Figura 4-73 Cambio sistema de iluminación. 124
Figura 4-74 Distancias cooperativas prueba 04. 125
Figura 4-75 Distancia del objeto prueba 04. 126
Figura 4-76 Puntos ubicados prueba 05. 127
Figura 4-77 Entorno tratado con puntos ubicados prueba 05. 127
Figura 4-78 Rutas Robot_A prueba 05. 128
Figura 4-79 Ruta Robot_B prueba 05. 128
Figura 4-80 Rutas robots prueba 05. 129
Figura 4-81 Distancia absoluta robots prueba 05. 129

Figura 4-82 Errores absolutos de robots prueba 05. 130
Figura 4-83 Error entre ángulos de los robots prueba 05. 131
Figura 4-84 Rutas cooperativas prueba 05. 131
Figura 4-85 Movimiento cooperativo prueba 05. 132
Figura 4-86 Distancias absolutas cooperativas prueba 05. 132
Figura 4-87 Distancia Movimiento cooperativo prueba 05. 133
Figura 4-88 Puntos Obtenidos para movimiento prueba 06. 134
Figura 4-89 Puntos en entorno reducido prueba 06. 134
Figura 4-90 Rutas reconstruidas para robots prueba 06. 135
Figura 4-91 Rutas individuales de los robots prueba 06. 135
Figura 4-92 Distancias Individuales prueba 06. 136
Figura 4-93 Error Rutas prueba 06. 136
Figura 4-94 Rutas Cooperativas prueba 06. 137
Figura 4-95 Rutas cooperativas ejecutadas de los robots prueba 06. 138
Figura 4-96 Distancia cooperativa prueba 06. 138
Figura 4-97 Error distancia cooperativa prueba 06. 139

ÍNDICE DE TABLAS

Tabla 2-1 Planteamientos de inferencia..... 41
Tabla 3-1 Características de Arduino Mega 2560 46
Tabla 3-2 Características de driver 298N. 48
Tabla 3-3 características CI TA7257P..... 49
Tabla 3-4 Variable de entrada error..... 69
Tabla 3-5 Variable Salida motor derecho..... 70
Tabla 3-6 Variable salida motor izquierdo. 71
Tabla 3-7 Reglas de Control..... 71
Tabla 3-8 Tabla de comandos existentes..... 77
Tabla 4-1 Variable lingüística del error..... 82
Tabla 4-2 Recolección de datos Robot_A. 85
Tabla 4-3 Recolección de datos Robot_B. 86
Tabla 4-4 Resumen de errores de distancia robots solos prueba 03..... 111
Tabla 4-5 Resumen de Errores de Distancia Cooperativos prueba 03 115
Tabla 4-6 Resumen Movimiento del Objeto prueba 03. 117
Tabla 4-7 Tabla Resumen Movimiento Individual prueba 04..... 121
Tabla 4-8 Resumen Distancia Cooperativa prueba 04. 125
Tabla 4-9 Tabla Resumen Movimiento Individual prueba 05..... 130
Tabla 4-10 Resumen Errores Cooperativos prueba 05 133
Tabla 4-11 Resumen Errores Movimientos Individuales prueba 06. 137

RESUMEN

Esta tesis de maestría tiene como propósito la aplicación de algoritmos de visión artificial para el control de dos agentes cooperativos dentro de un entorno semi estructurado para la movilización de un objeto dentro del mismo.

Se inicia realizando un estudio del estado del arte en el cual se puede apreciar que los sistemas robóticos cooperativos para su funcionamiento se apoyan en sistemas sensoriales que les permiten la interconexión entre los agentes, de este análisis nace el interés de que el sistema funcione únicamente con visión artificial.

Se parte de dos juguetes montacargas a los que se los automatizó para que sean capaces de recibir datos vía bluetooth, con la información de las acciones a ejecutar para seguir una ruta determinada por el sistema de visión artificial, el mismo que está conformado por una cámara fotográfica simple la cual provee la imagen mediante una conexión USB a un computador personal.

Para el desarrollo de los algoritmos de análisis y control se utiliza Matlab como plataforma de programación debido a su versatilidad en el manejo de matrices. El algoritmo computacional está basado en la forma de análisis que hace el cerebro humano para el movimiento de un objeto, para lo cual establecen marcas sobre el objeto, realiza las acciones de movimiento y posteriormente vuelve a ejecutar estos dos pasos hasta llegar al objetivo. Para esto se utilizan algoritmos como: filtrado de colores RGB, segmentación, medición de ángulos, etiquetado, cálculo de distancias, curvas de Bézier, búsqueda de caminos, entre otros.

De los algoritmos de visión artificial se obtiene información para que el controlador difuso, diseñado para el sistema, produzca las acciones de control sobre los motores de los agentes para que estos corrijan su ángulo de orientación o avancen por la ruta. Conforme los agentes avanzan a lo largo de la ruta se registran los puntos de movimiento para su posterior análisis, obteniéndose errores de posición de pocos centímetros y una respuesta del controlador con errores en los ángulos que no superan el ±1°.

El sistema cumple con el objetivo del movimiento del objeto desde un punto A hasta un punto B, con la particularidad que los agentes se ubican frente al objeto y después lo desplazan, un especial punto a destacar es que el sistema utiliza solo visión artificial para determinar el movimiento y no se apoya en realimentación de la velocidad de las llantas como trabajos similares.

PRESENTACIÓN

En este trabajo se presenta un sistema de control centralizado en base de un sistema difuso como algoritmo de control, el mismo que obtiene información de imágenes capturadas mediante una cámara común y determina las acciones de control, las interpreta y las transmite vía bluetooth a dos robots de similares características para que estos muevan un objeto desde un punto A hasta un punto B dentro de un entorno semi estructurado. Este documento se encuentra organizado de la siguiente manera:

Capítulo 1:

En este capítulo se presenta un estudio del estado del arte en lo referente a sistemas cooperativos de robots y sus aplicaciones orientados a solucionar problemas de transporte de elementos mediante el seguimiento de rutas. La robótica cooperativa y sus aplicaciones que las relacionan con la visión artificial.

Capítulo 2:

En este apartado se da un enfoque a los componentes teóricos que forman parte de esta tesis empezando por la robótica, pasando por la visión artificial con cada uno de sus componentes y factores que intervienen en su funcionamiento. También se explican cada uno de los métodos de procesamiento de imágenes utilizados dentro del punto de vista de su sustentación teórica. Se explica la forma de operación del controlador difuso que determina las acciones de control y sus componentes.

Capítulo 3:

En esta parte del trabajo se explica cómo está conformado el sistema en su totalidad, así como también los diseños realizados tanto del hardware como del software. Se explica mediante imágenes descriptivas cada uno de los diseños realizados y mediante diagramas de flujo la lógica que se ha utilizado para cada uno de los algoritmos que se han utilizado para la conformación del sistema inteligente.

Capítulo 4:

Este capítulo está dedicado a realizar las pruebas de operación y recolección de datos para su posterior análisis. En su contenido se inicia con la evaluación del movimiento de los robots de manera independiente y posteriormente se hace un análisis similar, pero con el trabajo cooperativo de los dos robots en base de rutas definidas por el sistema de visión artificial.

Capítulo 5:

Esta sección recoge las conclusiones obtenidas de trabajo realizado, así como también las recomendaciones sugeridas para trabajos futuros que se enfoquen en el mismo tipo de sistemas.

CAPITULO 1

1.1. ESTADO DEL ARTE

El deseo imparable de los seres humanos de construir máquinas que los alivianen en sus actividades diarias han hecho que el mundo de la robótica se encuentre en constate y vertiginosa evolución. Día a día los sistemas robóticos han ido evolucionando desde máquinas simples que cumplían y cumplen una función o tarea específica de forma individual a sistemas que cooperan entre sí y que poseen inteligencia para reducir su esfuerzo y cumplir con un objetivo afín entre los agentes.

Esta característica de trabajo en conjunto es algo que los investigadores han tratado de reproducir mediante algoritmos computacionales generando sistemas de inteligencia distribuida o centralizada, esto basado el comportamiento de los equipos de trabajo, en donde cada miembro del grupo se encarga de una tarea en específico y con la suma en conjunto de todas las acciones se llega al fin u objetivo de la tarea general.

Para que exista la iteración entre los robots y con su medio circundante es necesaria la utilización de sensores que permitan esta operación, es así que la electrónica ha desarrollado múltiples tipos de estos tratando de equiparar a los sentidos del ser humano.

Considerando que uno de los sentidos que recoge más información del medio circundante es la vista, esta funcionalidad ha sido reproducida apareciendo una nueva forma de conocimiento, dentro de los sistemas computacionales, que es la visión artificial, la misma que basada en el tratamiento de las imágenes adquiridas por medio de una cámara, elemento primario, permite la obtención de mucha información para que los sistemas robóticos puedan comprender y analizar su entorno.

1.2. ROBÓTICA COOPERATIVA

El avance de los sistemas multi-robot da pasos gigantes día a día, es así que se han formulado varias líneas de investigación sobre este particular de tal forma que se los podría clasificar de la siguiente manera: Inspiraciones biológicas, comunicación, arquitecturas, localización, mapeo, exploración y movimiento de objetos y robots reconfigurables [1] .

Los sistemas cooperativos de robots, presentan muchas ventajas con respecto a los sistemas con agentes únicos, como por ejemplo [2].

- Existen tareas que son imposibles ser realizadas por un solo agente, por su complejidad o limitación en el espacio, haciendo necesaria la intervención de más de un agente [2].
- La utilización de varios agentes permite el aumento de la eficiencia y la eficacia en la consecución de una tarea, por ejemplo, en exploración de entornos en donde la utilización de múltiples agentes minimiza el tiempo de exploración [2]
- La fabricación de varios robots simples es más barato y sencillo, con respecto a uno con mayor equipamiento, como por ejemplo para el trasporte de cargas, se minimiza la potencia de los actuadores de cada uno [2].
- Los sistemas robóticos, por su cualidad de máquinas, están expuestos a fallas, razón por la que la utilización de múltiples agentes permite enfrentar a una falla de una forma especial, de tal manera que cuando un agente tiene problemas de operación, los restantes pueden suplir su salida del sistema [2].

Pero no todo en los sistemas robóticos multi-agente son ventajas debido a que presentan dificultades como:

- Los sistemas de comunicación son complejos de debido a los tiempos de respuesta de las redes que permiten la interconexión entre los agentes.
- Problemas propios de la robótica móvil como: percepción, localización, navegación, inteligencia, entre otros.
- Y una de las más críticas que es la generación de rutas evitando las colisiones entre los agentes debido a la dinámica natural del sistema.

Existen varias aplicaciones que se puede dar a la robótica móvil cooperativa, por ejemplo:

1.2.1. CONTROL DE FORMACIONES.

Es una aplicación en la que los robots tienen que seguir una trayectoria en base de las órdenes dadas por un maestro de la formación [3], en donde la formación se genera para conducir a la flota a lo largo de un espacio de trabajo evitando colisiones durante la formación como en la navegación. Todo el trabajo se lo hace en base de la percepción sensorial de los agentes. Este tipo de comportamiento genera inconvenientes al momento de mantener la estructura de la formación debido a distintos problemas con los sistemas robóticos [4], como por ejemplo tiempos grandes de respuesta de la comunicación o problemas mecánicos.

1.2.2. TRANSPORTE DE OBJETOS.

Es este tipo de aplicación lo que se busca es que los agentes se encarguen de la movilización de un objeto de un lugar a otro, este movimiento puede ser por sujeción, truncamiento o simple arrastre del objeto, esto se puede apreciar en trabajos como el de Mataric [5], en donde diseño una plataforma robótica que utilizaba sensado, acción y control. Una aplicación de este tipo de aplicación es el mencionado en el trabajo de Edwin Andrés Beltrán Gonzalez [6], en donde dos robots de tipo lego Mindstorms, con configuración de orugas se encargan del transporte de una barra a lo largo de un espacio estructurado, estableciendo una operación de tipo maestro-esclavo, en el cual el maestro tiene una trayectoria establecida y de la misma obtiene la información necesaria para establecer las instrucciones al robot esclavo para el trabajo cooperativo [6].

Una de las aplicaciones más atractivas y comercialmente más difundida de la robótica cooperativa es el campeonato de futbol robótico o RoboCup [7], el mismo que data de 1997 en Japón, y que se constituye en una palestra que mide el desarrollo en áreas como: visión artificial, robótica móvil, inteligencia artificial, trabajo cooperativo y de manera directa su capacidad de hacer, recibir pases y su efectividad para hacer goles.

En la actualidad esta competencia cuenta con una difusión a nivel mundial, con eventos realizados en muchos países entre ellos: China, Alemania, Brazil, Portugal, México, Corea, en las que se muestra la potencialidad de los sistemas robóticos multi agente en cada una de sus categorías como: Soccer y Rescue, además se incluye la categoría @Home, la misma que se enfoca a las aplicaciones de robótica como servicio para asistencia en el hogar.

La categoría F-180, categoría de talla pequeña, consiste en un equipo de robots los mismos que deben caber en un círculo de 180 mm de diámetro y su altura no debe sobrepasar los 150 mm [8] [9], en donde mediante el uso de una cámara cenital se entrega la información al equipo para el procesamiento un nodo central que se encarga de la toma de decisiones [7]. Existen varias categorías más como: Middle Size Robot, humanoides, Aibo, entre otras, en donde la precepción ya sea por visión o demás elementos sensoriales que se encuentran a bordo les permiten la interacción para realizar las tácticas a cada uno de los agentes.

La Robocup tiene gran importancia dentro de los eventos de robótica de competencia a nivel mundial es así que para el 2016 ya está en organización este evento que se llevará a cabo en Alemania, en donde a igual que en ediciones anteriores la competición finaliza con un simposio en donde desarrolladores de este tipo de sistemas comparten sus experiencias en pro de mejorar el nivel de competición año tras año.

Por consiguiente se puede entender que el desarrollo de estos sistemas se ha constituido en la estructuración de líneas de investigación de los centros educativos, como por ejemplo: la Universidad Colombiana Santo Tomás con su equipo STOx's, que está formado por un grupo de docentes y estudiantes, ya tiene varias participaciones en este evento desde el año 2011 y cuentan con robots con tracción omnidireccional de cuatro ruedas con motores brushless, controlados mediante un sistema embebido programado en lenguaje C++ sobre una plataforma de Linux. La comunicación es bidireccional de tipo radio frecuencia a 2.4 GHz [10].

Este tipo de trabajos se enfocan también a doctorados, como por ejemplo lo menciona Miguel Solís, estudiante de doctorado de la Universidad Técnica Federico Santa María en su artículo de la revista “INGENIUM”, en donde se plantearon la creación de un equipo de seis robots para competir en la copa de Brasil 2014. Para lo cual se formó un equipo multidisciplinario entre estudiantes de pregrado y post grado para obtener un sistema autónomo en el que no intervenga la acción humana y sobre todo se pueda investigar el desarrollo de robots cooperativos dentro de entornos competitivos [11]. De esto se puede comprender que el trabajo cooperativo de robots se constituye en una tarea de amplio desarrollo y cargado de bastante trabajo.

A nivel nacional no se encuentran registros de trabajos orientados dentro de este tópico de estudio, es así que el último encontrado es sobre una participación realizada en el evento organizado por la FIRA (Federation of International Robot-soccer Association) [12], por parte de un equipo de la ESPOL (Escuela Superior Politécnica del Litoral) [13]. De esta participación se describe con mucho detalle el trabajo realizado en tres tesis ya que dada su complejidad fue dividido en tres grupos de trabajo en donde cada uno de ellos se encargó de una parte: componente Inteligente [14], componente de visión y componente electro mecánico [15]. Los robots poseen realimentación de la velocidad de sus llantas y al ser de estructuras tipo triciclo ocupan ecuaciones establecidas para este tipo de arquitecturas, de tal forma que cada uno de los móviles poseen velocidades establecidas dentro del diseño que no superen los límites experimentales para evitar que estos se vuelquen cuando realicen curvas [15]. Para identificar a los robots dentro del entorno utilizan cuadrados de colores en su parte superior.

Un investigador ecuatoriano que está relacionado de forma directa con RoboCup es Juan Pablo Mendoza, que es capitán del equipo CMDragons de la Universidad Carnegie Mellon en Estados Unidos. Este estudiante, del PhD en robótica, junto a su equipo quedaron campeones de la edición 2015, en la categoría robots pequeños [16]. Con lo expuesto hasta el momento relacionado con los robots que juegan futbol, se puede notar que el campo de acción en esta área es muy poca en el Ecuador.

Pero el trabajo cooperativo entre robots no es específico de los robots móviles, en el caso de los robots poli articulados industriales también es abordada este tipo de tarea, por ejemplo en *"Diseno y Comparación de Controladores para Robots Cooperativos de Tipo Industrial"* [17], se presenta un trabajo cooperativo entre dos manipuladores de tipo SCARA idénticos, razón por la que se espera un comportamiento idéntico, la estrategia de control es de tipo maestro-esclavo. Se muestra desde el modelamiento, diseño, simulación y comparación entre controladores, ya que se utilizan controladores PD (Proporcionales Derivativos), PID (Proporcionales Integrales Derivativos) y Difusos, así como también las combinaciones entre ellos para el control. En este trabajo no existe visión artificial, la realimentación del sistema es en base de cada una de las tres articulaciones de los robots [17].

Kosuke [18], en su trabajo propone la estimación cooperativa de un grupo de robots móviles, los mismos que están destinados a trabajar en tareas de socorro en una zona amplia, en la que es muy importante que los robots conozcan su posición exacta. Sin embargo, cada robot móvil es imposible que conozca por si solo su posición, para estimarla lo hace a partir de los datos de sus sensores de abordo, estos lo constituyen dos cámaras omnidireccionales y un acelerómetro. Esta información puede ser errada debido al ruido blanco que genera la adquisición de la imagen, para lo cual se propone el uso de Filtros Bayesianos, que en base de procesos estocásticos estima su posición con respecto a sus compañeros cooperativos [18].

Kubota [19], propone un método de para la construcción de tres formaciones de un sistema multi robot para la exploración y vigilancia. De esta manera es posible cubrir un área mucho más grande en búsqueda de víctimas de desastres. En primer lugar, se aplica un método de coordinación de movimientos al sistema multi objetivo para la coordinación de las salidas de un sistema difuso de control que evita las colisiones. Luego se aplica el modelo primavera para calcular la siguiente posición a la que se tiene que mover el robot para mantener la formación establecida. Para finalmente tener el supervisor de la formación para mantenerla. Cuando los robots están en movimiento, la información recopilada es enviada a un monitor en el que

se crean ambientes 2D y 3D. Los agentes que intervienen en este trabajo son denominados MOBiMac, los mismos que están provistos de dos servomotores, ocho sensores y una cámara para captura de imágenes. Hay que destacar que este trabajo esta propuesto como una simulación, mas no existe registro sobre la implementación [19].

Una de las características de los sistemas multi-robots es que permiten alivianar el esfuerzo para realizar una tarea de un solo robot, distribuida en varios agentes, G. Eoh [20] propone en su trabajo la acción cooperativa de varios agentes para el transporte de un objeto con sobrepeso, en donde en estudios previos analiza que el éxito de esta acción está relacionada con la arquitectura y la coordinación, es así que propone una tarea empujador – jalador en la cual varios agentes empujan y otros jalan del objeto para lograr la tarea. Obtiene como resultado que este tipo de formación es más robusta en el caso de tareas en línea recta y simétrica.

Otra aplicación para la robótica cooperativa es el recate y vigilancia, enfocándose en esta segunda acción es que Acevedo [21] presenta una propuesta de control robusto de robots para la vigilancia de un perímetro con velocidades heterogéneas bajo restricciones de comunicación basado en la frecuencia. Establece para la vigilancia una estrategia denominada ruta-partición en la cual los robots patrullan un segmento del perímetro relacionado con sus capacidades e intercambian información entre ellos periódicamente. La simulación y los resultados experimentales realizados con un robot demuestran la robustez de la solución.

En el trabajo mostrado por Modesti [22], en el mismo se muestra un sistema de visión artificial para que sirva de apoyo en el guiado de un AGV (Vehículo de Guiado Automático), se apoya en la utilización de visión estereoscópica para medir la distancia en base de la visión artificial similar al procesamiento que hace un sistema de visión de un ser humano. Al sistema cognoscitivo lo denominan como inteligencia visual debido a que se apoya en técnicas neuronales, celda perceptrón, para la búsqueda de la trayectoria que está trazada en donde mediante la ubicación de figuras geométricas se establecen los puntos de parada, la detección de obstáculos y la medición de la distancia.

Rosales [23] en su trabajo doctoral plantea el control dinámico de sistemas robóticos mediante la utilización del Algebra Lineal y los Métodos Numéricos como medio de resolución de los sistemas de control debido a la baja carga computacional que genera este tipo de matemática. Se enfoca en la navegación en interiores, en donde los controladores proporcionan leyes de control en particular para las velocidades lineales y angulares del robot. Desarrolla un controlador dinámico para garantizar la estabilidad del sistema al momento de seguir las trayectorias de los sistemas no holonómicos, para seguido de eso se utilizan estos resultados para la formación de sistemas multi-robot, con un sistema de control descentralizado el mismo que tiene un comportamiento reactivo ante la detección de obstáculos para la evasión de los mismos. Como aporte de este trabajo se tiene que el sistema de control basado en el Álgebra Lineal y los métodos numéricos permiten a un robot o una formación de robots desplazarse por una trayectoria con una buena respuesta de acciones de control.

1.3. PLANIFICACIÓN DE TRAYECTORIAS

La planificación de las trayectorias básicamente consiste en llegar desde un punto a otro evadiendo los obstáculos que encuentre en su movimiento, es así que para robots móviles multi agente siempre ha sido un gran reto debido a que de una buena planificación depende un buen movimiento, es así que lograr que los sistemas robóticos se desplacen sin colisionar es uno de los serios problemas dentro de los sistemas multi agente.

La idea general en la planificación de trayectorias consiste en desvincular el problema de la cinemática y dinámica del robot de la obtención de una ruta (óptima o no) libre de obstáculos [24].

Para la búsqueda de trayectorias lo primero es establecer los puntos o nodos por donde va a pasar el robot durante su movimiento para esto se puede aplicar grafos de visibilidad, campos potenciales, en esencia se tiene que asignar a cada nodo un peso que lo identifique dentro del espacio [25], con esta información basta con el cálculo de la trayectoria más corta entre los nodos para lo cual se puede utilizar un algoritmo de búsqueda en grafos como por ejemplo A*, Dijkstra.

A más de los métodos citados, existen otros que también han sido utilizados para encontrar un camino libre de colisiones, estos métodos son, por ejemplo: Métodos de Visibilidad Gráfica, Diagramas de Voronoi, Freeway y Silhouette. [26].

A* es un algoritmo heurístico de búsqueda del camino más corto, data de 1998 cuando Peter E. Hart, Nils J. Nilsson y Bertram Raphael, cuando sus diseñadores lo plantearon, con el único inconveniente de la carga computacional requerida.

En el trabajo de Yandún [27], se presenta un estudio de diferentes métodos para la planeación de trayectorias para un Robotino de Festo, con entorno de desarrollo en LabView. Los entornos son considerados fijos y establecidos, con lo cual el sistema se limita a conocer su posición inicial y final. Se establece que *"La navegación es la técnica de conducir un robot móvil mientras atraviesa un entorno para alcanzar un destino o meta sin chocar con ningún obstáculo"* [28], se utiliza dentro del desarrollo grafos de visibilidad que se establecen entre dos puntos que puedan ser unidos mediante una línea recta. El siguiente método son los Diagramas de Voronoi, el mismo que se encarga de ubicar puntos centrales, lugares geométricos, en espacios libres por donde podrían generarse las trayectorias. Otro método es Roadmap Probabilístico (PRM), en el cual se van generando n cantidad de trayectorias aleatorias uniformes libres de colisión. Cada uno de los métodos mostrados se encargan de crear la red de nodos, a la que aplica el algoritmo A* o A estrella para determinar la trayectoria con menor cantidad de coste para que el robot llegue desde el punto A hasta el punto B. Se citan varios métodos adicionales como por ejemplo: Modelado del Espacio Libre, Descomposición en celdas, Campos Potenciales. En las conclusiones de este trabajo se establece que los Diagramas de Voronoi es uno de los mejores métodos para la planificación de trayectorias, ya que la trayectoria generada es la más alejada de los obstáculos.

CAPÍTULO 2

2.1. MARCO TEÓRICO

En este capítulo se dará un enfoque de la teoría que se ha utilizado para el desarrollo de esta tesis, debido a que para la consecución del mismo se han relacionado varias disciplinas, como la robótica móvil, los microcontroladores, electrónica de potencia, comunicaciones, inteligencia artificial.

2.2. ROBÓTICA

La robótica es una ciencia que está relacionada con la construcción de máquinas para satisfacer necesidades del ser humano, por esta razón es que integra a la ingeniería mecánica, ingeniería eléctrica, ingeniería electrónica e ingeniería de software, en donde la interrelación entre cada una de ellas permite satisfacer el diseño, la construcción, la operación, la disposición estructural, la manufactura y las aplicaciones para los robots.

2.2.1. CLASIFICACIÓN GENERACIONAL DE LA ROBÓTICA

La robótica ha ido evolucionando a lo largo del tiempo es así que se tienen varias generaciones para dar una clasificación [29]:

2.2.1.1. Primera generación.

Los constituyen manipuladores simples que son sistemas mecánicos multifuncionales con un sistema de control básico en lazo abierto de secuencia fija o de secuencia variable, no existe una realimentación del entorno.

2.2.1.2. Segunda generación.

En esta los robots disponen de sistemas que les permiten el aprendizaje de secuencias previamente ejecutadas por un operador humano y la apreciación de su entorno, es decir el operador realiza movimientos y el robot las memoriza y repite. El sistema posee control en lazo cerrado.

2.2.1.3. Tercera generación.

Los robots ahora disponen sistemas de sensores que les permiten conocer el estado de su entorno para así mediante un sistema computacional más sofisticado, procesar un algoritmo para después ejecutar los resultados del mismo.

2.2.1.4. Cuarta generación.

En esta el sistema de sensado es en tiempo real, esto se logra debido al avance de los sistemas computacionales que permiten un mayor procesamiento de la información y que la toma de decisiones sean constantes a lo largo del tiempo de ejecución.

2.2.1.5. Quinta generación.

Esta generación estará conformada pos sistemas robóticos equipados con gran cantidad de inteligencia artificial y con modelos de comportamiento que se asemejan al comportamiento biológico, en nano estructuras.

2.2.2. CLASIFICACIÓN EN BASE DE LA ARQUITECTURA DE LA ROBÓTICA

Otra clasificación que se puede dar a los sistemas robóticos es en base de su estructura o arquitectura de la siguiente manera [29] [30]:

2.2.2.1. Poli articulados.

Este tipo de robots están estructurados de muy variadas formas, en los que su característica principal es el sedentarismo, es decir se mantienen en un plano estático y su estructura les permite movimientos limitados por su estructura y grados de libertad. Dentro de este grupo se encuentran los robots cartesianos, industriales, manipuladores, Scara, entre otros.

Robot Industrial Poli Articulado [31].

Figura 2-1 Robots poli articulados

2.2.2.2. Móviles.

Estas máquinas están dotadas de gran capacidad de desplazamiento, están basados en carros o plataformas que poseen sistema locomotor rodante. Estos garantizan el transporte de elementos desde un punto a otro y pueden ser guiados de forma remota o mediante recolección de información de su entorno mediante sensores.

Robot Móvil con Ruedas [32].

Figura 2-2 Robot móvil

2.2.2.3. Androides.

Estos robots tratan de reproducir total o parcialmente el comportamiento del ser humano. Su desarrollo ha sido muy limitado y sin utilidad práctica, razón por la que fundamentalmente se encuentran dedicados para el estudio y experimentación.

Robot Humanoide [33].

Figura 2-3 Robot humanoide Nao

2.2.2.4. Zoomórficos.

Estos robots en cambio tratan de reproducir el comportamiento de los animales teniendo de esta manera una subdivisión de caminadores y no caminadores.

Robot con Patas. [34]

Figura 2-4 Big Dog. robot zoomórfico.

2.2.2.5. Híbridos.

Este tipo de robot se constituyen en mezclas de los antes citados.

Robot con Orugas y Brazo Articulado [35].

Figura 2-5 iRobot 510 PackBot, robot híbrido

En esta tesis se trabaja con robótica móvil, debido a que los agentes que se crearon están basados en dos pequeños montacargas que permiten movimientos en base de tracción diferencia, que mediante la automatización se convirtieron en dos robots con características de robots móviles y con un sistema de control de tercera generación con un computador central como maestro del sistema de control descentralizado.

2.2.3. CLASIFICACIÓN EN BASE DE LA UBICACIÓN DE LA INTELIGENCIA

Un punto muy importante por definir en los sistemas robóticos es la ubicación del sistema inteligente, cuando la inteligencia está ubicada en el mismo robot se los conoce como *"robots con inteligencia en el robot"* y cuando el proceso se análisis está ubicado en un sistema procesado externo, se los conoce como "robots sin inteligencia en el robot", ahora si la inteligencia se encuentra compartida se los conoce como "robots de inteligencia parcial" [14].

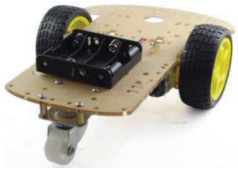
2.3. ESTRUCTURAS ROBÓTICAS MÓVILES CON RUEDAS

2.3.1. COMPONENTES

Este tipo de robots están constituidos básicamente por lo elementos listados a continuación los mismos que son la base de una estructura robótica móvil estándar:

2.3.2. ESTRUCTURA.

También suele ser llamado chasis y lo constituye el elemento de soporte de todos los componentes, además se incluye un sistema de trasmisión mecánica para mejorar el torque de las llantas. Figura 2-6.



Estructura triciclo para robot [36].

Figura 2-6 Chasis comercial de robot móvil.

2.3.3. ACTUADORES.

Lo constituyen básicamente motores de corriente continua, además pueden ser considerados como actuadores a cualquier tipo de elemento que permita la interacción con el entorno, por ejemplo: luces, brazos, parlantes, llantas etc. Figura 2-7. En el caso de actuadores especiales se deberán utilizar circuitos de potencia o drivers.

Motores para tracción [37].

Figura 2-7 Motores para robots móviles.

2.3.4. SENSORES.

Son los encargados de la precepción del entorno para lo cual se basan en variados principios físicos, por ejemplo: sonares, infrarrojos, de contacto, cámaras, etc. Figura 2-8.

Figura 2-8 Algunos sensores utilizados en robots móviles.

2.3.5. INTELIGENCIA.

Lo constituye un computador de abordo en una tarjeta de desarrollo. En este elemento es en donde el diseñador programará los algoritmos necesarios para que el robot cumpla con las condiciones de diseño para él establecidas. Figura 2-9.

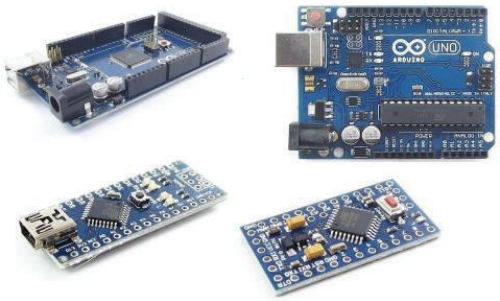


Figura 2-9 Algunos controladores para robots

2.4. ROBÓTICA COOPERATIVA.

La robótica cooperativa es una rama de la robótica general que se encarga del análisis, estudio y ejecución de sistemas robóticos que trabajan de forma coordinada para realizar una tarea. Presenta varias ventajas con respecto a los sistemas robóticos individuales, por ejemplo:

- Pueden realizar tareas imposibles para sistemas de un solo agente.
- Aumenta la eficiencia y eficacia de la ejecución de una tarea.
- Los agentes, estructuralmente son más simples.
- Pueden suplir entre varios agentes la ausencia de uno.

Pero también presentan desventajas como:

- Sistemas de comunicación complejos.
- Problemas propios de la robótica como inteligencia, percepción, etc.
- Evitar las colisiones entre agentes al realizar la tarea.

Para los sistemas de robóticos se puede establecer dos formas de control:

2.4.1. CONTROL CENTRALIZADO

En esta forma de control se establece la existencia de una unidad central la misma que se encarga de la toma de decisiones. Para este fin los agentes adquieren la

información del entorno, la envían a la unidad central, la misma que la procesa y define las acciones a realizarse en el sistema.

Figura 2-10 Esquema control centralizado.

Tiene la ventaja que la unidad central dispone de mucha información para la toma de decisiones, pero esto también puede convertirse en un inconveniente ya que el gran volumen de información puede congestionar al sistema y de esta forma generando retardos en la obtención de las acciones de control. Otro problema que presenta es que, al existir problemas en la unidad central se afecta a todo el sistema [38]. Figura 2-10.

2.4.2. CONTROL DESCENTRALIZADO.

En esta forma de control cada uno de los agentes tiene la capacidad de tomar sus propias decisiones sin ninguna intervención externa. Esta forma de control es muy dinámica para el sistema, permitiendo que la ejecución sea a mucha velocidad ya que no es necesaria la transmisión y almacenaje de información. Al estar la inteligencia del sistema distribuida en cada uno de los agentes, esto le permite que si un agente falla, el sistema no se paraliza [38]. Figura 2-11.

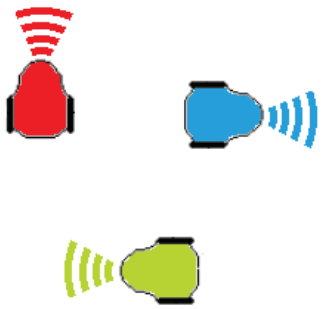


Figura 2-11 Esquema de control descentralizado.

En base de la interacción que existe entre los agentes se pueden tener dos tipos de sistemas:

2.4.3. MUY ACOPLADOS O COLECTIVOS.

Para este tipo de sistemas los agentes están en la capacidad de comunicarse entre sí de tal forma que pueden compartir información sobre sus acciones, de esta manera adicionada a su autonomía e inteligencia son capaces de pedir ayuda y coordinar movimientos con los demás agentes.

La inteligencia del sistema colectivo es igual a la sumatoria de las inteligencias parciales de cada uno de los agentes [38]. Figura 2-12.

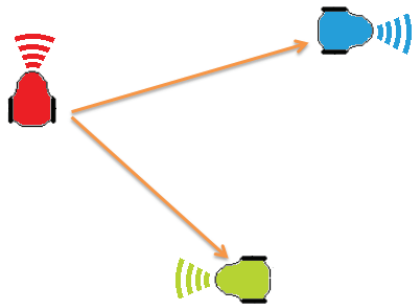


Figura 2-12 Sistema muy acoplado o colectivo.

2.4.4. POCO ACOPLADAS O MANADAS

En este caso cada uno de los agentes realizan sus acciones sin que exista interacción entre ellos, es decir cada uno funciona de manera independiente y regido por sus propias reglas de operación [38]. Figura 2-13.

Figura 2-13 Sistema poco acoplado o manada.

2.5. INTELIGENICA ARTIFICIAL

Basándose en la formación de la palabra: ***"Inteligencia es la capacidad o facultad de entender, razonar, saber, aprender y de resolver problemas"*** [39] y ahora ***"Artificial es todo aquello que no es creación de la naturaleza sino elaboración humana"*** [40], de estas definiciones se puede asegurar que es la capacidad de permitirle a una creación humana la capacidad de entender, razonar, saber y aprender para resolver problemas.

La inteligencia artificial (IA) es una ciencia multidisciplinaria que integra a las ciencias de la computación, las neurociencias, la lógica, la matemática, la física, la filosofía, entre otras, para la creación de sistemas capaces de resolver problemas ya sean cotidianos o enfocados a aspectos específicos utilizando como base el comportamiento del cerebro humano.

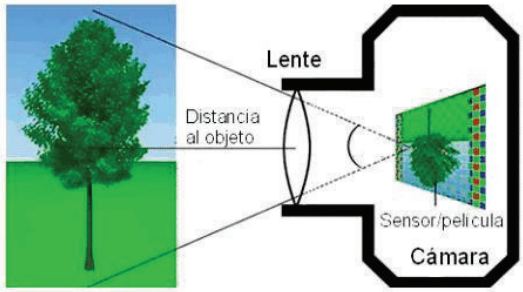
2.5.1. VISIÓN ARTIFICIAL

La visión artificial es una parte de la inteligencia artificial encargada de comprender el entorno enfocándose en el análisis automático de imágenes en base de sus propiedades, ya sean estas geométricas como por ejemplo forma, tamaño o de las características de los materiales de los elementos que las conforman, como por ejemplo el color, textura.

Uno de los grandes problemas que se tiene con la visión artificial es que la adquisición de las imágenes es en dos dimensiones y el medio circundante a los agentes, que utilizan esta técnica de inteligencia artificial, está en tres dimensiones por esta razón es que en muchos casos los resultados obtenidos poseen grandes rangos de incertidumbre, esto se puede solucionar con la adquisición de varias imágenes o con la formulación de algoritmos que minimicen la incertidumbre, pero todo esto confluye en una mayor carga computacional, razón por la que se tiene que llegar a un equilibrio entre la incertidumbre y la carga computacional que requiera el algoritmo a desarrollarse.

2.5.2. CÁMARA

En todo sistema que involucre la relación entre dos entornos diferentes, como en este caso: el entorno físico con el sistema electrónico, es necesaria la utilización de un elemento primario, que haga la misma función de un ojo humano. Para este caso deberá capturar la imagen, para después digitalizarla y poder entregarla en forma de datos, para este efecto se utiliza una cámara la misma que en su estructura posee los componentes necesarios para entregar la imagen como una matriz de datos con la información del entorno.



Operación de la cámara [41]

Figura 2-14 Esquema de captura de una cámara

En donde a la imagen se la puede cuantificar por su tamaño, el mismo que es medido en pixels, acrónimo de picture element, de acuerdo a la cantidad que ésta matriz tenga en alto y ancho. Un pixel corresponde a un punto de elemento de color que conforma la imagen.

2.5.3. TIPOS DE IMÁGENES.

Las imágenes a nivel computacional sor representadas en forma de matrices bidimensionales, pero existen varios tipos de imágenes, así por ejemplo [42]:

2.5.3.1. Imagen RGB.

Es una imagen a color conformada por la tricromía de tres matrices en donde cada una de ellas contiene la información de cada uno de los componentes básicos de color: rojo, verde, azul; debido a que de las combinaciones de cada uno de ellos se pueden formar los distintos colores. Figura 2-15.

2.5.3.2. Imagen CMYK.

Es una imagen de color formada por una cuatricromía de cuatro matrices en conde cada una tiene los componentes de color Cyan, Magenta, Yellow y Key (negro), de la combinación de cada uno de los componentes se forman los distintos colores. Figura 2.-15.

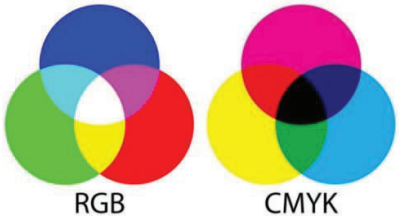


Figura 2-15 Componentes RGB y CMYK

2.5.3.3. Imagen Gris.

Ésta en cambio es una imagen formada por una sola matriz en donde cada uno de sus elementos puede tomar valores de 0 a 255, en donde los distintos valores en cada uno de los pixels dan las distintas tonalidades de gris para estructurar la forma de la imagen. Figura 2-16.

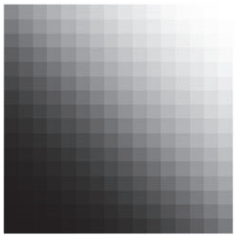


Figura 2-16 Imagen escala de grises

2.5.3.4. Imagen Binaria.

Ésta en cambio en una imagen representada por una sola matriz, pero con la peculiaridad de que cada uno de los pixeles toman valores discretos de ceros o unos, de tal forma que se tiene una imagen en blanco y negro. Figura 2-17.

Figura 2-17 Imagen Binaria

En base de lo hasta aquí expuesto se puede definir a una imagen como una función de dos dimensiones *f(x,y)*, en donde los valores de x e y, definen las coordenadas del pixel y su valor determinará las características del mismo, esto para el caso de imágenes en escalas de grises y binarias. Para el caso de imágenes RGB y CMYK, la función ahora se definirá *f(x,y,m)*. Dada esta condición, a la imagen se la podrá definir espacialmente de la siguiente forma:

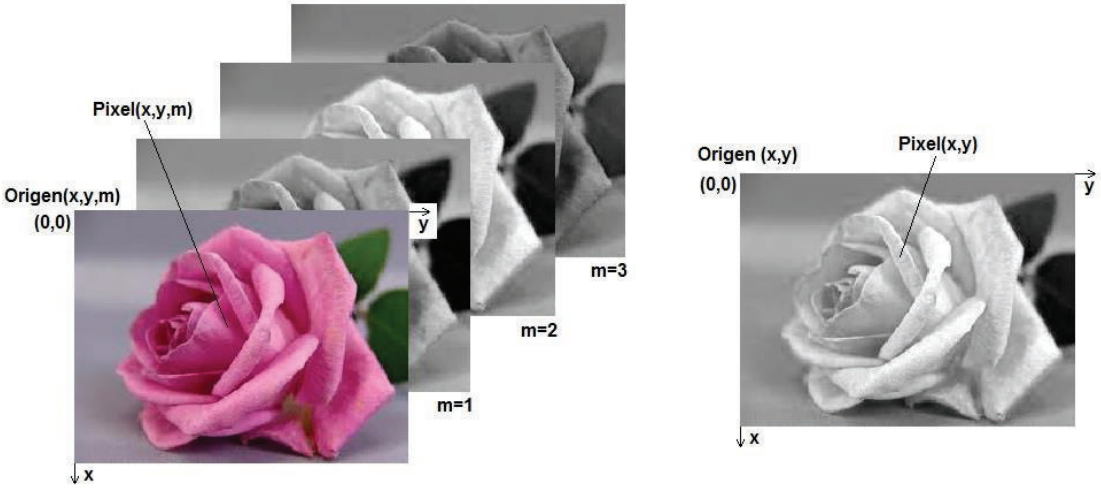


Figura 2-18 Descripción de las coordenadas de las imágenes

2.5.4. ILUMINACIÓN

La iluminación es un factor extremadamente importante dentro de la visión artificial, debido a que al mantenerla constante permite que las imágenes que son capturadas presenten condiciones estables que facilita el desarrollo de los algoritmos de procesamiento. Por ejemplo, de una mala iluminación se pueden obtener imágenes que presenten sombras y al momento de ser procesadas éstas sean consideradas como parte de la escena, más no como un defecto de la aplicación de la luz.

Existen varias formas de iluminar un entorno, entre ellas se tiene [43]:

2.5.4.1. Iluminación direccional.

Consiste en la aplicación directa de un haz de luz focalizado sobre el objeto, el problema de este tipo de iluminación es que dependiendo de su punto de

localización de la fuente de luz puede generar sombras por los objetos colocados en el entorno. Figura 2-19.

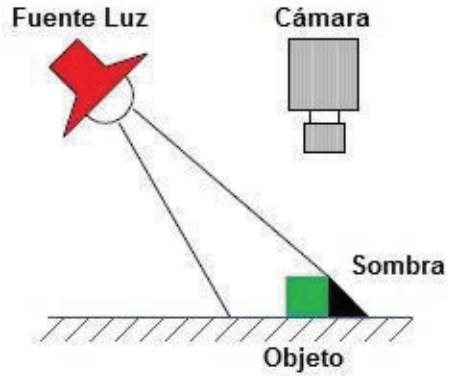


Figura 2-19 Sistema de iluminación directa

2.5.4.2. Iluminación difusa.

Es este tipo de iluminación se tiene la intervención de varias fuentes luminosas las mismas que se encuentran distribuidas a lo largo de todo el entorno permitiendo que se eliminen las sombras en la imagen adquirida, en este caso la luz puede verse reflejada en las paredes del entorno, normalmente se utiliza como fuentes de luz a sistemas fluorescentes, luces halógenas, luces incandescentes, entre otras. Figura 2-20.

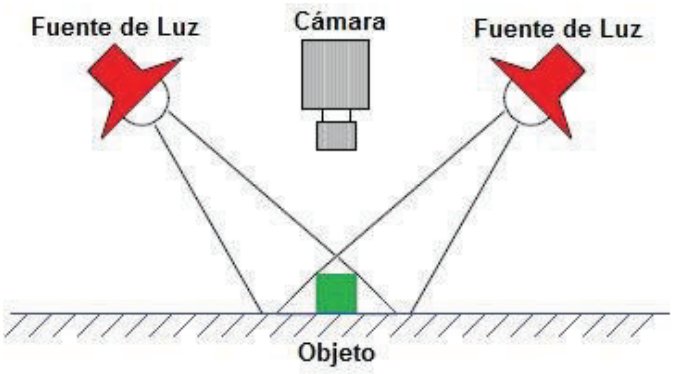


Figura 2-20 Sistema de Iluminación difusa.

Existen otras formas de iluminación, pero no se las analiza debido a que no son aplicables para el caso en estudio.

2.6. OPERACIONES CON IMÁGENES

Con las imágenes se pueden realizar distintos tipos de operaciones, debido a que en su esencia matemática son matrices. A continuación, se enumeran las operaciones que fueron utilizadas en el desarrollo de este trabajo:

2.6.1. CONVERSIÓN RGB A GRIS

Esta es una operación básica, la misma que se permite la conversión de las tres matrices en RGB a una sola matriz y ésta en escalas de grises, en la que cada pixel posee valores de 0 a 255, para esto se aplica la siguiente ecuación:

$$\boldsymbol{Griz(x,y) = 0.2989 * R(x,y) + 0.5870 * R(x,y) + 0.1140 * G(x,y)} \qquad (2.1)$$

Estos valores se han sido determinados experimentalmente en base de la percepción humana hacia los tres componentes de color.

2.6.2. CONVERSIÓN RGB A BINARIA.

Esta operación también es conocida como binarización, umbralización o cuantificación y básicamente consiste en comparar los niveles de gris existentes en una imagen con un valor de umbral definido o predeterminado, de tal forma que si el nivel de gris es menor o igual que el umbral se le asigna al pixel un valor de cero (0) negro, caso contrario si es mayor se le asigna un valor de uno (1) blanco.

$$\boldsymbol{g(x,y) = \begin{cases} 1 & f(x,y) > unbral \\ 0 & f(x,y) \leq umbral \end{cases}} \qquad (2.2)$$

Este proceso puede también ser inverso, dependerá de la necesidad que se tenga para la binarización, o en todo caso se puede mantenerlo bajo la lógica planteada y después hacer el complemento o negación de la imagen.

2.6.3. ETIQUETADO DE IMÁGENES

La operación de etiquetado está basada en criterios de similitud y continuidad entre los pixels que forman una región, de tal forma que se considera que la imagen estará formada por una cantidad de regiones que están disjuntas, aisladas unas de

otras. La forma de separación de las áreas puede estar basada en características propias de la imagen, como por ejemplo el color.

Las condiciones que rigen la asociación para la segmentación en base de regiones son las siguientes:

$$\boldsymbol{I} = \cup_{i=1}^{n} \boldsymbol{R_i} \tag{2.3}$$

$$\boldsymbol{R_i} \cap \boldsymbol{R_j} = \emptyset \quad \boldsymbol{i} \neq \boldsymbol{j} \tag{2.4}$$

$$\boldsymbol{P_i} \in \boldsymbol{R_j} \leftrightarrow \boldsymbol{P_{R_j}}(\boldsymbol{P_j}) = \mathbf{1}, \quad \textbf{caso sontario será } \mathbf{0} \tag{2.5}$$

En donde I es la matriz binarizada original, R_i una región de la imagen, P_i un pixel cualquiera. P_{RJ} la regla se similitud de la región.

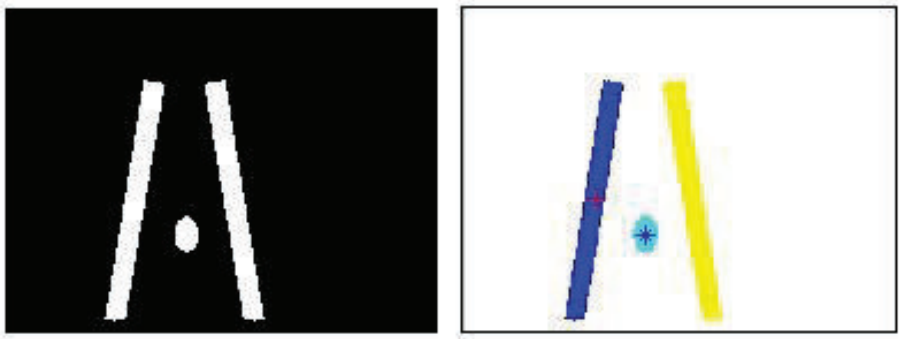


Figura 2-21 Imagen binarizada e imagen segmentada por área

Una vez segmentada a la imagen en cada área a esta se le asigna una etiqueta para que se pueden realizar operaciones como: la contabilización de los píxels de cada región para determinar el área, de los píxels más distantes se puede calcular los extremos de la región, calcular el centroide, entre otras. [44, pp. 28-48]

2.6.4. FILTRADO DE IMÁGENES

El filtrado de una imagen (f) consiste en aplicarte una transformación (T) para obtener una nueva imagen (g) que tiene acentuadas o disminuidas ciertas características.

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{T}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})] \tag{2.6}$$

En base de la teoría de sistemas el pasar una señal por un sistema lineal, la salida es la convolución entre la función de transferencia del sistema con la señal de entrada, aplicado para el caso espacial se tendría:

$$\boldsymbol{g(x,y) = h(x,y) * f(x,y)} \qquad \textbf{(2.7)}$$

En forma más explícita se tendrá:

$$\boldsymbol{h(x,y) * f(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} h(m,n) f(x-m, y-n)} \qquad \textbf{(2.8)}$$

en donde (x) tomará valores desde 0,1,2,…M y (y) tomará valores desde 0,1,2 …..N.

De la ecuación 2-8, se puede apreciar que la operación consiste en ir desplazando la matriz (h) a lo largo de toda la matriz (f) y realizando operaciones pixel a pixel, obteniendo como resultado el pixel central que es el modificado por el filtro, esto se aprecia en la Figura 2-22.

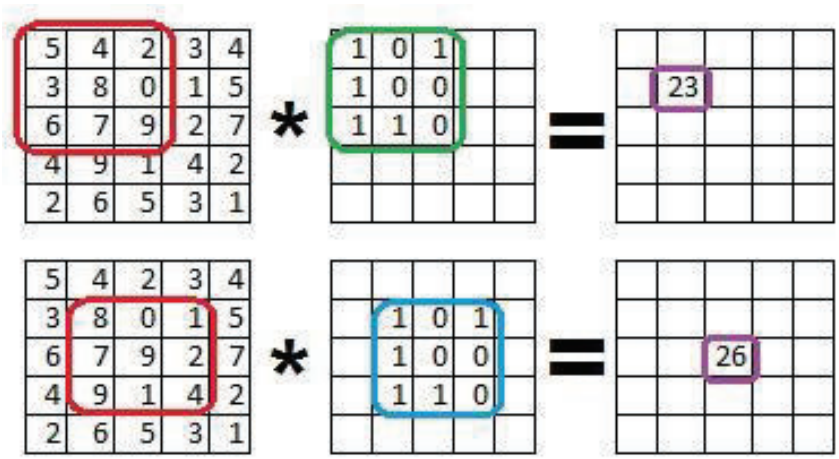


Figura 2-22 Forma de ejecución de la convolución.

2.6.5. FILTRO MEDIANA

Este filtro forma parte de los filtros de suavizamiento al igual que los filtros media aritmética o gaussiano y sustituye el pixel central por la mediana de los pixeles en el vecindario. Para esto se utiliza la siguiente máscara:

$$\boldsymbol{h} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \qquad (2.9)$$

Con esta máscara se obtiene los nueve valores, se los ordena de mayor a menor, se obtiene el central (número 5), el mismo que correspondería al valor del pixel central de la aplicación del filtro.

Este filtro tiene las siguientes características:

- Ideal para imágenes con poco ruido.
- Elimina ruido sal pimienta, puntos blancos negros.
- Reduce efecto borroso de la imagen.
- Elimina pequeños detalles.

2.6.6. SEGMENTACIÓN DE IMÁGENES POR COLOR.

Para este procedimiento se parte de una imagen RGB por consiguiente se tiene tres matrices en donde cada una posee los componentes de cada color, es decir una matriz de rojo, otra de verde y una tercera de color azul. Al parecer bastaría con esto para obtener los componentes de color rojo, por ejemplo, pero hay que recordar que un pixel en color tiene componentes de los tres colores (R, G, B) para obtener las distintas tonalidades, esto se precia en la Figura 2-23.

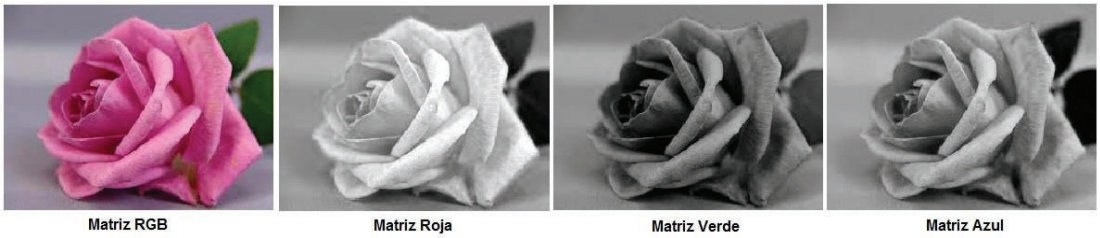


Figura 2-23 Componentes de color de la imagen RGB

Por lo tanto, un pixel de color tendrá componentes en cada una de las matrices RGB, claro que existirá un mayor componente en matriz que más se aproxime al tono de color del pixel. Las matrices de componentes RGB tienen sus elementos con valores de 0 a 255, razón por la que al momento de la separación en colores también se tiene que aprovechar para convertir a los elementos de la matriz a una resolución de tipo doble para posteriormente no perder información de los cálculos realizados, debido a que se obtendrán valores fraccionarios.

Para iniciar este algoritmo primero se restan pixel a pixel del componente que se busque los demás de esta manera se tiene una matriz en donde quede el máximo componente. Por ejemplo, se analiza para la extracción de color rojo.

$$\boldsymbol{Im_{RBG}} = \begin{cases} \boldsymbol{Im_R} \\ \boldsymbol{Im_G} \\ \boldsymbol{Im_B} \end{cases} \tag{2.10}$$

$$\boldsymbol{Im_{\mathrm{mr}}} = \boldsymbol{Im_R} - \boldsymbol{Im_G} - \boldsymbol{Im_B} \tag{2.11}$$

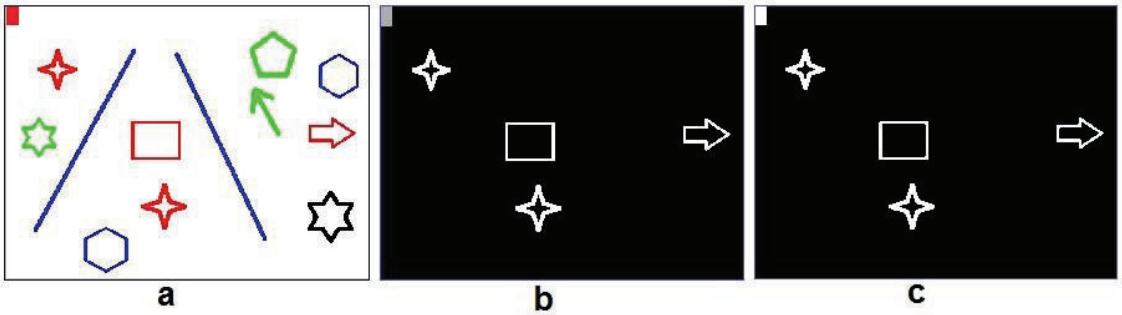
Se tiene que crear una máscara en base de la imagen máximo rojo Im_{mr}, para el caso se le aplica la siguiente fórmula:

$$\boldsymbol{masc} = \boldsymbol{Im_{mr}}/5 \tag{2.12}$$

Se multiplica la máscara por la matriz de máximo rojo para acentuar los componentes de color rojo, esta operación pixel a pixel.

$$\boldsymbol{Im_{mr}} = \boldsymbol{masc}.* \boldsymbol{Im_{mr}} \tag{2.13}$$

Con esta matriz se aplica un filtro mediana y se tiene únicamente el componente de color rojo, esta segmentación se la observa en la Figura 2-24.



a) Imagen original RGB, b) Matriz máximo, c) Matriz filtrada.

Figura 2-24 Segmentado color rojo

Lo hasta aquí expuesto correspondería para el color rojo, ahora en caso de los otros componentes como verde o azul, el procedimiento sería el mismo, pero las ecuaciones de matriz de imagen máximo cambiarían a:

$$\boldsymbol{Im}_{\mathrm{mg}} = \boldsymbol{Im}_{\boldsymbol{G}} - \boldsymbol{Im}_{\boldsymbol{B}} - \boldsymbol{Im}_{\boldsymbol{R}} \qquad (2.14)$$

$$\boldsymbol{Im}_{\mathrm{mb}} = \boldsymbol{Im}_{\boldsymbol{B}} - \boldsymbol{Im}_{\boldsymbol{G}} - \boldsymbol{Im}_{\boldsymbol{B}} \qquad (2.15)$$

2.6.7. BÚSQUEDA DE IMÁGENES O PATRONES

Esta técnica es también conocida como Template Matching, la misma que consiste en los siguientes pasos:

- Capturar la imagen del entorno y binarizarla.
- Cargar la imagen plantilla y binarizarla.
- Realizar la correlación cruzada normalizada, de esta acción se obtiene una nueva matriz del mismo tamaño que la matriz del entorno, en donde cada uno de los elementos tienen valores entre -1 y 1.

  La ecuación 2-16, explica matemáticamente esta operación:

$$\boldsymbol{C}_{\boldsymbol{cc}} = \sum_{\boldsymbol{i}=\boldsymbol{M}}^{\boldsymbol{M}} \sum_{\boldsymbol{j}=\boldsymbol{M}}^{\boldsymbol{M}} \left\{ \frac{[\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i}},\boldsymbol{y}_{\boldsymbol{j}}) - \boldsymbol{f}_{\boldsymbol{m}}] * [\boldsymbol{g}(\boldsymbol{x'}_{\boldsymbol{i}},\boldsymbol{y'}_{\boldsymbol{i}}) - \boldsymbol{g}_{\boldsymbol{m}}]}{\Delta \boldsymbol{f} \Delta \boldsymbol{g}} \right\} \qquad (2.16)$$

  De la ecuación 2-16, se puede observar que la plantilla recorre por toda la imagen de entorno realizando el cálculo de la correlación.

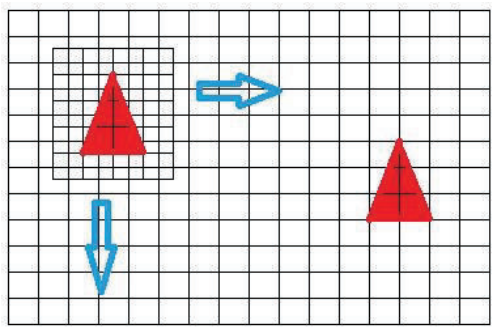


Figura 2-25 Desplazamiento de la plantilla

- En la matriz de correlación cruzada normalizada, se encuentra el elemento que tenga mayor valor, dado que mientras más parecido este el fondo (imagen

entorno), coincida con la plantilla ese pixel corresponderá al punto central de la imagen buscada.

2.6.8. REDUCCIÓN DE IMAGEN

Para este trabajo se ha creado un algoritmo para la compresión de la imagen del entorno debido a que en su tamaño original tiene dimensiones de 640x480 pixels y esta cantidad de elementos dentro de una matriz obliga a una carga computacional extremadamente alta, sobre todo en lo referente a la búsqueda de del camino más corto para el movimiento de los agentes.

A este algoritmo se lo denominado como reducción por áreas y consiste en crear un barrido a lo largo de todo el entorno contabilizando la cantidad de pixels dentro del espacio de muestra (cálculo del área), si el área calculada está dentro de un umbral se asigna a una nueva matriz (imagen comprimida) como espacio vacío un cero, caso contrario se lo considera como lleno y se le asigna un uno, razón por la que esta nueva matriz también es de tipo binario.

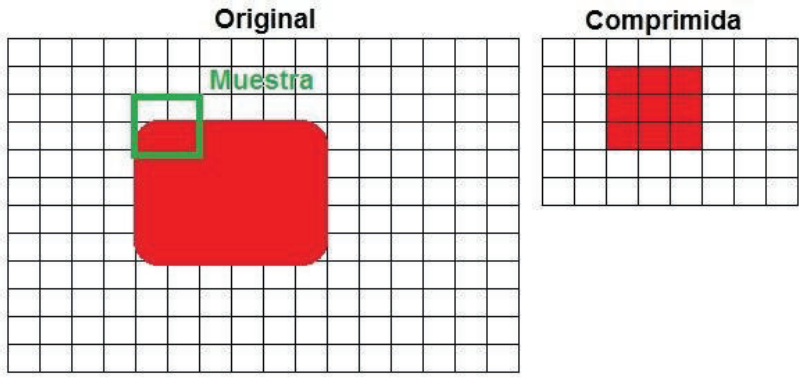


Figura 2-26 Compresión por áreas

2.6.9. DILATACIÓN

A esta acción forma parte de dos operaciones, las mismas que se las conoce como operaciones morfológicas, estas son: la dilatación y la erosión. Operan con imágenes binarias y en el caso particular de este trabajo de grado se ha utilizado la dilatación con el fin de crear un espacio de holgura entre el movimiento de los móviles y los límites del espacio estructurado para evitar colisiones con el entorno,

ya que los agentes poseen las horquillas las mismas que definen una distancia entre el punto de movimiento de la ruta y el entorno.

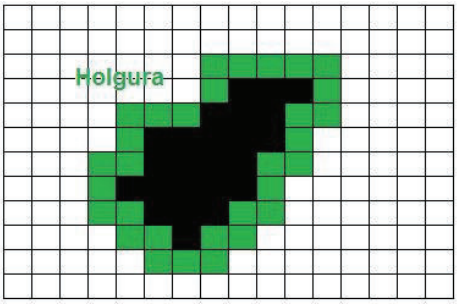


Figura 2-27 Imagen dilatada

2.7. ALGORITMO A*

Este algoritmo heurístico y forma parte de los algoritmos de búsqueda en grafos para buscar caminos evitando colisiones, fue presentado por primera vez el 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael y tiene como objetivo determinar el camino de menor coste entre un punto de origen y un punto de destino [45]. Este algoritmo es flexible y eficiente, encuentra una solución siempre y cuando esta exista, de esto se deduce que da fiabilidad y estabilidad al momento de calcular la ruta [46]. Este algoritmo se encargará de encontrar la ruta más corta desde un punto A hacia un punto B con información obtenida del sistema de visión artificial.

El algoritmo A* funciona de la siguiente manera:

- Se inicia con la adquisición del entorno y dentro este se tienen que ubicar perfectamente a: punto de origen, punto final o punto de destino, cada uno de los obstáculos que impiden el libre movimiento. esto se puede apreciar en la Figura 2-28.

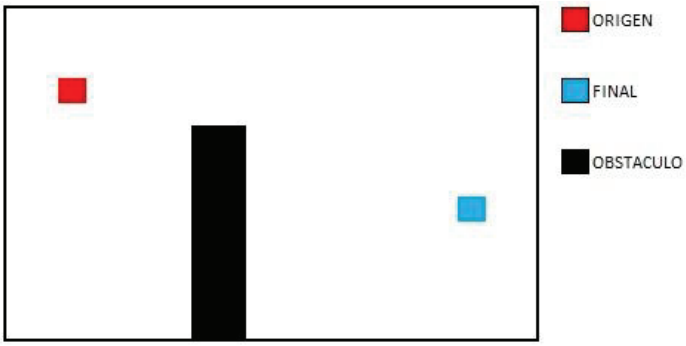


Figura 2-28 Entorno de trabajo

- Lo siguiente es representar a imagen como una matriz binaria en la que para cada espacio libre del movimiento se le asigna un valor de cero y para cada espacio ocupado se le establece un valor de uno, de esta manera se tienen perfectamente ubicados los espacios para el movimiento. El resultado de esta operación se lo puede apreciar en la Figura 2-29, nótese que cada uno de los pixeles tiene un valor discreto asignado.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

ORIGEN

FINAL

OBSTACULO

Figura 2-29 Imagen binarizada del entorno

- Cada uno de los valores de cero se constituyen en posibles nodos para la ruta del movimiento del camino más corto, razón por la que tienen que ser etiquetados, a esta nueva matriz se la conoce como matriz indexada.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	24
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	1	1	92	93	94	95	96	97	98	99	100	101	102
103	104	105	106	107	108	109	1	1	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	1	1	128	129	130	131	132	133	134	135	136	137	138
139	140	141	142	143	144	145	1	1	146	147	148	149	150	151	152	153	154	155	156
157	158	159	160	161	162	163	1	1	164	165	166	167	168	169	170	171	172	173	174
175	176	177	178	179	180	181	1	1	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	1	1	200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	1	1	218	219	220	221	222	223	224	225	226	227	228
229	230	231	232	233	234	235	1	1	236	237	238	239	240	241	242	243	244	245	246

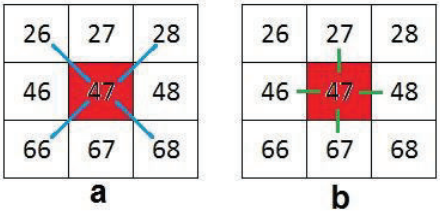
ORIGEN

FINAL

OBSTACULO

Figura 2-30 Matriz de entorno indexado

- Cada nodo posible de ruta tiene 8 vecinos con los cuales puede conectarse, es así que se determina la distancia a cada uno de ellos, de tal forma que cada nodo tendrá 8 distancias posibles por donde ir la ruta.



a) Distancias diagonales b) Distancias horizontales y verticales

Figura 2-31 Distancias entre vecinos

Estas ocho direcciones se las tiene que almacenar igual que el índice del nodo del cual se está calculando, hay que tener en cuenta que las distancias tanto verticales como horizontales son más cortas que en diagonal, razón por la que el momento que se dé la búsqueda estas distancias tendrán más opción en conformar la ruta final que las diagonales. En la Figura 2-31 se puede apreciar que la cantidad de nodos que se encuentran libres y son susceptibles de formar parte de la ruta es elevada, esto da a notar que el cálculo de las distancias va a requerir de una carga computacional bastante considerable.

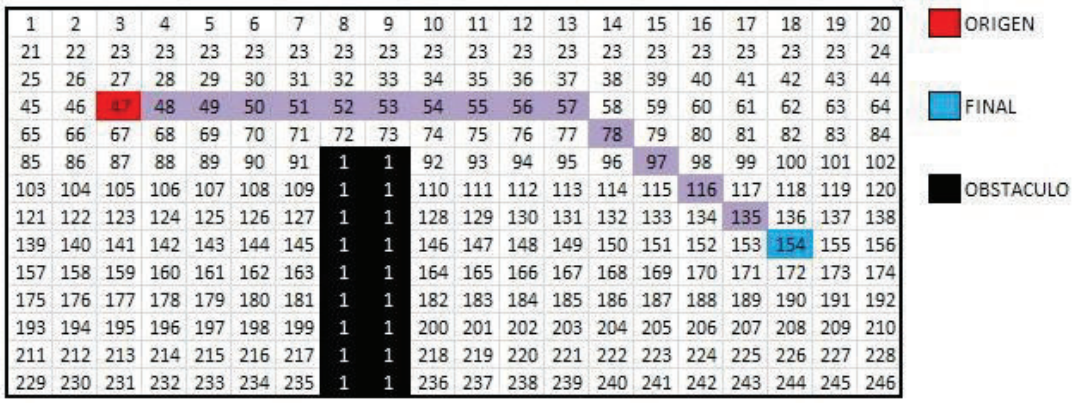


Figura 2-32 Camino encontrado

- Se define un acumulador de las distancias en donde se guarda la distancia, que deberá ser la más corta. De esto se sobreentiende que en casos será mejor

hacer una diagonal que un movimiento vertical y luego horizontal. El orden como se va buscando la ruta lo determinan los índices de la matriz, es decir si empieza en el 8 y el final está en 20, el siguiente a analizar después del 8 será en nueve, caso contario si el inicial está en 30 y el final en 5, el siguiente en análisis será el 29.

2.8. RECONSTRUCCIÓN DE RUTA

Del algoritmo A*, la ruta que se obtiene en las curvas es con movimientos de 45° o múltiplos de 45° Figura 2-32, estos cambios hacen que el movimiento de los agentes no sea fluido, razón por la que será necesario suavizar esos cambios bruscos, para esto se ha recurrido al trazado de las curvas de Bézier. Este es un método que fue presentado por Pierre Bézier en 1962 y tiene la característica de describir curvas de grado n con solo n+1 puntos, es decir para describir una curva cuadrática es necesario únicamente 3 puntos y dentro del trazado siempre se incluyen a los puntos final e inicial. Al mismo tiempo que Bézier el ingeniero Paul de Casteljau trabajaba en un método matemático para la evaluación de las curvas de Bézier basado en los polinomios de Bernstein de donde aparece el algoritmo que lleva su nombre, estos polinomios están definidos mediante la ecuación 2-17.

$$\boldsymbol{B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-1} t^i P_i} \qquad \textbf{(2.17)}$$

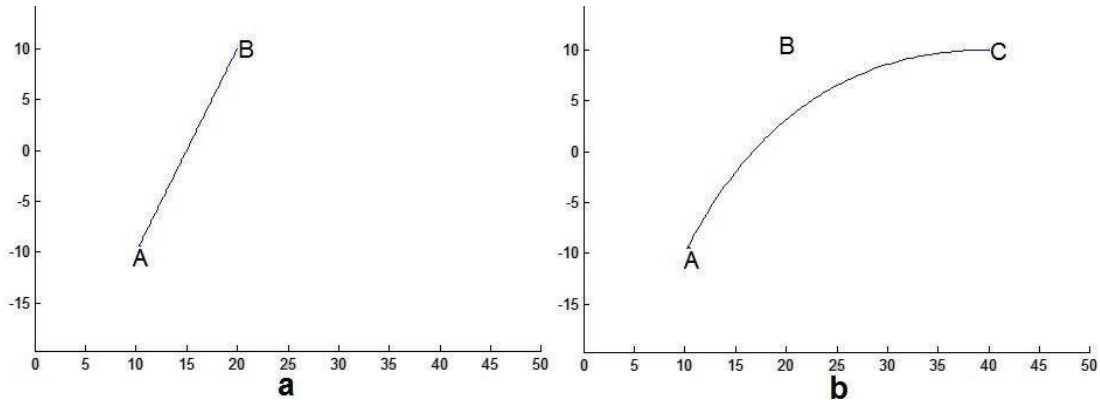
Para este caso en particular se utilizará únicamente curvas de primero y segundo orden, de tal forma que la ecuación para 2 puntos A, B será:

$$\boldsymbol{B(t) = A + (A - B)t, \quad t \in [0, 1]} \qquad \textbf{(2.18)}$$

Ahora para tres puntos A, B, C:

$$\boldsymbol{B(t) = (1-t)^2 A + 2t(1-t)P + t^2 C, \;\; t \in [0, 1]} \qquad \textbf{(2.19)}$$

Los trazos obtenidos con este algoritmo se los pueden observar en la Figura 2-33:



a) Curva 2 puntos b) Curva 3 puntos

Figura 2-33 Trazado de curvas de Bézier

2.9. LÓGICA DIFUSA

La lógica difusa o lógica borrosa es una forma de análisis creado por Lotfi Zadeh, profesor de la Universidad de Berkley y que permite obtener valores de afirmación parciales, es decir puede tomar un valor parcialmente verdadero de 0.6 o parcialmente falso de 0.3, esto difiere de la lógica tradicional en la que los valores son 1 verdadero o 0 falso. Por lo tanto, se pueden tener valores parciales del análisis, por ejemplo, tomando referencia a la humedad de algo, se podría tener que está: muy seco, seco, un poco mojado o mojado [47].

Al tener esta característica de valores parciales del análisis con este tipo de lógica se podrá construir controladores los cuales se comporten de una manera particular y su sintonización sea en base de la experiencia de un operador que está relacionado directamente con el proceso, de esta manera permite pasar el conocimiento humano a un algoritmo de control.

Las ventajas que posee este tipo de control con respecto al control clásico son:

- No requiere de un modelo matemático del proceso.
- No se requiere identificar al sistema.
- No se requiere aproximar al modelo, ya que este no existe.
- No se necesita la linealización del modelo.

Estas características facilitan el diseño del controlador, pero como nada es ideal la complejidad de esta forma de diseño radica en que es necesario conocer las reglas lingüísticas que definen al proceso, estas son reglas de tipo “si….entonces” que definen el comportamiento del sistema [48].

2.9.1. CONCEPTOS BÁSICOS

Para comprender esta forma de análisis es necesario definir algunos conceptos:

2.9.1.1. Universo discurso.

Es la totalidad de elementos que intervienen en el análisis con lógica difusa, este universo es finito y lo define el diseñador del controlador en base de una característica cuantificable.

2.9.1.2. Función de membresía.

Es la medida de la pertenencia de un elemento al conjunto, por ejemplo, una persona pequeña tendrá poca pertenencia en el conjunto alto, en cambio alguien alto tendrá una alta pertenencia dentro del conjunto alto. Las funciones membresía están formadas por varias partes estas son:

- **El núcleo. -** es el conjunto de valores en que tienen la membresía más alta (1) la función toma el valor de uno.

- **Fronteras. -** es el conjunto de elementos que tienen membresía mayor a cero y menor a uno.

- **Soporte. -** es la unión entre el núcleo y las fronteras.

- **Puntos de cruce. -** son los elementos que tienen una membresía igual a 0.5.

- **Ancho de Banda. -** es el valor absoluto de la diferencia entre los dos puntos de cruce.

Estos elementos que forman parte de la función de pertenencia se pueden apreciar en la Figura 2-34.

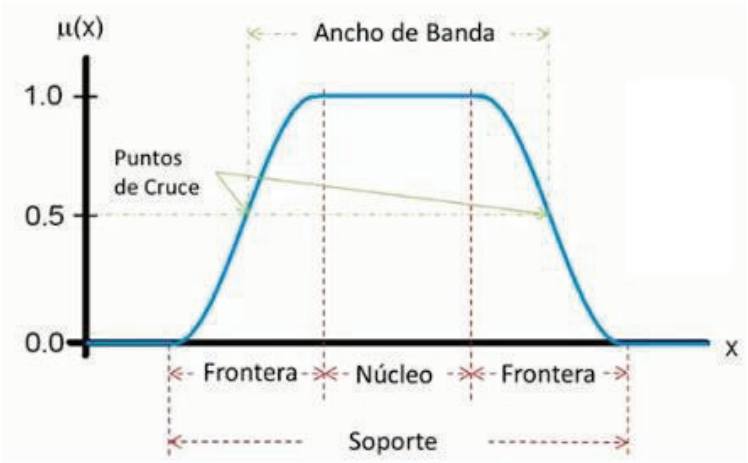


Figura 2-34 Partes de una función de pertenencia

Existen varias formas típicas para las funciones de pertenencias estas son: triangular, trapezoidal, Gausiana, Campana generalizada, sigmoidal.

2.9.2. RAZONAMIENTO DIFUSO

Es la forma como se comporta el análisis para los sistemas difusos, en este caso lo primero es definir una variable lingüística, que son variables las mismas que pueden ser representadas mediante términos lingüísticos para describir el estado de la variable. Para describir una variable lingüística se tiene que definir:

$$(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{T}(\boldsymbol{x}), \boldsymbol{G}, \boldsymbol{M}) \qquad \textbf{(2.20)}$$

Nombre “x”: es característico e identifica a la variable.
Universo de discurso “X”: es el espacio en el cual se va a movilizar la variable.
Valores lingüísticos “T(x)” o términos lingüísticos que acepta la variable.
La regla sintáctica “G”: son los valores que genera los valores lingüísticos en base de funciones de membresía.
La Regla semántica “M”: es la forma como se define la función de membresía de la variable.

Para aclarar estos conceptos, se define un ejemplo referido a la cantidad de luz existente en un invernadero, de tal forma que:

- Nombre: Luz
- Universo: 0 a 1000 lúmenes.
- Valores lingüísticos: {casi nada, poca, normal, semi iluminada, iluminada}

Estos valores tienen que ser definidos en base de las funciones de membresía, por ejemplo:

- M(casi nada)=sigmf(x;-0.05,100)
- M(poca)=gaussF(x;100,300)
- M(normal)=gaussF(x;100,500)
- M(semiiluminada)=gaussF(x;10,700)
- M(casi nada)=sigmf(x;0.05,900)

Las funciones de membresía deben ser asignadas en base de la lógica, por ejemplo, al final del universo discurso no puede ser una gaussiana como en la zona intermedia, ya que esta limitará totalmente el universo discurso sin dar la posibilidad de valores mayores o menores. Esto se aprecia en la Figura 2-35:

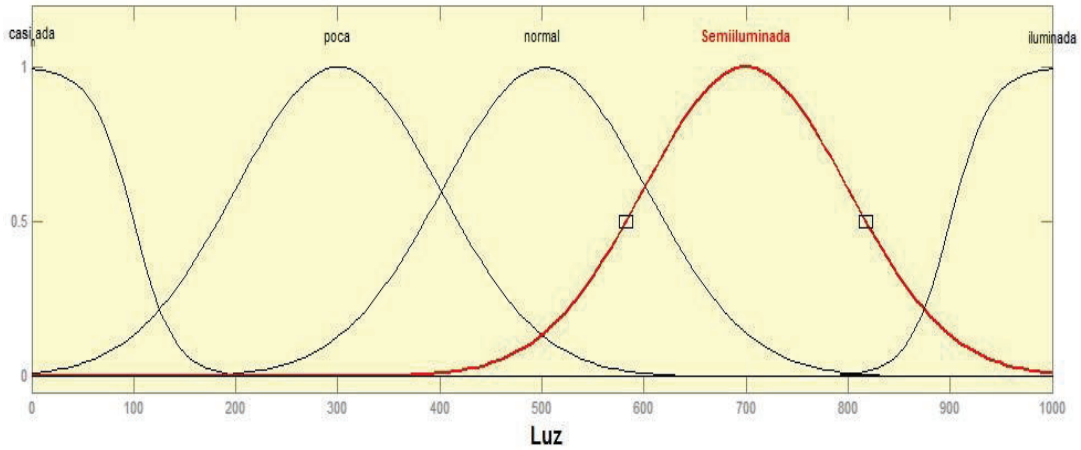


Figura 2-35 Ejemplo variable lingüística

2.9.3. REGLAS SI-ENTONCES

Estas reglas son la esencia de los controladores difusos ya que de ellas dependerá el comportamiento del control en base de procesamiento de la información de la entrada para obtener resultados en la salida.

Se tendrá una definición de la siguiente manera:

Si x es A, entonces y es B

De forma más clara:

Si la luz es poca, entonces la temperatura es baja.

Si la presión es alta, entonces el color del medidor es rojo.

Si el color es rojo, entonces la mora está madura.

Estas reglas si-entonces tienen que ser lo más claras posibles para que al momento del procesamiento computacional sea lo más ágil posible para reducir la carga computacional del sistema.

2.10. CONTROL DIFUSO

Recordando un poco sistemas de control, el sistema más eficaz es el aquel en el cual se tiene la realimentación de la variable a controlar, es decir en donde se obtiene una realimentación de esta, para lo cual se trabaja con el diagrama mostrado en la Figura 2-36, en donde la variable a ser controlada es realimentada mediante un sensor al sistema. Se compara con el set point, y se obtiene el error, el mismo que ingresa al controlador el cual determina una acción de control que es aplicada a la planta para modificar la variable a ser controlada, estas acciones se realizan cíclicamente hasta que el error sea cero, lo que significa que la planta se encuentra controlada.

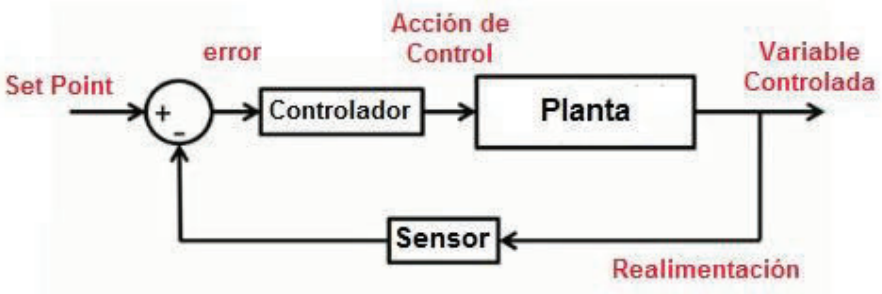


Figura 2-36 Control realimentado

Para el caso del control difuso el esquema de la Figura 2-36 se mantiene, con la particularidad que el boque controlador ahora es el sistema difuso diseñado.

Este tipo de control también es conocido como control no convencional y tiene otras formas de control como el basado en redes neuronales artificiales, algoritmos genéticos, sistemas expertos entre otros. El controlador difuso visto como diagrama de bloques es el mostrado en la Figura 2-37.

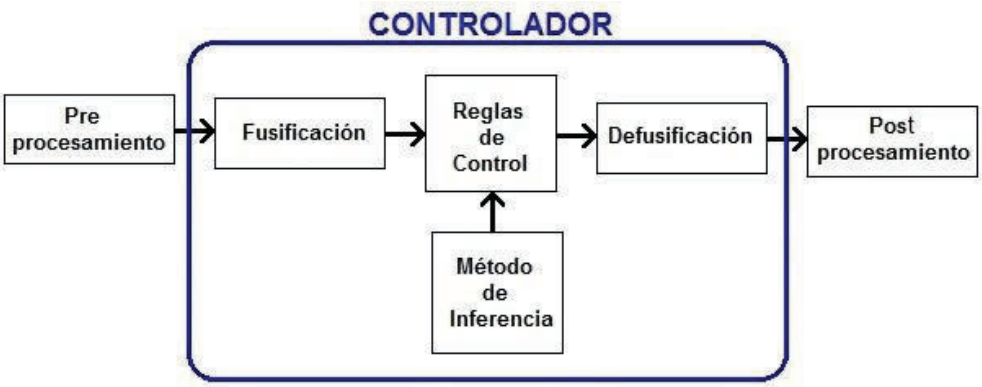


Figura 2-37 Controlador difuso

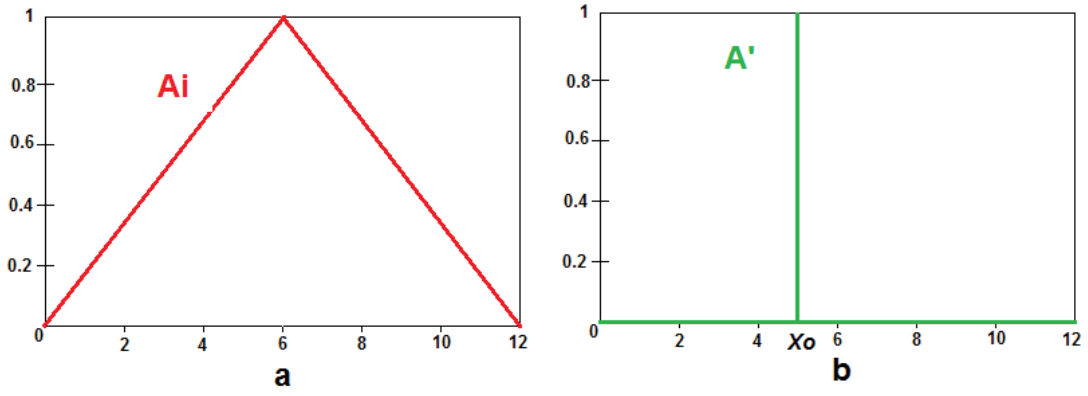
Analizando cada uno de estos bloques se tiene:

Pre procesamiento. - acondiciona las señales que ingresan al sistema de procesamiento, como por ejemplo un filtrado, regulación de la ganancia o muestreo de la señal.

Fusificación. - convierte las magnitudes de las señales en cantidades difusas, obteniendo el valor de pertenencia en cada uno de los valores lingüísticos, es decir evalúa la señal de entrada y le asigna un valor de pertenencia.

Para esto, primero se tiene que identificar la función de membresía y la función de entrada al controlador, para el caso la función de membresía esta será definida por el usuario, en cualquiera de los tipos triangular, trapezoidal, etc. La función de entrada en cambio podría ser el error, que tiene la particularidad de ser una función

de tipo singleton, que básicamente es una función que posee un solo valor de 1, las funciones se las puede identificar en la figura 2-38 [49].



a) Función de membresía b) Función de entrada

Figura 2-38 Funciones que intervienen en la fusificación.

Con las funciones se realiza la intersección entre las funciones gráficamente y que equivale a la minimización y maximización entre las dos funciones mediante la ecuación 2-21:

$$\vee_x\left[\boldsymbol{u}_{A'}(\boldsymbol{x})\wedge\boldsymbol{u}_{A_i}(\boldsymbol{x})\right] = \boldsymbol{u}_{A_i}(\boldsymbol{x}_o) \qquad (2.21)$$

En donde, la ecuación expresa que se debe obtener el máximo de los mínimos entre $u_{A'}$ y u_{Ai}, que en realidad es la evaluación de u_{Ai} para el valor de x_o. Gráficamente se obtiene la figura 2-39. En donde al valor que tiene U_{Ai}, se lo conoce como el valor de fusificación.

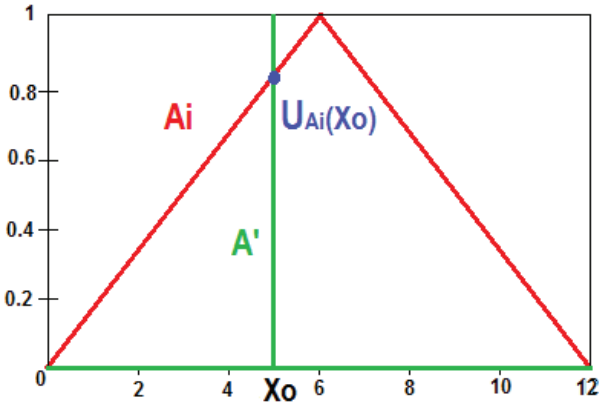


Figura 2-39 Resultado de la fusificación.

Reglas de control. - es el conjunto de reglas si-entonces que definen el comportamiento del controlador en base de la experiencia del diseñador del controlador. Estas reglas establecen las relaciones existentes entre las entradas y las salidas, por ejemplo:

Si x es A1, entonces y es B1.

En la expresión se identifica que x es la entrada, y es la salida, A1 y B1, son los valores correspondientes.

Método de inferencia. - es el algoritmo que seguirá el sistema computacional para inferir la conclusión a partir de las premisas para él establecidas mediante las reglas si-entonces, en base de los datos obtenidos a partir del proceso de fusificación. De los métodos de inferencia define los tipos de controladores, por ejemplo: Mamdami, Sugeno, Tsukamoto, entre otros, de estos el primero es el método general y los demás se constituyen en simplificaciones [50] [49].

En base de las reglas si entonces se puede establecer la relación del sistema de inferencia, en donde:

Tabla 2-1 Planteamientos de inferencia

Planteamiento Lógico	Planteamiento Matemático
x es A’	
si x es A_1, entonces y es B_1.	$u_{B'_1}(y) = u_{A_1}(x_o) \wedge u_{B_1}(y)$
si x es A_2, entonces y es B_2.	$u_{B'_2}(y) = u_{A_2}(x_o) \wedge u_{B_2}(y)$
y es B’	$u_{B'}(y) = u_{B'_1}(y) \vee u_{B'_2}(y)$

De estas expresiones se tiene que A’ es el ingreso al controlador, B’ será la salida del controlador y la cantidad de reglas si-entonces pueden ser determinadas por el diseñador, en la Tabla 2-1 se establece para dos reglas.

En las reglas se tiene que ***“si x es A_1”***, esto se constituye en el proceso de fusificación, ***“entonces y es B_1”*** corresponde a una minimización con el consecuente del sistema difuso, por último ***“y es B’ ”***, corresponde a la

maximización entre los valores obtenidos de cada una de las reglas si-entonces para obtener un área que correspondería a los posibles valores de salida del controlador. Visto gráficamente se tiene la Figura 2-40.

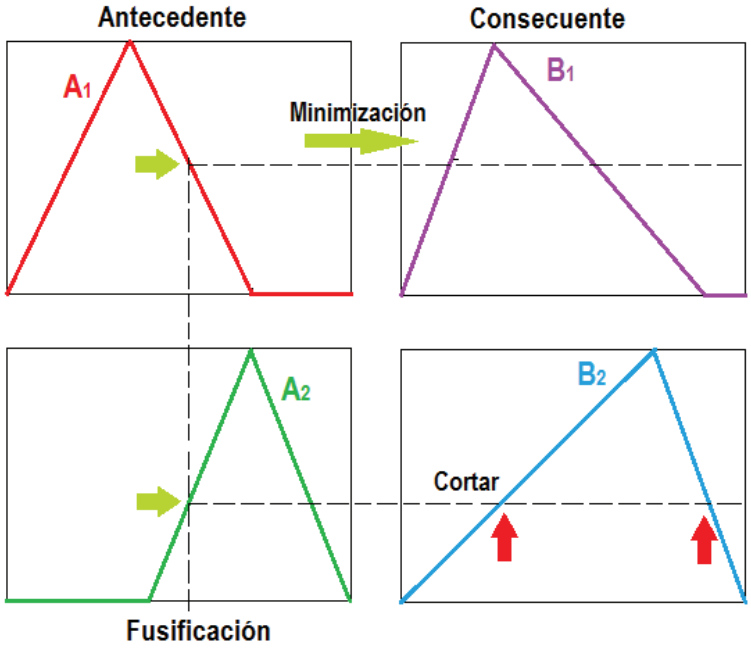
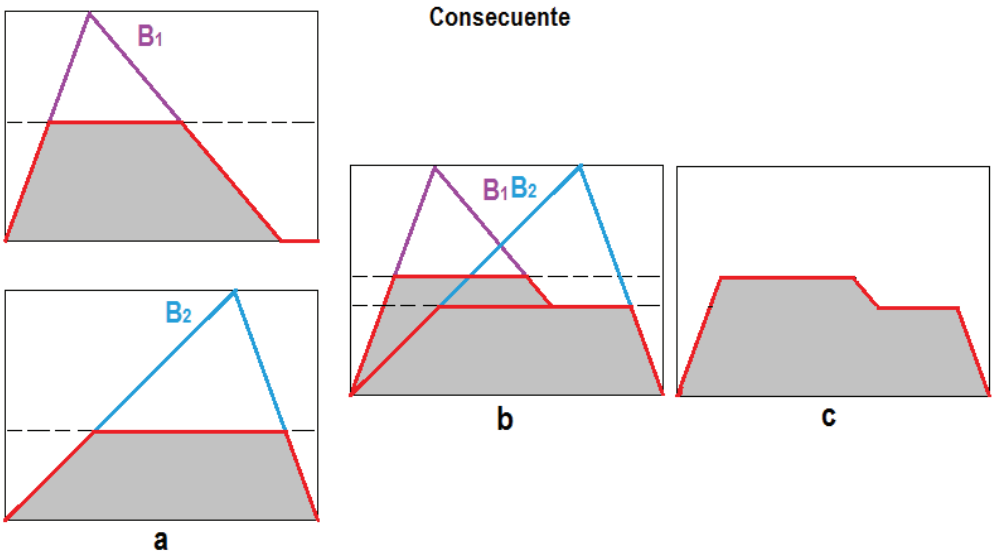


Figura 2-40 Primera parte de inferencia

Con los puntos obtenidos, se determinan las áreas y se las une, obteniéndose la Figura 2-41.



a) Resultado de minimización b) Unión de respuestas c) Repuesta final de inferencia

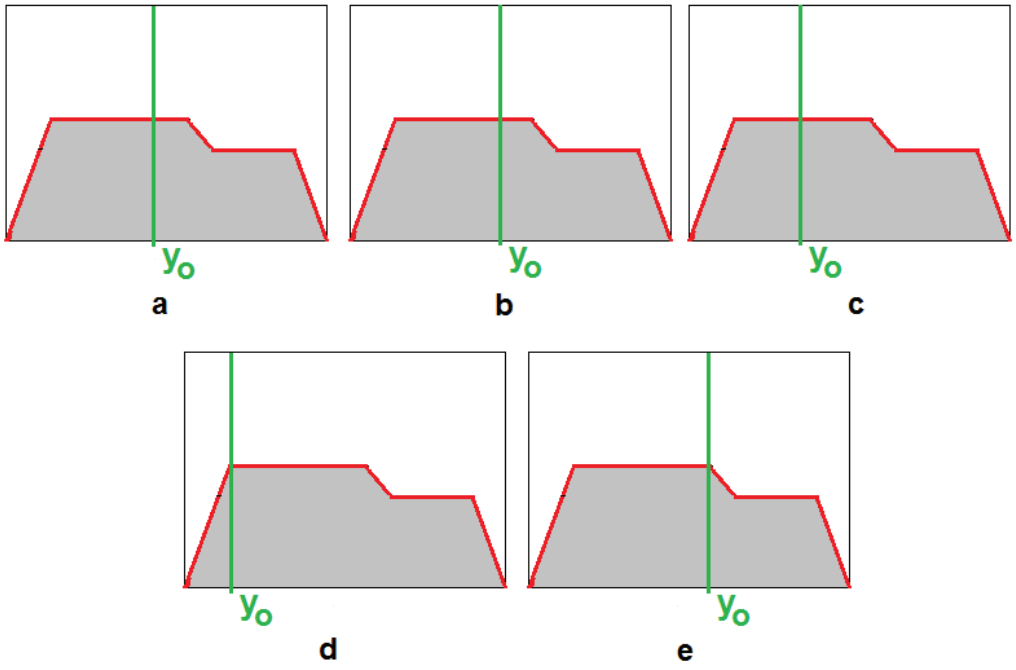
Figura 2-41 Segunda parte de inferencia

Defusificación. - convierte los valores del controlador en una magnitud certera, un único número, para generar la señal de control. De los métodos de inferencia se ha obtenido un conjunto de valores para esto se puede utilizar métodos como: centroide, bisectriz, máximo central (MOM), máximo más pequeño (SOM), máximo más grande (LOM) [50].

Cada uno de estos métodos se comportan de la siguiente manera:

- Centroide. - ubica el centro de masa del área bajo la curva.
- Bisectriz. - divide al área en dos áreas iguales.
- Máximo central (MOM). - ubica el centro de los máximos obtenidos.
- Máximo más pequeño (SOM). - ubica el máximo más pequeño.
- Máximo más grande (LOM). - ubica el máximo más grande.

Estos métodos se pueden apreciar en la Figura 2-42.



a) Centroide b) Bisectriz c) Máximo central (MOM) d) Máximo más pequeño (SOM) e) Máximo más grande (LOM).

Figura 2-42 Métodos de defusificación.

Post Procesamiento. - genera la señal de control a partir de la respuesta del controlador para ser aplicada al actuador que se relaciona directamente con la planta, como por ejemplo un filtrado o una conversión digital análoga [48].

Para diseñar un sistema de control difuso se procede de la siguiente manera:

- Definir las variables lingüísticas tanto de entrada como de salida y los términos lingüísticos asociados a cada una de ellas.
- Establecer las reglas difusas que en base de su evaluación determinen el comportamiento del controlador.

CAPÍTULO 3

3.1. VISIÓN GENERAL DEL SISTEMA

El sistema está conformado de varios componentes, los mismos que han sido determinados en base de las características de operación que se requiere, de forma general el sistema está estructurado como se ve en la Figura 3-1.

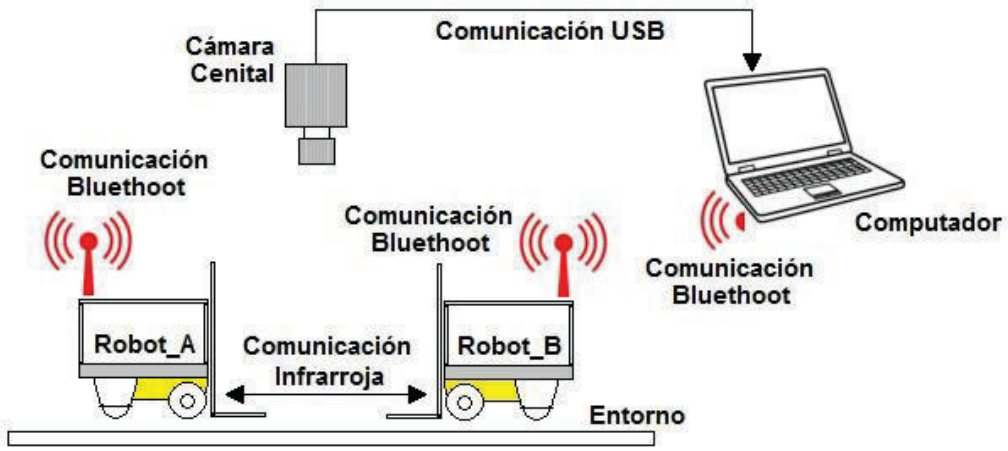


Figura 3-1 Estructura general del sistema

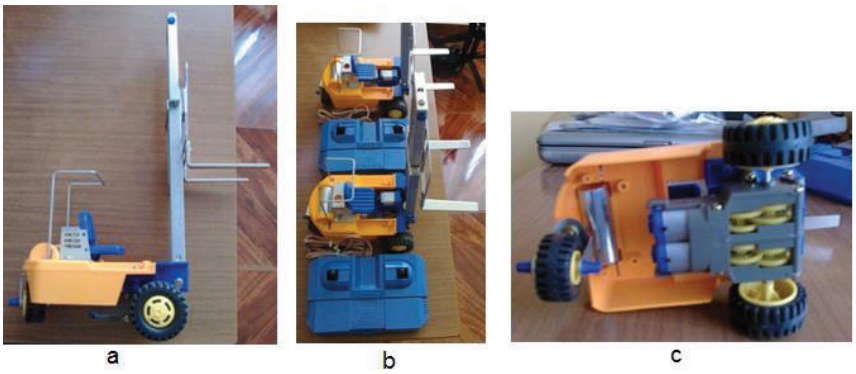
La operación del sistema está dada de la siguiente forma:

El computador es el elemento maestro y se encarga de toda la operación del sistema, en una primera instancia ordena a la cámara la captura de la imagen mediante un protocolo de comunicación USB. Posteriormente procesa la información y determina las acciones de control para ser enviadas a cada uno de los agentes: Robot_A y Robot_B, esto lo hace vía comunicación inalámbrica bluetooth. Entre los dos agentes se da una comunicación de tipo infrarroja que sirve de apoyo para la sincronización de movimientos.

3.2. DISEÑO DEL HARDWARE

El hardware del sistema está constituido por dos agentes robóticos, los mismos se han obtenido de un par de juguetes de tipo montacargas a los cuales se les ha provisto de la instrumentación, tarjeta de desarrollo y actuadores para que se conviertan en dos pequeños robots para funcionamiento cooperativo. Esto se

puede apreciar en la Figura 3-2, en donde se aprecia el estado inicial los agentes que únicamente son controlados de manera manual, Figura 3-2 b, razón por la que el sistema no posee ningún tipo de instrumentación. El sistema de tracción para locomoción está formado por una trasmisión de tipo diferencial marca Tamiya y para la elevación de la horquilla se dispone de un tercer motor, acoplado a un sistema de tornillo sinfín y una cadena para la elevación.



a) Montacargas b) Controles original c) Sistema de transmisión

Figura 3-2 Estado inicial de los agentes como juguetes

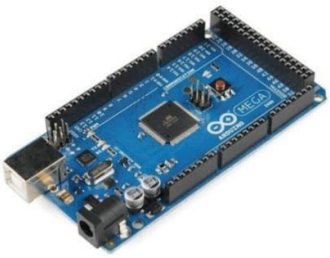
3.2.1. TARJETA DE DESARROLLO

Lo primero es proporcionarle al robot un sistema procesado o tarjeta de desarrollo, para esto se decidió utilizar una placa comercial por el ahorro de tiempo que implica su utilización, en específico se utilizó una placa Arduino Mega 2560, Figura 3-3, la misma que posee las características mostradas en la Tabla 3-1:

Tabla 3-1 Características de Arduino Mega 2560

Microcontrolador	Atmega2560
Voltaje de Entrada	7 – 12 V_{DC}
Voltaje de operación	5 V_{DC}
Pines Digitales	54, de los cuales 15 PWM
Pines Analógicos	16
Corriente entrada - salida	20 mA
SRAM	8 KB
Memoria Flash	256 KB, 8KB gestor de arranque
EEPROM	4 KB
Reloj	16 MHz.

Se podría pensar que este tipo de tarjeta de desarrollo está sobre dimensionada, pero se tiene la concepción de que estos agentes sirvan en un futuro cercano para realizar más estudios relacionados con la robótica cooperativa y visión artificial. Para más detalle referirse al anexo 2.

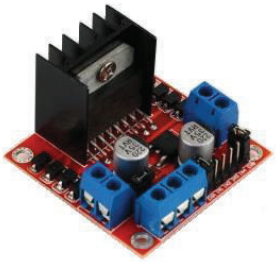


Placa de desarrollo Arduino. [51]

Figura 3-3 Placa Arduino Mega 2560

3.2.2. MOTORES DE TRACCIÓN

Los montacargas presentan una estructura de tipo diferencial, que es una de las más sencillas para el control. Esta estructura está formada por dos ruedas que están situadas diametralmente opuestas a un eje perpendicular a la dirección de movimiento del robot, por esta razón es necesaria la utilización de dos motores para proporcionar movimiento a cada una de las llantas. Los movimientos se generan dando diferentes velocidades y sentidos de giro a los motores, por ejemplo: para un giro a la derecha se mueve el motor izquierdo a mayor velocidad que el derecho, esto genera un movimiento con una curvatura amplia, pero si se quiere reducir la curvatura se tiene que bajar la velocidad del motor derecho o hacerla más cerrada incluso se podría invertir el sentido de giro. De lo expuesto se entiende que es necesaria la variación de la velocidad y sentido de giro para lo cual se utilizará un driver comercial L298N el mostrado en la Figura 3-4.



Driver Motores Corriente Continua. [52]

Figura 3-4 Driver dual de motores

Este driver tiene las características mostradas en la Tabla 3-2:

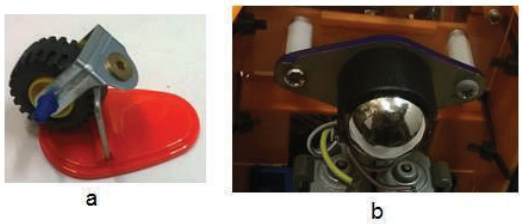
Tabla 3-2 Características de driver 298N.

Circuito Integrado básico	L298N
Voltaje de alimentación	5 – 46 V_{DC}
Corriente Máxima	3A
Regulador adicional	LM7805
Tensión Lógica	5V_{DC}
Corriente lógica	0 – 36 mA
Potencia Máxima	25 W
Motores a conectar	2

El driver de la Figura 3-4, posee dentro de su estructura diodos de clamp para proteger a los semiconductores de los puentes H. Además de forma experimental se determinó que las corrientes de los motores de tracción, obteniendo que tienen un consumo de corriente nominal de 0.75A, de tal forma que el driver utilizado tiene la capacidad de corriente suficiente para el manejo de los motores. Para más detalle de este driver debe referirse al anexo 4.

3.2.3. RUEDA LOCA

Para mantener la horizontalidad del robot es obligatorio proporcionarle un tercer punto de apoyo para lo cual es necesaria la utilización de una rueda adicional; originalmente se tenía una rueda tipo castor la cual fue sustituida por una rueda de tipo omnidireccional, debido a que durante el movimiento generaba trabas para el libre desplazamiento del móvil por a la posición en la que quedaba del movimiento previo en casos quedaba totalmente paralela al eje de las llantas provocando una fricción realmente alta. El detalle de las llantas se tiene en la Figura 3-5.



a) Rueda original castor b) Rueda nueva omnidireccional

Figura 3-5 Rueda loca de los agentes.

3.2.4. MOTOR DE HORQUILLA

Los agentes para elevar el objeto a transportar disponen de una horquilla cada uno, las mismas que mediante un sistema de cadena y tornillo sinfín pueden subir las horquillas propulsado por medio de un motor de 12 VDC que consume una corriente de 0.75A que es del mismo tipo que los de tracción, por consiguiente, se lo podría controlar mediante un driver L298N, pero este driver es para dos motores, por lo tanto, quedaría subutilizado. Por esta razón es que para el control del motor de la horquilla se optó por la utilización de un CI TA7257P, el que posee en su interior un puente H completo para el manejo de un motor y posee las siguientes mostradas en la Tabla 3-3, para más detalle referirse al anexo 3:

Tabla 3-3 características CI TA7257P.

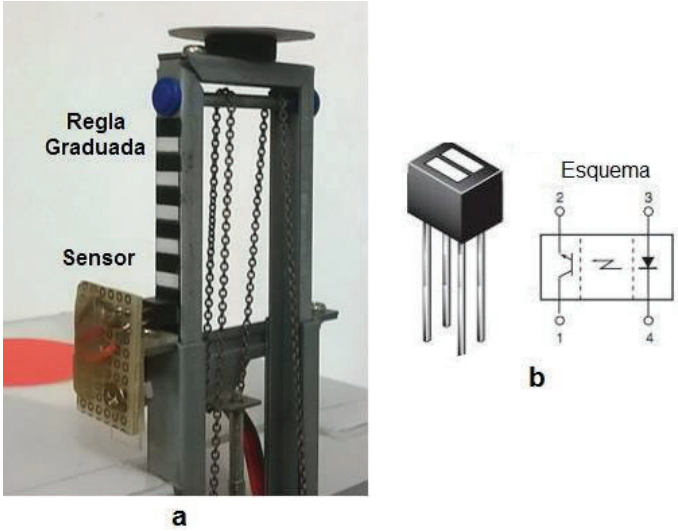
Voltaje de alimentación	6 – 18 V_{DC}
Corriente	1.5A y pico 4.5A.
Diodos	Incluidos
Funciones	CW, CCW, STOP, Brake
Pines de control	2

Circuito Integrado Puente H. [53].

Figura 3-6 Driver TA7257P.

Para obtener una elevación controlada de la horquilla es necesario cerrar un lazo de control, por esta razón es impresindible conocer la posición de la horquilla para esto se ha incluido una regla graduada en base de marcas blancas y negras, como se observa en la Figura 3-7. Para la lectura de la posición se ha colocado un sensor QRD1114 que es un sensor opto reflectivo que en su encapsulado tiene incluido un diodo emisor infrarrojo y un receptor óptico en base de transistor, el mismo que

tiene que estar a una distancia de 0.5 a 1 cm de la superficie reflectiva para una adecuada operación Figura 3-7.



a)Sistema Regla sensor b) Sensor QRD1114 [54]

Figura 3-7 Sistema regla sensor.

Para la utilización del sensor QRD1114 es necesario garantizar los valores de la recepción del foto transistor, por consiguiente se tiene que incluir un circuito integrado con Smith trigger para así asegurar los valores lógicos que ingresan a la tarjeta de desarrollo, para esto se utilizó un CI7414, el mismo que es un inversor lógico con ventana de histéresis, con la configuración mostrada en el circuito de la Figura 3-8.

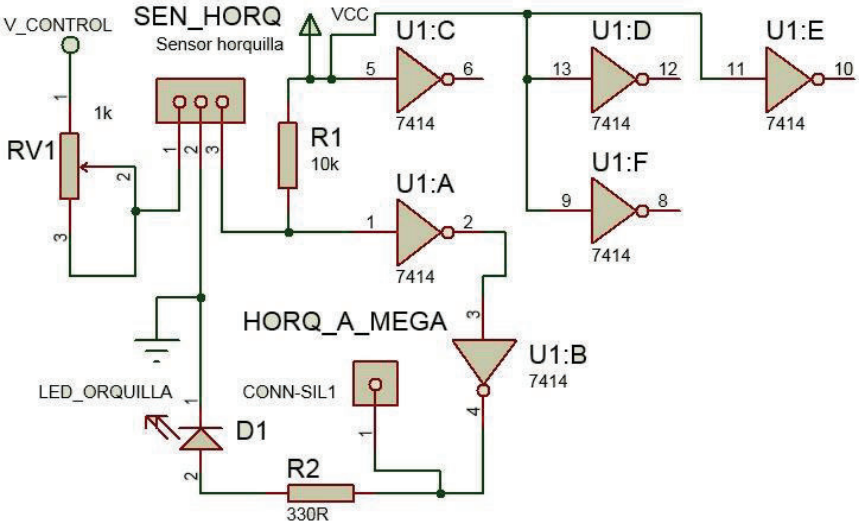


Figura 3-8 Acondicionador de sensor.

3.2.5. COMUNICACIÓN INFRARROJA

Una forma de comunicación existente entre los agentes es mediante un enlace con luz infrarroja, para esto se creó un circuito oscilador con compuertas NAND bajo la configuración de oscilador para que genere la portadora de la información y esta señal mediante una operación NAND se modula con la información y con la señal resultante se la aplica al diodo emisor de IR. Figura 3-9.

Para la recepción se ha utilizado un circuito integrado IRM8601, para la operación de este solo se necesitan unos cuantos elementos más, como los mostrados en la Figura 3-9.

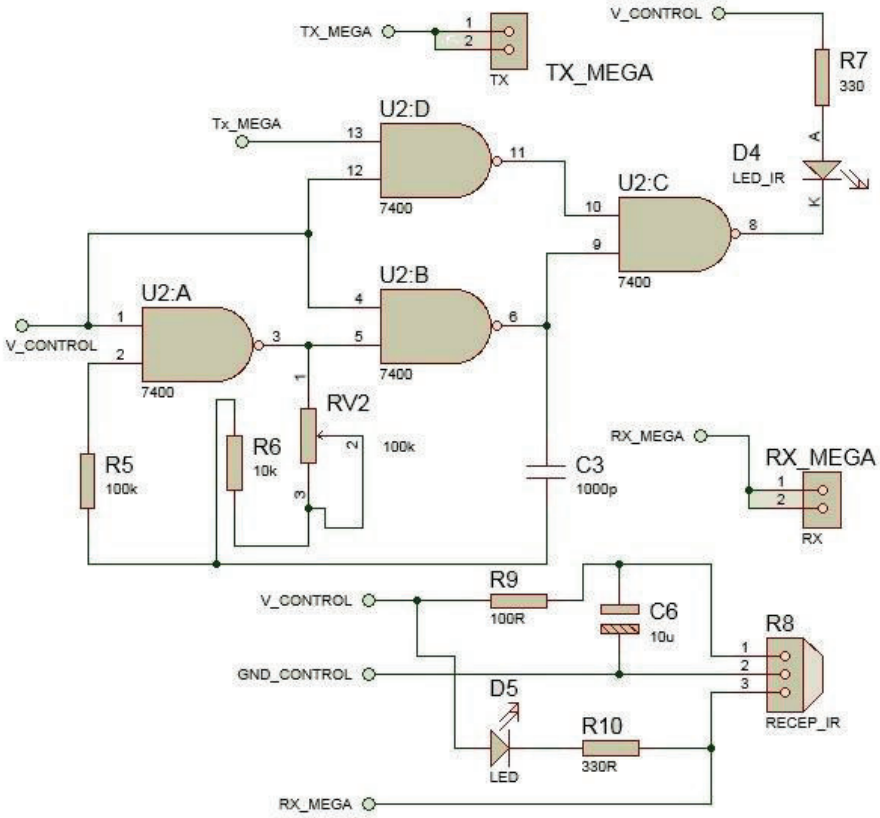


Figura 3-9 Sistema de comunicación infrarroja.

3.2.6. FUENTE DE ALIMENTACIÓN

Para el funcionamiento de los agentes es necesaria la provisión de corriente continua razón por la que se han colocado baterías, en específico dos, una para la

parte de control y otra para la parte de potencia, en donde cada circuito de alimentación debe contar con un sistema de mando o activación, Figura 3-10.

En base de las dimensiones de los agentes se ha buscado en el mercado baterías que tengan una buena provisión de corriente, recargables, una tasa alta de descarga y con pequeñas dimensiones, es así que se ha optado por la utilización de baterías lipo (litio – polímero) de dos celdas, por lo tanto, el voltaje entregado será de 7.2 voltios y una corriente de 1000 mA.

En función de ese voltaje, deberá ser implementado un circuito regulador para el sistema de control que trabaja con 5 voltios de corriente continua, este se aprecia en la Figura 3-10.

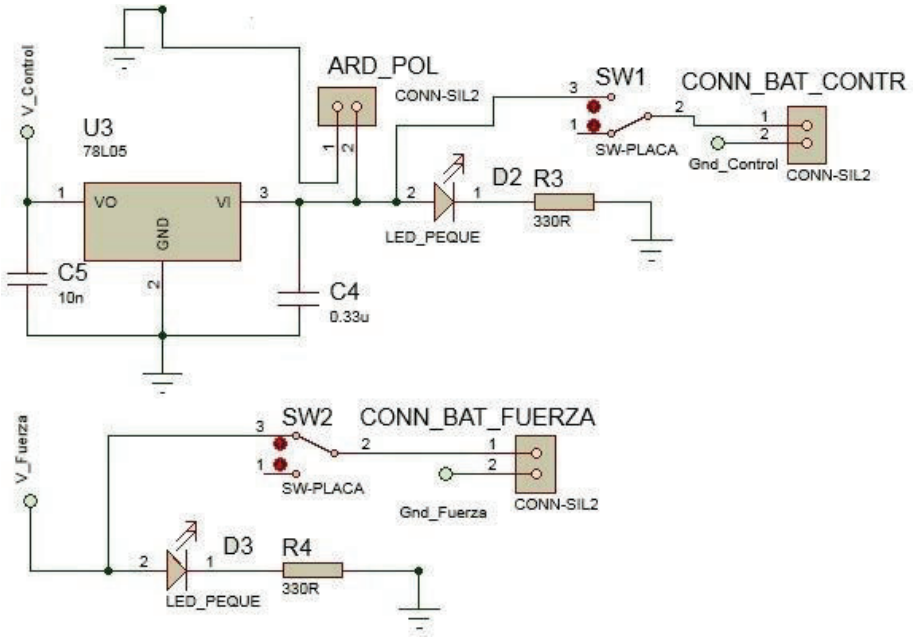


Figura 3-10 Circuitos de alimentación de fuente.

La placa que contiene los elementos electrónicos se le han adicionado algunas prestaciones extras, como por ejemplo un buzzer para que sirva como señal auditiva en el proceso de desarrollo de los algoritmos, además se ha dejado varios puntos para tomas de voltaje, cada una de ellas son sus respectivos conectores, las mismas que servirán en un futuro para obtener los voltajes de polarización para placas adicionales, esto se ve en la Figura 3-11. Todos estos circuitos electrónicos

han sido colocados en un solo circuito impreso, el cual ha sido diseñado de tal forma que esta placa sirva como parte del chasis y que ayude como elemento para la sujeción de los otros módulos como por ejemplo el driver de motores de tracción y la placa Arduino.

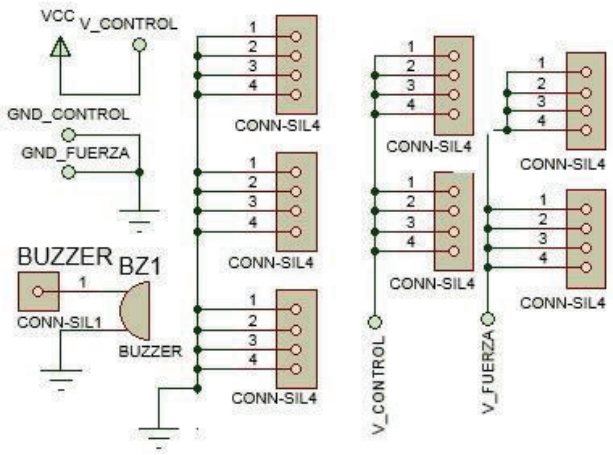


Figura 3-11 Prestaciones adicionales de la placa.

Para esto se tomaron las dimensiones de los agentes obteniendo una base con la forma mostrada en la Figura 3-12.

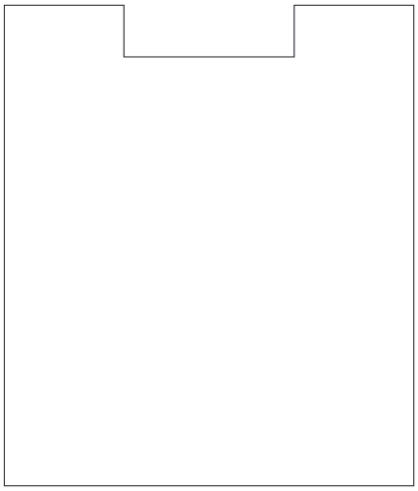
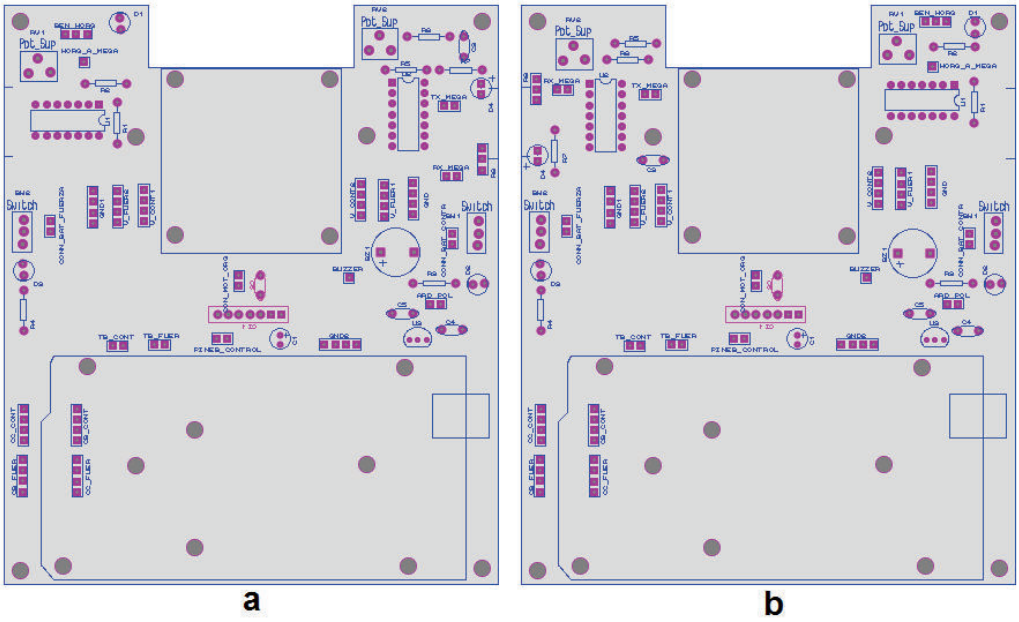


Figura 3-12 Forma para el circuito impreso.

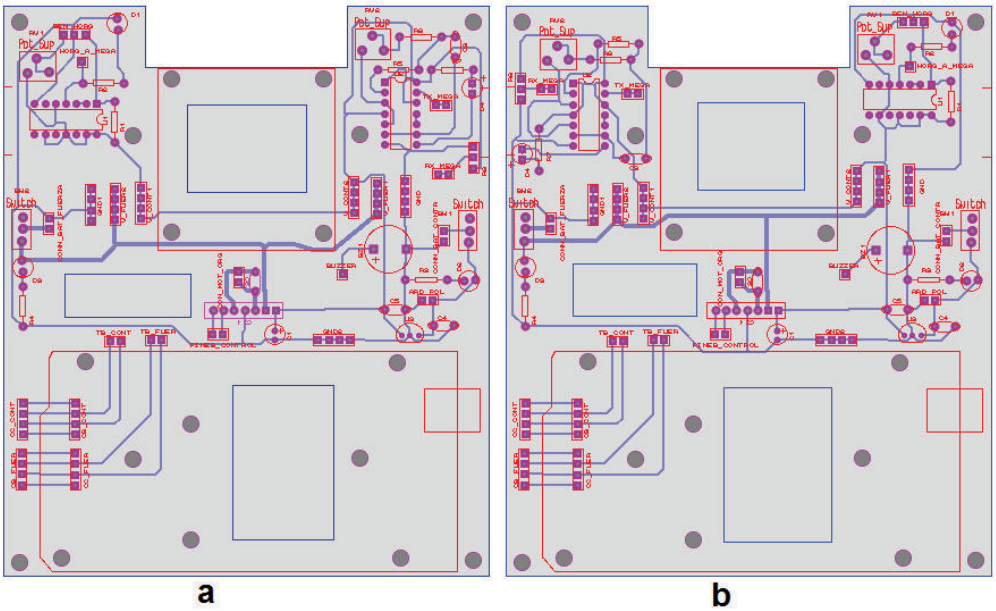
Se distribuyeron los elementos en las bases diseñadas, obteniendo las distribuciones mostradas en la Figura 3-13.



a) Robot_A b) Robot_B

Figura 3-13 Distribución elementos en placa de robots.

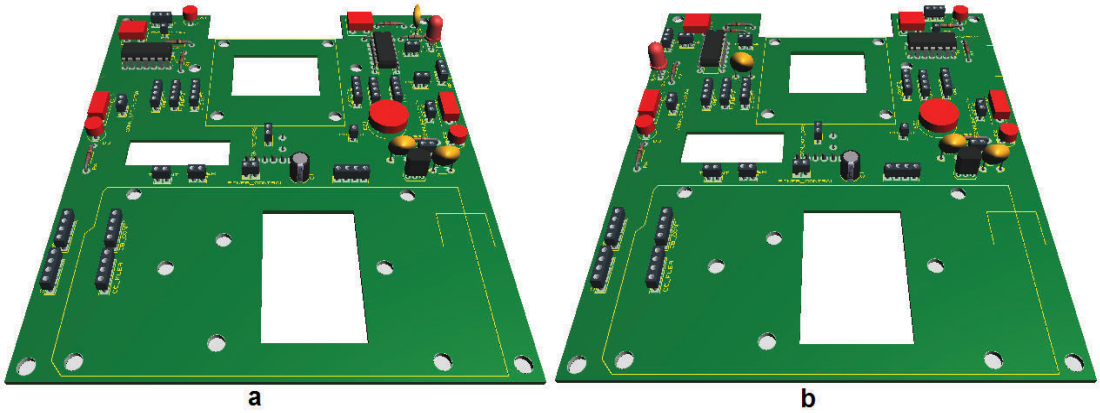
Re realiza el ruteado de la placa, obteniendo lo mostrado en la Figura 3-14.



a) Robot_A b) Robot_B

Figura 3-14 Placas con el ruteo de los circuitos.

Para tener una idea como quedaron los circuitos distribuidos en la placa final de los robots, se tiene en la Figura 3-15.



a) Robot_A b) Robot_B

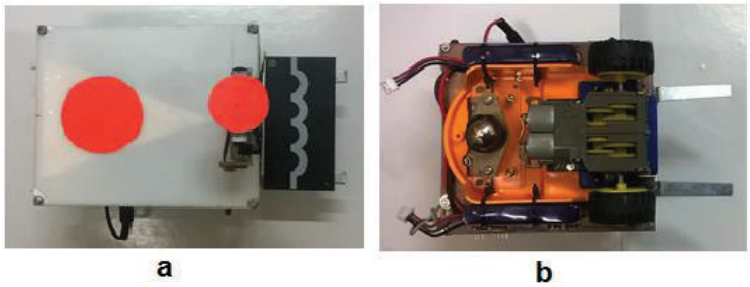
Figura 3-15 Circuitos en tres dimensiones.

La última modificación que se ha dado al hardware original de los montacargas de juguete fue proveerles de marcas, debido a que los algoritmos de visión artificial basan su operación en buscar las marcas y con ellas determinar características de su posición para el guiado de su movimiento. La primera es un círculo de color sobre el sistema de elevación de la horquilla, el mismo que coincide con el centro del eje de las llantas, la segunda marca es otro círculo que se ubica en una tapa superior colocada de forma paralela a la placa diseñada con los circuitos integrados, para identificar estas marcas se lo puede hacer en la Figura 3-16.

También fue necesario cargarles de un poco más de peso para mejorar el coeficiente de fricción de las llantas, para lo cual se colocaron placas de metal adicional en las horquillas.

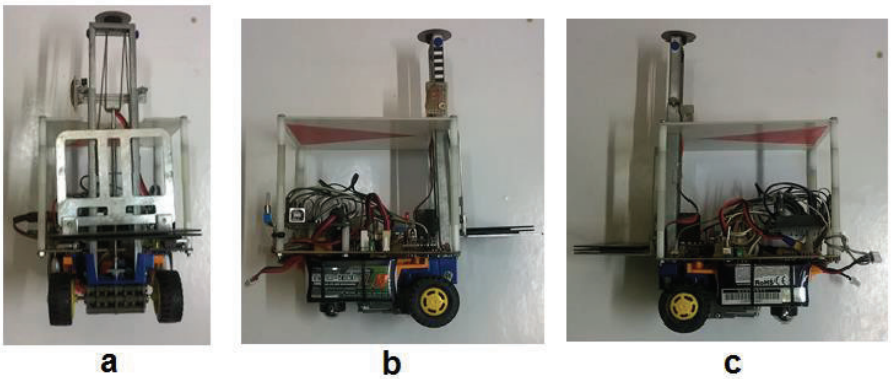
Figura 3-16 Marcas y pesos en los agentes.

El aspecto final de los robots móviles son los mostrados en las Figuras 3-17, 3-18 en donde se pueden apreciar varios de los detalles citados en esta sección.



a) Vista superior b) Vista Inferior

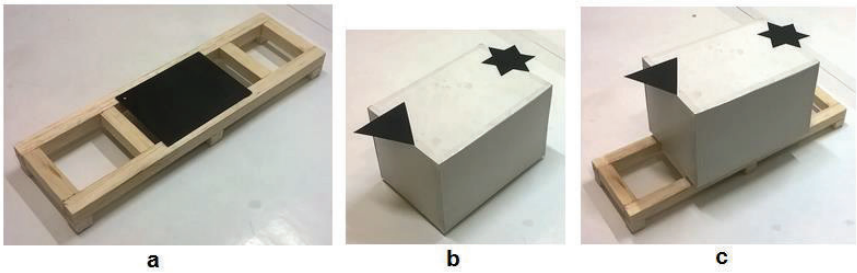
Figura 3-17 Vistas de los agentes automatizados.



a) Vista frontal b) Vista lateral izquierda c) Vista lateral derecha

Figura 3-18 Vistas laterales de los agentes automatizados.

Como el objetivo es el transporte de un objeto desde un punto a otro, se ha decidido que este sea un prisma cuadrangular el cual es identificado dentro del entorno mediante marcas, pero como los agentes están provistos de horquillas, estas tienen que ingresar debajo del objeto, se ha creado un pequeño pallete sobre el que está montado el objeto, esto se aprecia en la Figura 3-19.



a)Pallete b) Objeto c) Conjunto pallete-objeto

Figura 3-19 Objeto y Pallete.

3.3. DISEÑO DEL SOFTWARE

El sistema para su operación necesita de un algoritmo que le provea de la inteligencia suficiente para analizar las imágenes adquiridas y determinar las acciones de control, así como también con la capacidad suficiente para interpretar los resultados del procesamiento de análisis para ejecutar las acciones de control.

El sistema está formado por dos partes, la estación maestra la cual analiza y determina las acciones de control, para esta acción se ha decidido la utilización del software MATLAB R2013a, el cual tiene su mayor potencial en el manejo de matrices que para este caso particular las imágenes en formato digital son matrices. El otro componente lo constituyen los agentes robóticos, los cuales deberán interpretar las acciones de control obtenidas por el maestro para esta tarea se ha determinado la utilización de Arduino Soft, este es el software nativo de la tarjeta controladora Arduino Mega 2560.

3.4. DISEÑO DE SOFTWARE PARA EL MAESTRO.

El sistema básicamente funciona de manera general en base del esquema mostrado en la Figura 3-20. En donde cada bloque representa un programa para una tarea específica del sistema total.

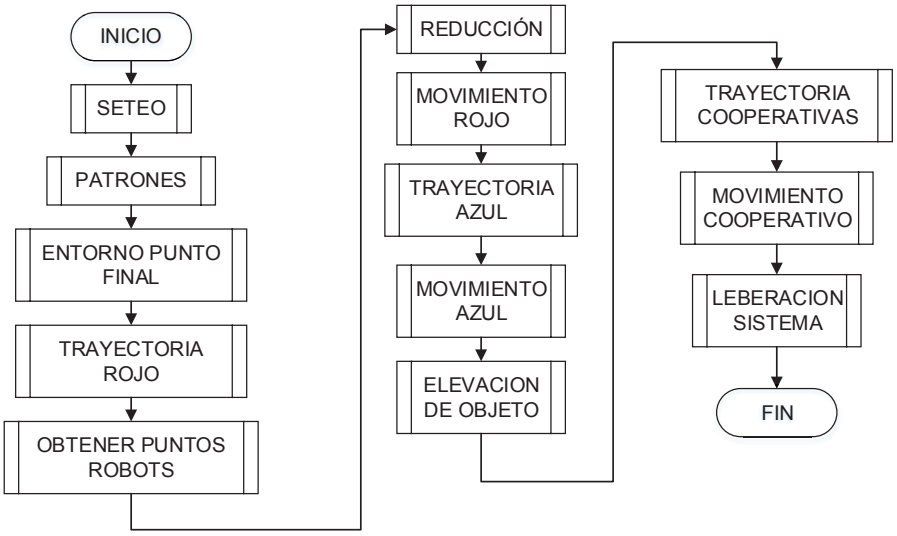


Figura 3-20 Esquema general del sistema.

Para un mejor detalle se analiza cada uno de los bloques de forma individual, especificando el resultado de la ejecución.

3.4.1. BLOQUE SETEO

En este bloque básicamente se realizan las siguientes acciones: se limpian todas las variables del sistema, se crea el canal de video para la captura de las imágenes, se visualiza el video capturado para encuadrar el entorno. Se crea los canales de comunicación bluetooth para cada uno de los agentes, se abre los canales y se establece y estabiliza la comunicación entre el maestro y los dos esclavos. A nivel de diagrama lógico se tiene el mostrado en la Figura 3-21.

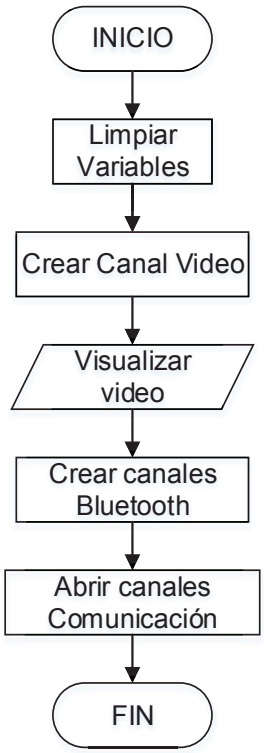


Figura 3-21 Bloque Seteo.

3.4.2. BLOQUE PATRONES

Para la ejecución de este bloque es necesario colocar en el ambiente una plantilla patrón, de esta se calculan dos datos: la separación entre los centros de los dos cuadrados y el lado de los mismos. Estas dos variables ayudan en cálculos posteriores. Se utiliza patrones para la calibración debido a que se desea que la altura a la cual se encuentra la cámara cenital no influya en el desempeño del sistema. Como las dimensiones que tiene la plantilla patrón son conocidas, es base de esos datos se puede escalar los gráficos obtenidos para tener las respuestas en unidades conocidas (cm).

El proceso funciona de la siguiente manera: se captura la imagen, a esta se le binariza por umbral. A la imagen binaria se la etiqueta y se obtiene sus propiedades, en específico sus centroides, con estos se calcula la distancia entre los dos, que para futuros cálculos será la distancia entre los robots, para esto se utiliza la ecuación 3-1, pero este valor tiene que ser redondeado para obtener valores enteros por que se trabajará en pixeles y estos no tienen valores fraccionarios.

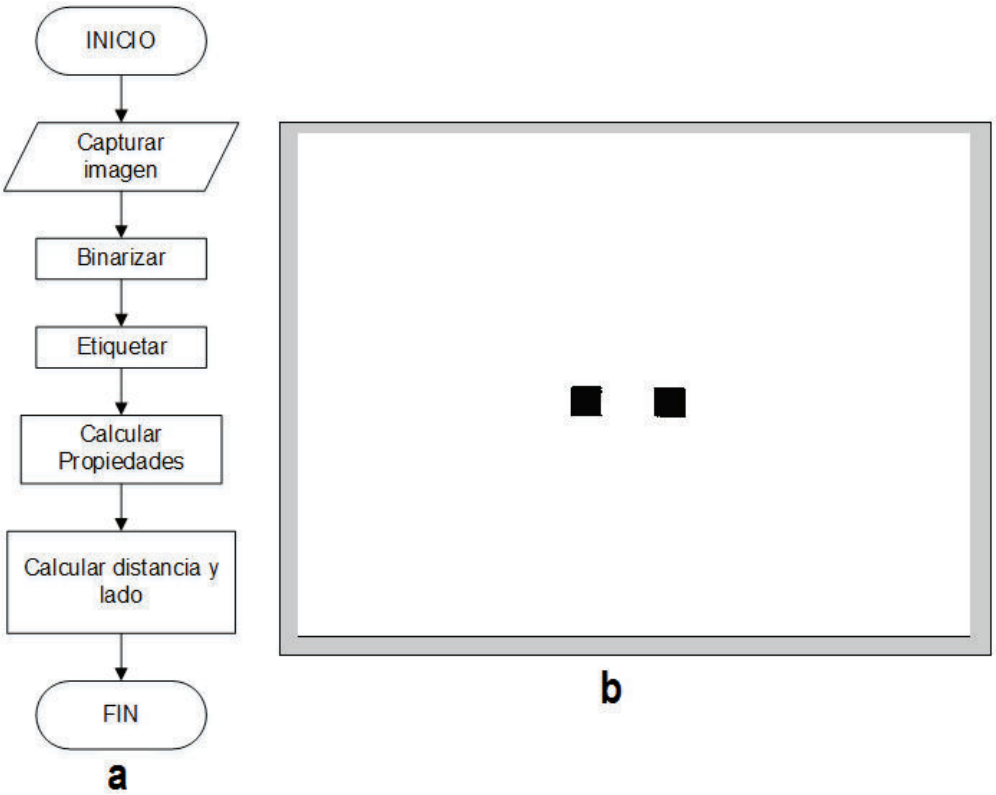
$$\boldsymbol{dist = redondeo}\left(\sqrt{(\boldsymbol{cent1}_x - \boldsymbol{cent2}_x)^2 - \left(\boldsymbol{cent1}_y - \boldsymbol{cent2}_y\right)^2}\right) \quad (3.1)$$

La otra característica utilizada es el área de los cuadrados, como son iguales se puede trabajar con cualquiera, se calcula el lado con la ecuación 3-2 que en futuros cálculos servirá para la compresión del entorno, este dato también tiene que ser redondeado, debido a que los valores de cálculos posteriores están en pixeles y estos se cuantifican en números enteros.

$$\boldsymbol{lado = redondeo}\left(\sqrt{\text{á}\boldsymbol{rea}}\right) \quad (3.2)$$

Otra aplicación que se le da a estos valores calculados es limitar los radios de en la búsqueda de las marcas para determinar la orientación de los robots, así como también para establecer un límite al avance de los robots a lo largo de las rutas para que el sistema no genere errores por exceso en los índices de las matrices que tienen los datos de la ruta.

A nivel de diagrama lógico se tiene el mostrado en la Figura 3-22a y el resultado de este algoritmo como imagen se puede observar en la Figura 3-22b.



a) Diagrama de Flujo b) Entorno con patrón.

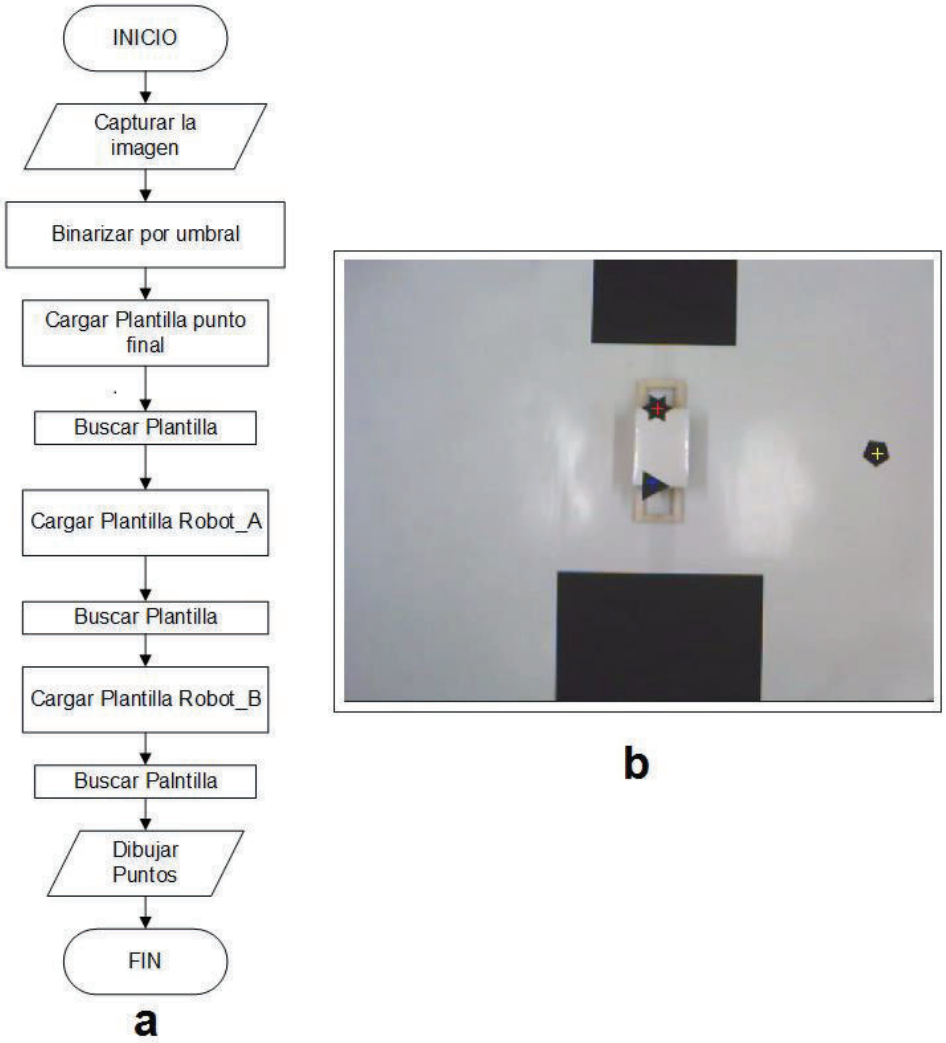
Figura 3-22 Bloque patrones.

3.4.3. BLOQUE ENTORNO PUNTO FINAL

Ahora se tiene que estructurar el entorno con el punto final, el objeto y los móviles, para nuevos entornos los algoritmos pueden volver a ejecutarse desde este punto, debido a que lo antes hasta aquí desarrollado ha sido para configurar propiedades.

En este bloque se captura la imagen y se la binariza, después se carga la plantilla del punto final y mediante la técnica de búsqueda de patrones se la ubica dentro del entorno, este mismo proceso se lo realiza con las plantillas que identifican los puntos finales de movimiento individual de los robots ubicando las posiciones para cada uno de ellos. De este proceso sobre la imagen adquirida se dibuja marcas que indicarán los valores obtenidos.

A nivel de diagrama lógico se lo puede apreciar en la Figura 3-23a y el resultado de la ejecución se aprecia en la Figura 3-23b.



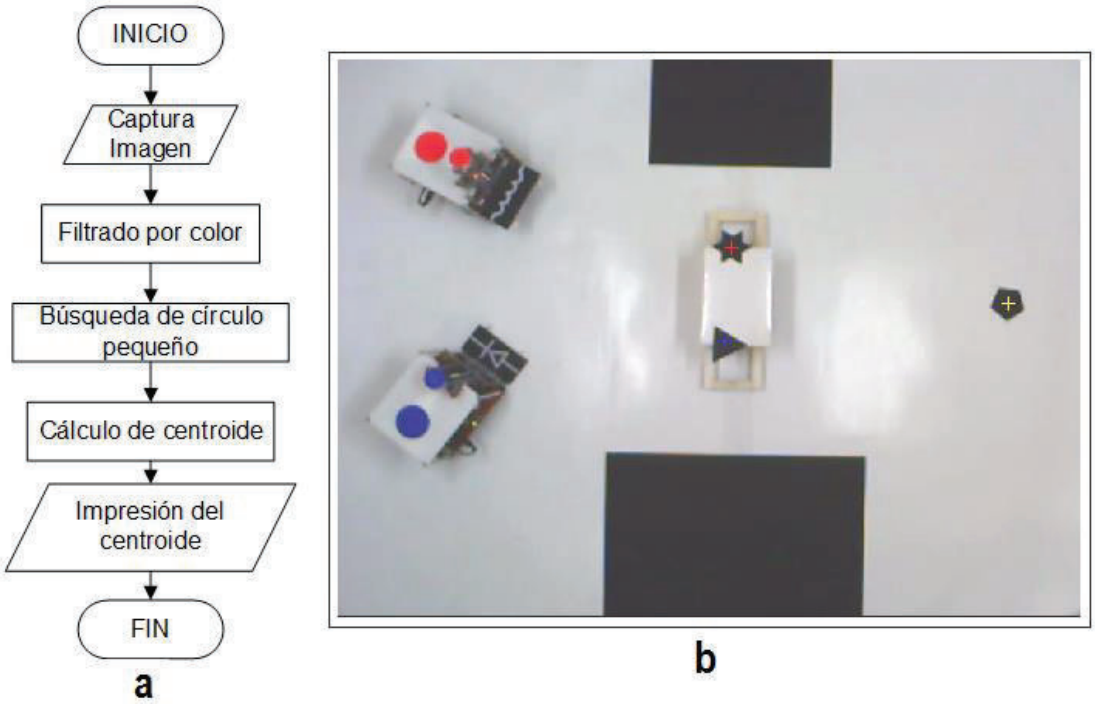
a) Diagrama de Flujo del algoritmo b) Imagen de ejecución del algoritmo

Figura 3-23 Bloque entorno punto final.

3.4.4. BLOQUE OBTENER PUNTOS

Se captura la imagen y se realiza un filtrado por color para únicamente obtener la información requerida de la imagen a analizar, independizando tanto color rojo como al color azul. Con cada una de las imágenes filtradas, se aplica un proceso de búsqueda de círculos, de donde se obtendrán dos datos, uno del círculo grande y otro del círculo pequeño. Con la información obtenida se evalúa para ubicar al círculo pequeño del mismo que de sacan sus propiedades y en específico su

centroide el mismo que será el punto de referencia. Por último, se grafica el punto obtenido. Todo este proceso se realiza para cada color, obteniéndose como resultado la figura 3-24b, el diagrama lógico se aprecia en la figura 3-24a.



a) Diagrama de flujo del algoritmo b) Ejecución de Algoritmo obtener puntos

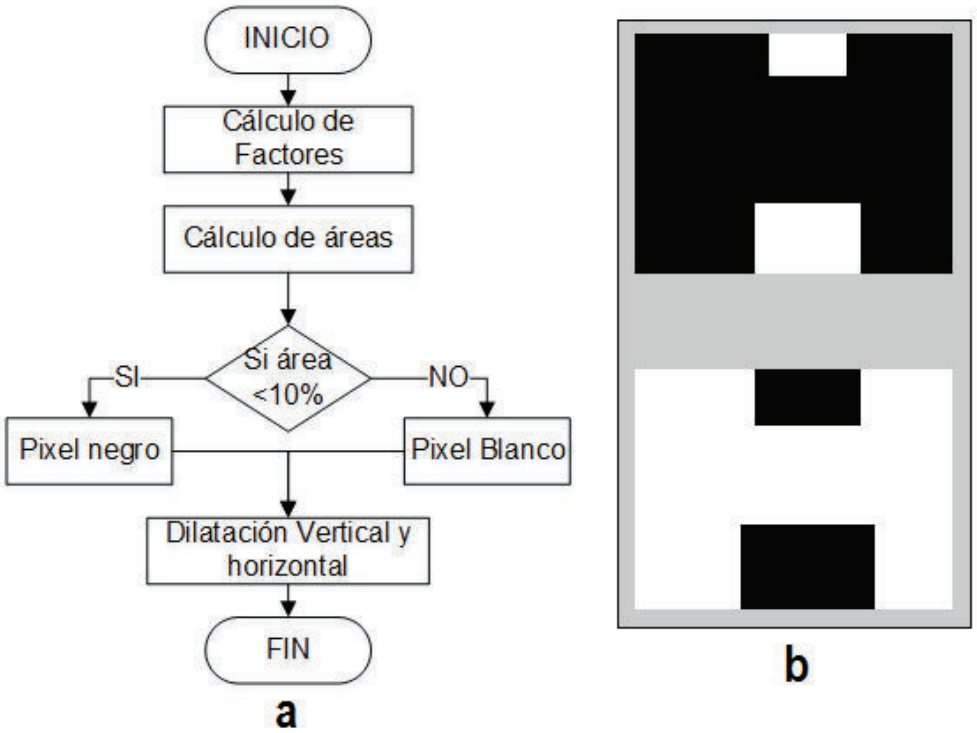
Figura 3-24 Algoritmo obtener puntos del entorno.

3.4.5. BLOQUE REDUCCIÓN

En este bloque se procesa la imagen del entorno con el algoritmo de reducción en base de áreas, de esta manera se obtiene una matriz de menor grado, recordar que el valor de la medida en reducción corresponde al valor de lado calculado con el bloque patrones.

En los límites de los objetos que conforman el espacio estructurado se da una holgura para el movimiento de los robots, esto se lo obtiene mediante una operación morfológica de dilatación. Lo último es colocar los puntos de inicio de movimiento y final de movimiento del Robot_A y del Robot_B dentro del espacio que ha sido reducido.

Este trabajo se realiza para obtener una matriz más pequeña para el momento de buscar la ruta, debido a que el algoritmo que busca la ruta requiere de una gran cantidad de procesamiento de información. El detalle de este algoritmo se lo parecía en la Figura 3-25.

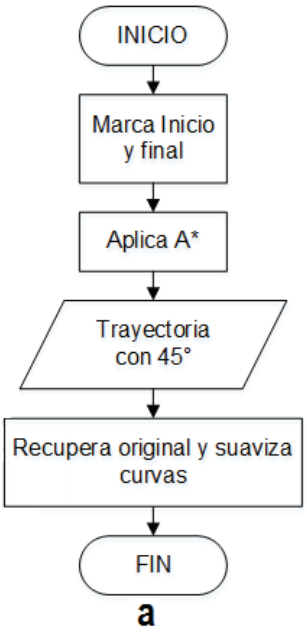


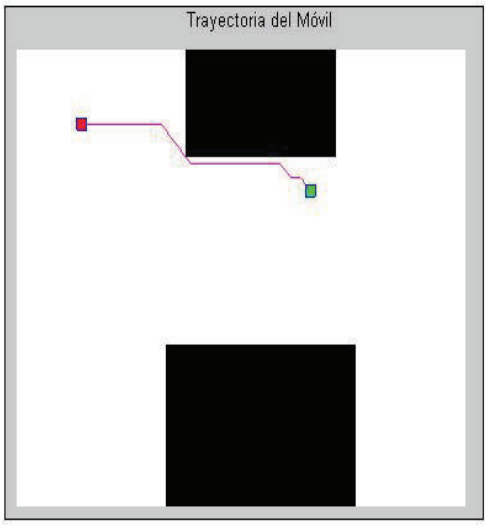
a) Diagrama de flujo del algoritmo b) Ejecución de Algoritmo reducción

Figura 3-25 Algoritmo reducción.

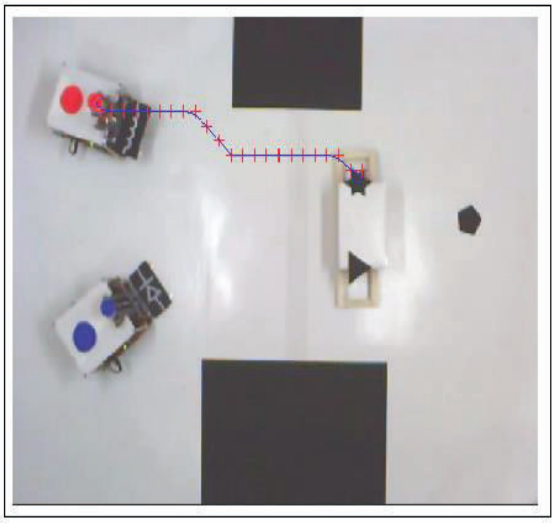
3.4.6. BLOQUE RUTA ROJO

A la imagen reducida se le aplica el algoritmo A* para que el sistema encuentre la ruta más corta para llegar desde el punto de inicio de movimiento, marca sobre la horquilla del Robot_A hasta el punto final, marca sobre el objeto para el mismo robot. La lógica completa se puede apreciar en la Figura 3-26.





b

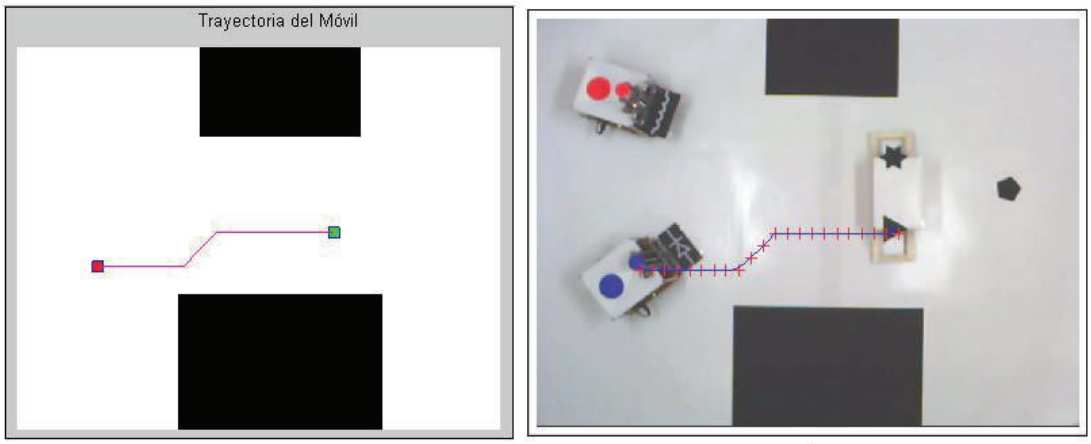


c

a) Diagrama de flujo b) Ruta en imagen comprimida c) Ruta reconstruida

Figura 3-26 Ruta Robot_A.

De este algoritmo se obtiene una ruta con ángulos múltiplos de 45° sobre la imagen reducida, esta tiene que ser reconstruida al entorno original y suavizados los ángulos, para esto se utilizan las curvas de Bézier de primer grado y segundo grado según sea el caso, de esta manera se obtiene la ruta para el Robot_A. Este mismo bloque se repite para el Robot_B, color azul. Figura 3-27 a,b.



a b

a) Ruta entorno reducido b) Ruta en entorno real.

Figura 3-27 Ruta Robot_B.

3.4.7. BLOQUE MOVIMIENTO ROJO

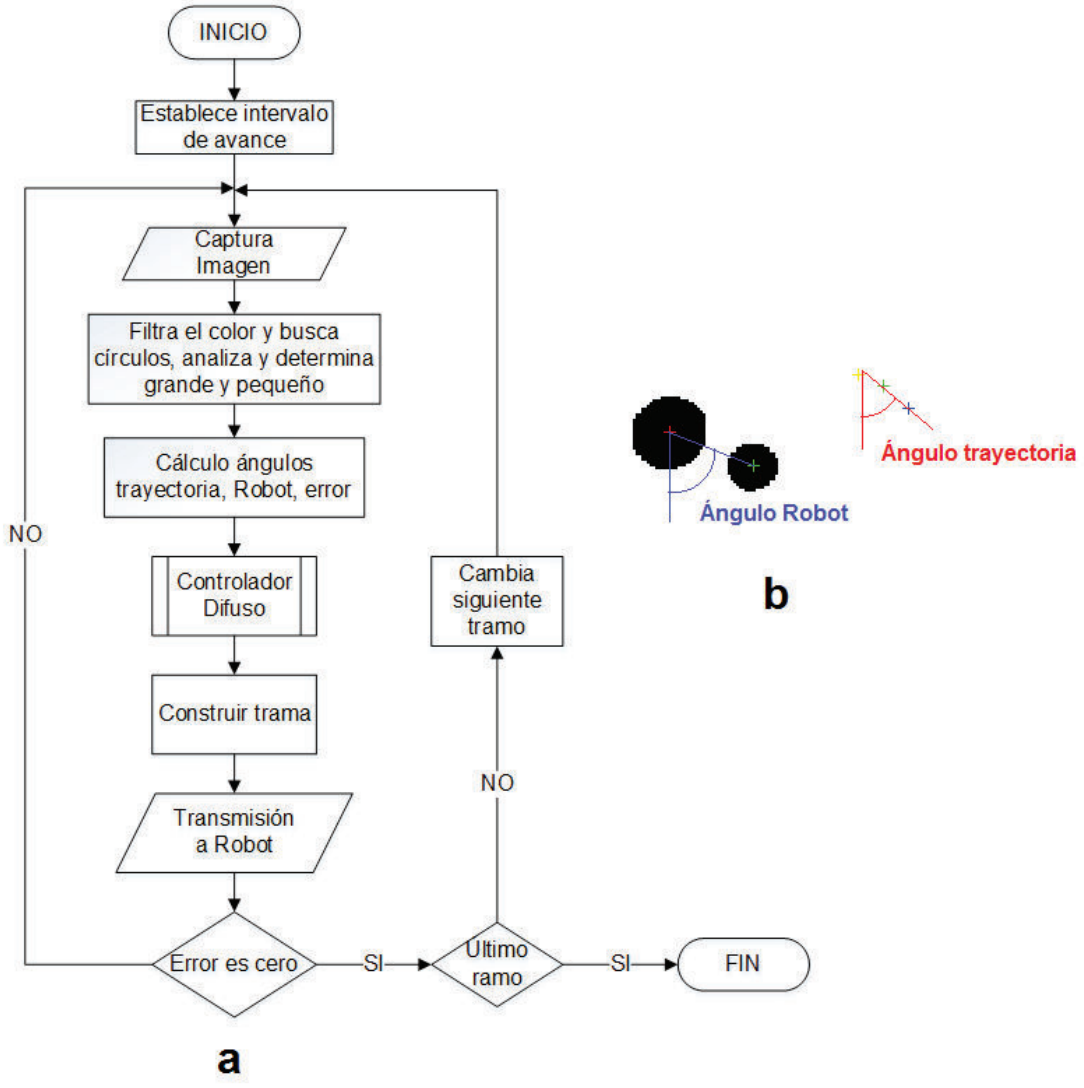
Con la ruta reconstruida se da la navegación esta consiste en ir avanzado por tramos dentro de la misma, para esto se establece un delta de avance y en base del mismo se irá navegando por el vector que contiene la ruta. Después de capturar la imagen del entorno, se realiza un filtrado de color para obtener únicamente el color rojo, a la imagen resultante se binariza y se buscan los círculos que contenga, se analizan los valores de sus propiedades para ubicar el círculo grande y el pequeño. Con las coordenadas de los centroides se calcula en ángulo que tiene el robot, para lo cual se apoya de la función trigonométrica tangente y su inversa. De tal forma que se obtienen valores de ángulos positivos y negativos con respecto a un eje central en la imagen.

Con la ruta se procede de la misma manera, se obtienen dos puntos a un cierto intervalo de separación en el vector que contiene las coordenadas de la ruta. Con

los puntos se obtiene el ángulo del vector que forman los dos puntos apoyado de sus coordenadas y de la función trigonométrica tangente y su inversa.

Con los dos valores de ángulos se calcula el error de la diferencia entre estos, el mismo que ingresa al sistema difuso para que calcule la acción de control.

$$\boldsymbol{error_{angulo} = angulo_{trayectoria} - angulo_{movil}} \qquad (3.3)$$



a) Diagrama de flujo b) Disposición de ángulos

Figura 3-28 Movimiento del Robot.

La respuesta del sistema difuso es la acción de control que es convertida en una trama y enviada al Robot_B para que sea interpretada, esto se realiza el número de veces que sea necesario hasta que el valor del error de ángulos sea igual a cero, en ese momento el robot está listo para avanzar y continuar con el movimiento en el siguiente tramo.

Este trabajo se realiza la cantidad de veces que sean necesarias hasta que el robot llegue a una distancia calculada de su punto objetivo. La distancia es determinada en base de los datos obtenidos del algoritmo patrón.

El mismo procedimiento se realiza también con el Robot_B. cuando se haya completado la totalidad de este algoritmo, se tendrá a los dos agentes en frente del objeto listos para elevarlo y transportarlo.

3.4.8. SISTEMA DIFUSO

Para comprender como está estructurado el sistema difuso, primero se tiene que establecer como está concebido el sistema.

El sistema es una matriz de 640x480 pixeles, la misma que tiene su origen en la parte superior izquierda y desde allí medidos únicamente valores positivos de las posiciones de los pixeles, por tal razón para establecer una orientación dentro del entorno se lo ha dividido en dos hemisferios los mismos que proveen de valores de medida de los ángulos positivos y negativos. Estos valores definen la orientación que tendrá el robot dentro del entorno con respecto al eje (meridiano). Figura 3-29.

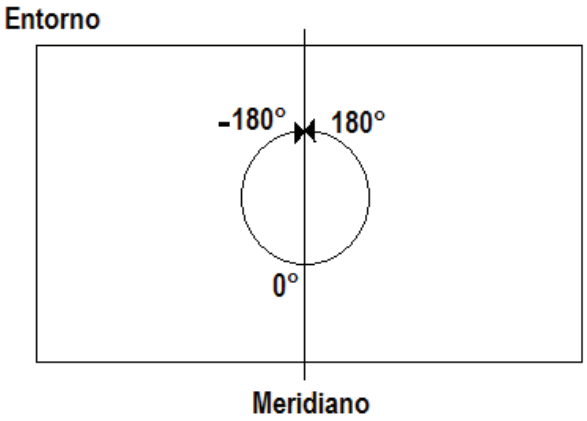


Figura 3-29 Entorno dividido.

Con la orientación dentro del entorno se pueden establecer las reglas de inferencia que ayudarán al cálculo las acciones de control del controlador difuso, para los cual la entrada al controlador será el error existente entre el ángulo de la ruta, que es el set point y el ángulo del robot, variable medida. Con esta premisa se establece que cuando el error sea cero, el robot deberá avanzar hacia adelante hasta llegar a su punto de destino, si el error es positivo deberá compensar una llanta con valor positivo y la otra con valor negativo, caso contrario si el error es negativo deberá invertir las acciones de control sobre las llantas. Para apreciar con más detalle cómo se encuentra estructurado el sistema, se tiene la Figura 3-30.

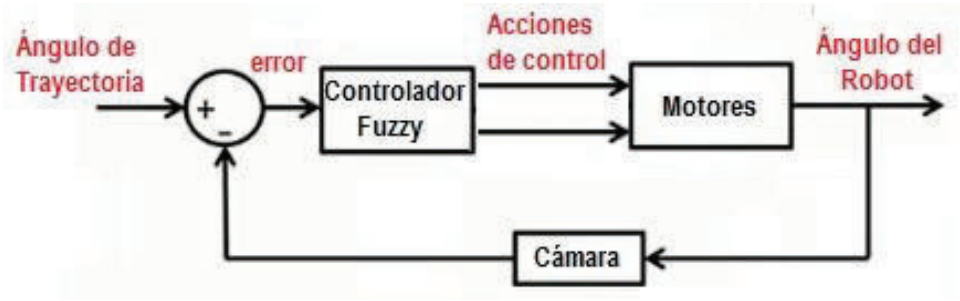


Figura 3-30 Estructura de control del sistema.

Para el diseño de este sistema se utilizó la herramienta de control difuso de Matlab por la rapidez que presta esta para el desarrollo, obteniéndose la imagen mostrada en la Figura 3-31.

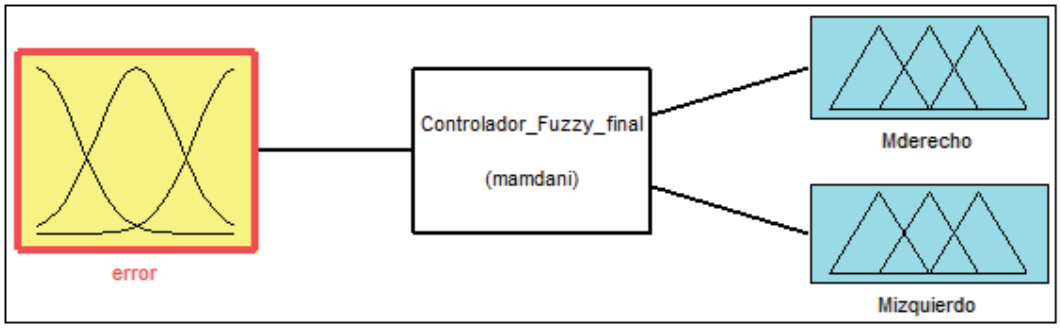


Figura 3-31 Controlador difuso MatLab.

Para llegar a la estructura mostrada en la Figura 3-31, se realizaron varios ensayos con distintas configuraciones de controladores hasta obtener el controlador definitivo, este es con reglas de inferencia de Mamdami, debido a que: este controlador es muy intuitivo, posee una gran aceptación entre los usuarios y sobre

todo es muy adecuado para la intervención humana. Al controlador le ingresa la variable error, la misma que es calculada de forma externa al controlador, en la Tabla 3-4 se describe a esta variable y cada uno de sus componentes.

Tabla 3-4 Variable de entrada error.

Variable lingüística de Entrada: error			
Universo Discurso: [-360 360]		Unidad: Grados	
Definición	Términos lingüísticos	Rango	Función de membresía
Error Negativo	EN	[-360 -360 -100 -0]	Trapezoidal
Error Central	EC	[-0.5 1 0.5]	Triangular
Error Positivo	EP	[0 100 360 360]	Trapezoidal

Se puede observar gráficamente a esta variable en la Figura 3-32, en la misma se aprecia que se utilizan funciones trapezoidales, debido a que cuando los errores son grandes lo que se espera es que la acción de control sea la más grande posible para que la corrección sea rápida, esto en la base menor del trapecio, en cambio en la rampa se esperan valores pequeños de corrección ante errores pequeños y que conforme estos van creciendo, la acción de control también haga lo mismo. Y se usa triangular para el error central (EC), debido a que en esta se espera zona el rango para el error es muy pequeño y se espera una acción de control grande, esto se obtiene con una pendiente muy pronunciada.

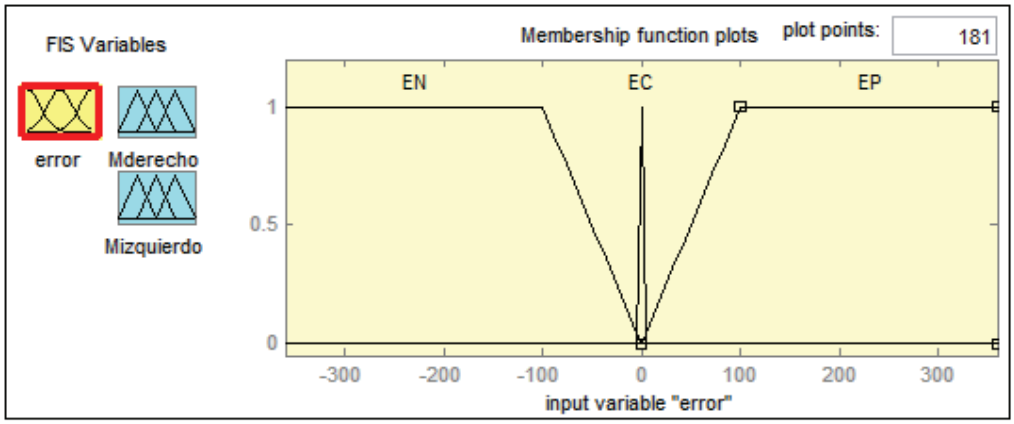


Figura 3-32 Variable de entrada error.

Para el caso de las variables de salida del controlador se tienen dos, la una corresponde al motor izquierdo y la otra al motor derecho, estas cumplen una peculiaridad que las dos son iguales a pesar que las acciones de control que realizan sobre los robots sean diferentes en su acción, la forma de estas variables de salida se puede apreciar en las Tablas 3-5 y 3-6, así como también el detalle gráfico en las Figuras 3-33 y 3-34.

Tabla 3-5 Variable Salida motor derecho.

Variable lingüística de Salida: Motor derecho			
Universo Discurso: [-10 10]		Unidad: Voltaje continuo.	
Definición	Términos lingüísticos	Rango	Función de membresía
Motor derecho negativo	MDR	[-10 -10 -5 0]	Trapezoidal
Motor derecho positivo	MDRM	[0 5 10 10]	Trapezoidal

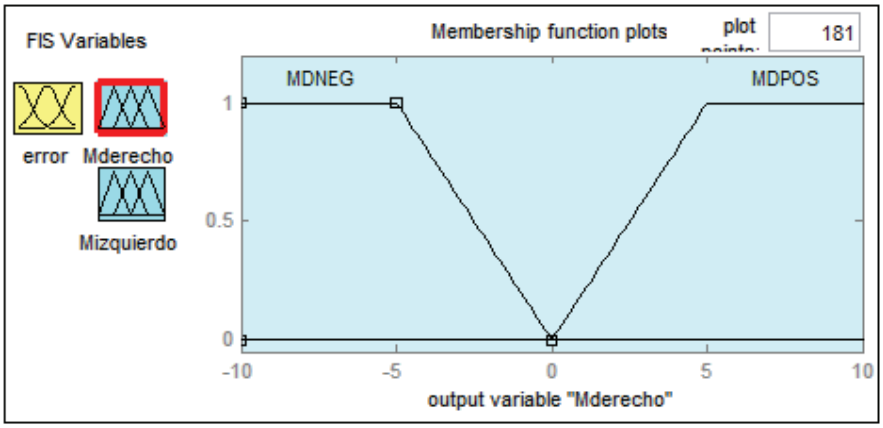


Figura 3-33 Variable de salida motor derecho.

Tabla 3-6 Variable salida motor izquierdo.

Variable lingüística de Salida: Motor izquierdo			
Universo Discurso: [-10 10]		Unidad: Voltaje continuo.	
Definición	Términos lingüísticos	Rango	Función de membresía
Motor izquierdo negativo	MINEG	[-10 -10 -5 0]	Trapezoidal
Motor izquierdo positivo	MIPOS	[0 5 10 10]	Triangular

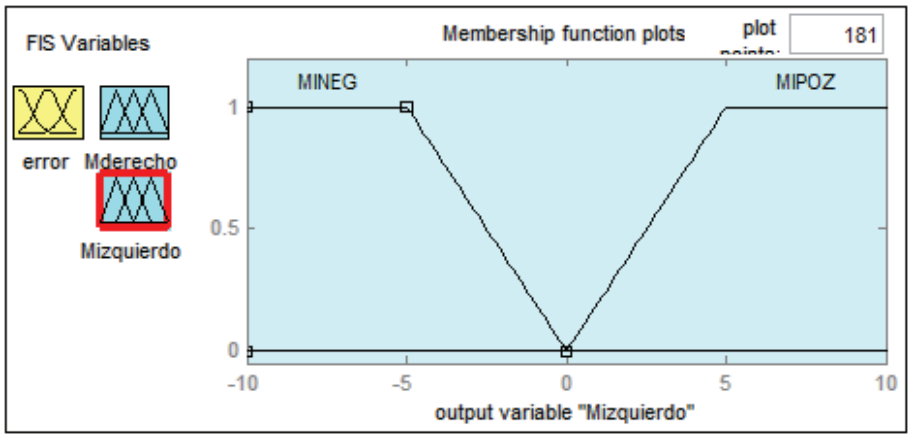


Figura 3-34 Variable salida motor izquierdo.

Se han definido las reglas de control para relacionar los términos lingüísticos del sistema, como se muestra en la Tabla 3-7. Considerando la división del espacio en los dos hemisferios para el cálculo del error.

Tabla 3-7 Reglas de Control.

EN	EC	EP
MDPOS	MDPOS	MDNEG
MINEG	MIPOS	MIPOS

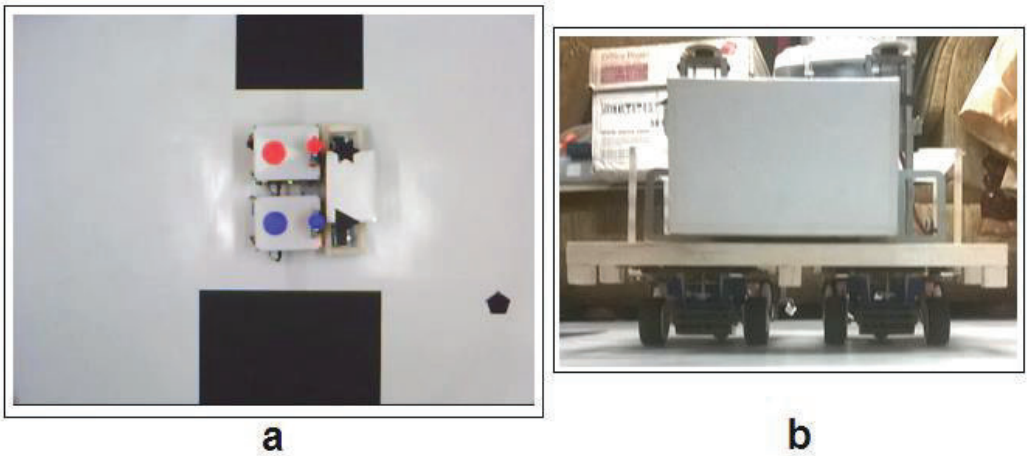
Para la interpretación de estas reglas se tiene que seguir la siguiente lógica:

Si (error es EC) **entonces** (motor derecho es MDPOS) (motor izquierdo es MIPOS)

De esta forma se puede aplicar a toda la tabla para la comprensión de su contenido. Por último, en el sistema de control requiere obtener un valor único, debido a que del proceso de inferencia se obtiene un conjunto de valores, este proceso lo constituye el método de defusificación, para de este controlador se ha utilizado el método del centroide.

3.4.9. BLOQUE ELEVACIÓN DE OBJETO

Una vez que se tenga a los dos robots ubicados en las posiciones finales de las rutass de sus movimientos iniciales se tiene que elevar al objeto, para lo cual se envían repetidamente instrucciones de elevación de a uno y otro de los robots de forma alternada, primero al Robot_A, luego al Robot_B, el movimiento es coordinado debido a que en función de interrupciones del controlador de cada agente se eleva al objeto un espacio vertical a la vez, Figura 3-35.



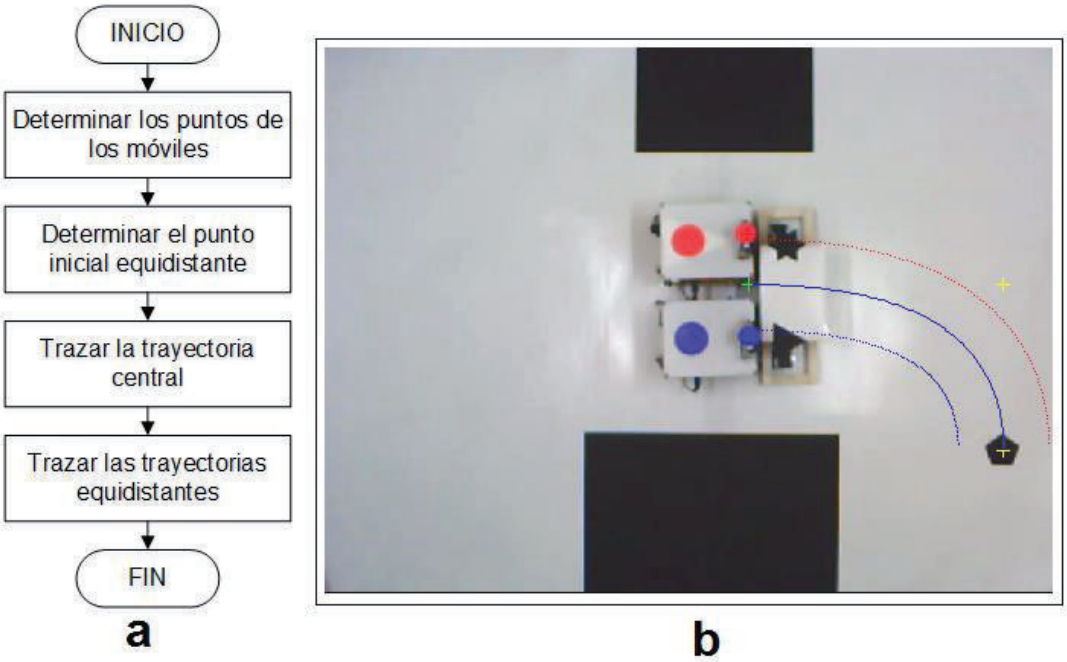
a) Vista superior objeto elevado b) Vista frontal de objeto elevado

Figura 3-35 Elevación del objeto.

3.4.10. BLOQUE RUTA COOPERATIVA

Para la determinación de la ruta cooperativa se sigue la siguiente lógica: primero se establece en punto medio entre los puntos de movimiento de los dos agentes, círculos pequeños, este punto corresponde al punto inicial de la ruta, el punto final corresponde al ya buscado al inicio del algoritmo, punto que se buscó mediante la búsqueda de la plantilla establecida. Mediante las curvas de Bézier se traza la ruta entre los dos puntos, estableciendo un tercer punto intermedio en base de la coordenada “x” del punto final y punto en coordenada “y” del punto inicial.

Una vez mediante un algoritmo de equidistancia a los puntos de la ruta obtenida, se establecen las rutas paralelas para cada uno de los agentes, considerando como distancia a la mitad de la separación entre los dos agentes, hay que recordar que este valor ya fue calculado previamente en el bloque de parámetros. Figura 3-36.



a) Algoritmo rutas cooperativas b) Rutas cooperativas trazadas

Figura 3-36 Rutas cooperativas.

3.4.11. BLOQUE MOVIMIENTO COOPERATIVO

Para el movimiento cooperativo se ha integrado la forma de movimiento de cada uno de los agentes por las nuevas rutas cooperativas calculadas, es decir el procesamiento inicia con la captura de la imagen, el filtrado en base de colores, una binarización de la imagen filtrada, posterior a esto se ubican los centroides de los círculos para el cálculo del ángulo de orientación del agente, de la ruta se determinan dos puntos para el movimiento y de la misma manera se calcula el ángulo.

Con estos dos ángulos se calcula el error, ahora se ingresa el error al sistema difuso de control de la respuesta se construye la trama que es transmitida al primer agente,

este proceso se lo repite hasta que el error sea cero, en ese instante el primer agente hace su movimiento.

Una vez concluido el movimiento del primer agente, se procede de la misma forma con el segundo agente realizando los mismos pasos hasta que haga su movimiento. Los pasos aquí descritos se los realizan tantas veces como sean necesarias hasta que el objeto haya llegado al final de su ruta.

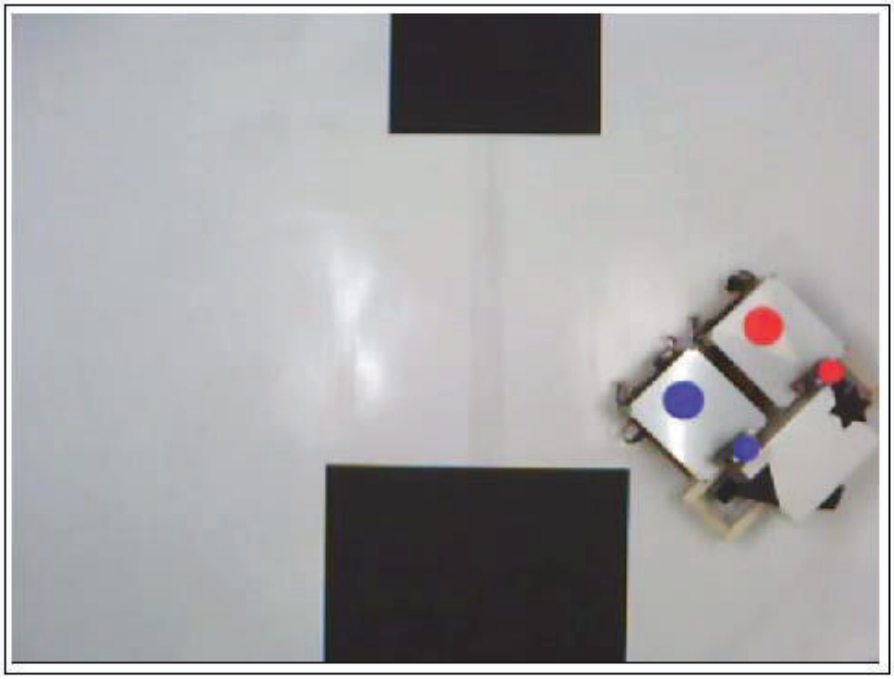


Figura 3-37 Imagen robots en punto final.

El resultado de este algoritmo se lo puede apreciar en la Figura 3-37 cuando el objeto ha llegado a su punto de destino, a nivel lógico se puede apreciar este algoritmo en la Figura 3-38.

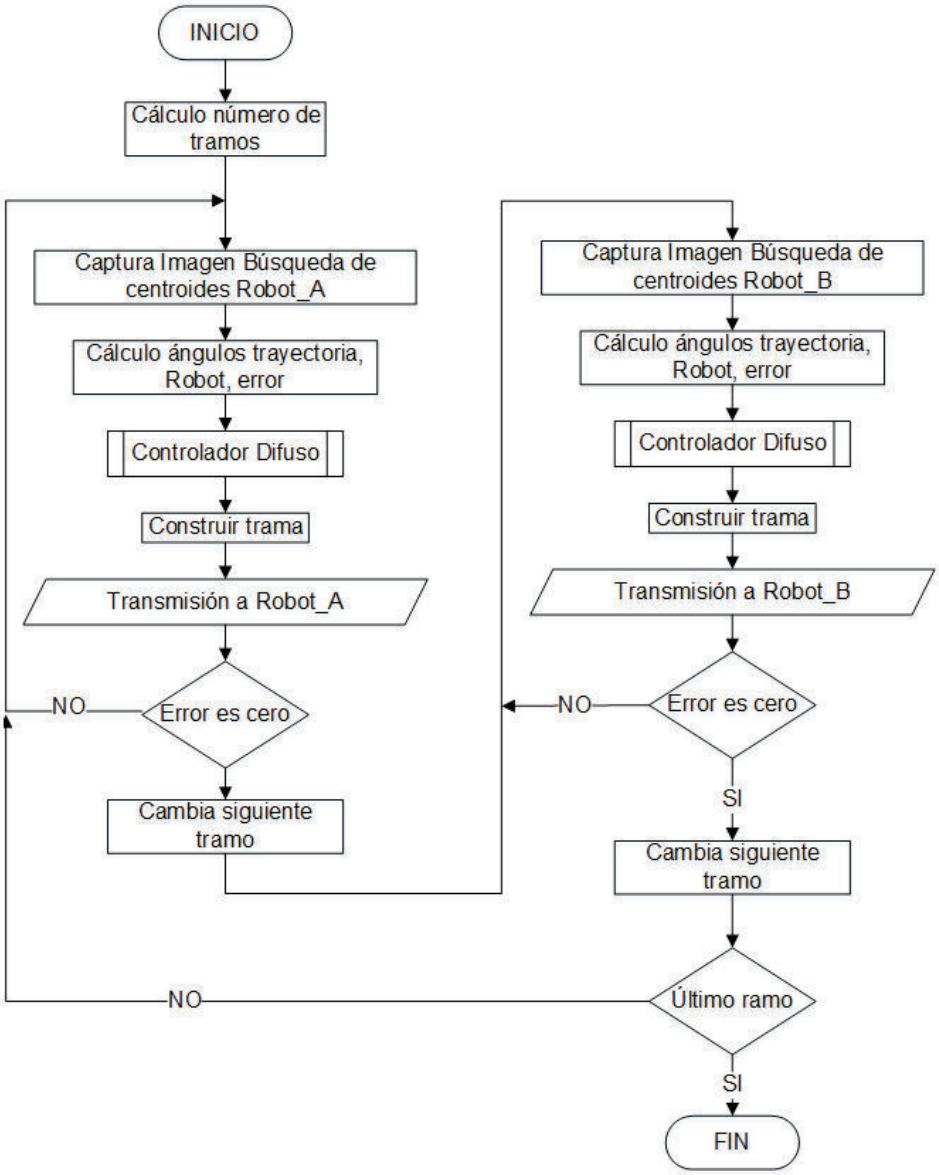


Figura 3-38 Algoritmo de movimiento cooperativo.

3.4.12. BLOQUE FINALIZACIÓN

En este bloque se realizan dos acciones: la primera es liberar a la cámara para terminar con la comunicación de la misma vía USB, la segunda es detener la comunicación inalámbrica con los módulos bluetooth.

Una vez liberadas las dos comunicaciones se eliminan los objetos creados para este propósito. La razón de hacer estas acciones es para que, en pruebas de operación futuras del sistema, este no genere un error porque los canales de comunicación y captura de imagen están relacionados con otras aplicaciones.

3.5. DISEÑO DE SOFTWARE DE LOS ESCLAVOS

Como se mencionó en la sección correspondiente al diseño del sistema electrónico, los controladores que se encuentran a bordo de cada uno de los agentes en una placa ARDUINO MEGA 2560, la misma que para su programación se utiliza un software propietario llamado ArduinoSoft, este un software que opera bajo ambiente Windows, en el cual se ha implementado un algoritmo que es común para cada uno de los agentes, donde se establece la diferencia entre el uno y otro es en los receptores Bluetooth, en donde cada uno de ellos se los programó mediante comandos AT y se seteo al primero con el nombre de Robot_A, robot con marcas rojas y el segundo tiene el nombre de Robot_B, robot con marcas azules.

El algoritmo procede de la siguiente manera: lo primero es definir las variables de entrada y constantes para la operación del sistema, así como también las funciones que tienen que hacer cada uno de los pines, los parámetros para la comunicación y para las interrupciones.

El sistema se encuentra en un estado de reposo hasta que mediante una interrupción generada por la comunicación serial entre el receptor bluetooth y el microcontrolador, se recibe el dato y este se va acumulando en un vector con la información recibida, siempre y cuando el valor receptado sea mayor que cero, debido a que la trasmisión es en códigos ASCII.

La forma del vector de recepción tiene una estructura definida, como se muestra en la Figura 3-39.

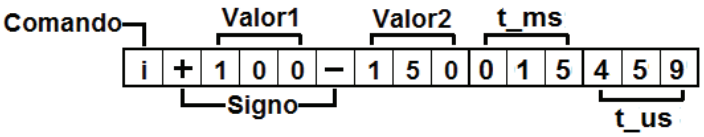


Figura 3-39 Forma de vector recibido.

El sistema tiene tres opciones de comando, debido a que las acciones que realizará serán: subir horquilla, bajar horquilla o mover las ruedas, esto se aprecia de forma más explicativa en la Tabla 3-8.

Tabla 3-8 Tabla de comandos existentes.

Identificador	Acción
“i”	Movimiento de llantas
“s”	Subir horquilla
“b”	Bajar la horquilla

Se ha considerado como eje del algoritmo a la letra izquierda, de allí que para el movimiento de las llantas se considera el comando “i”, cuando el microcontrolador recibe este comando, esto le indica que es el movimiento de las llantas. El siguiente dato corresponde al signo, de tal forma que si es “+”, la llanta realiza un movimiento hacia adelante, en cambio si el signo es “-” la llanta realizará un movimiento hacia atrás. Como componentes restantes del vector se tienen los bits de valor, en donde los bits de valor 1 corresponden a la velocidad del motor izquierdo y los de valor2 al motor derecho. Estos dos valores corresponden a las modulaciones en acho de pulso a ser aplicadas a los motores, de estos depende la variación del avance del movimiento. El siguiente espacio del vector corresponde a los tiempos de acción de las señales sobre los motores, esto se encuentra definido en dos tiempos: el primero se encuentra medido en milisegundos, que corresponderá a la acción fuerte de control y el segundo tiempo en microsegundos, que correspondería a una acción fina de control. Estas dos acciones se las ejecuta para tener una acción de control más precisa para minimizar el error de posición en el movimiento de los robots.

En el caso de la horquilla, esta trama únicamente se limita al comando, debido a que el control de la cantidad de movimiento lo realiza el sistema de marcas colocadas sobre la carrera de la horquilla y una interrupción del microcontrolador. La estructura lógica de este algoritmo se lo puede apreciar en la Figura 3-40.

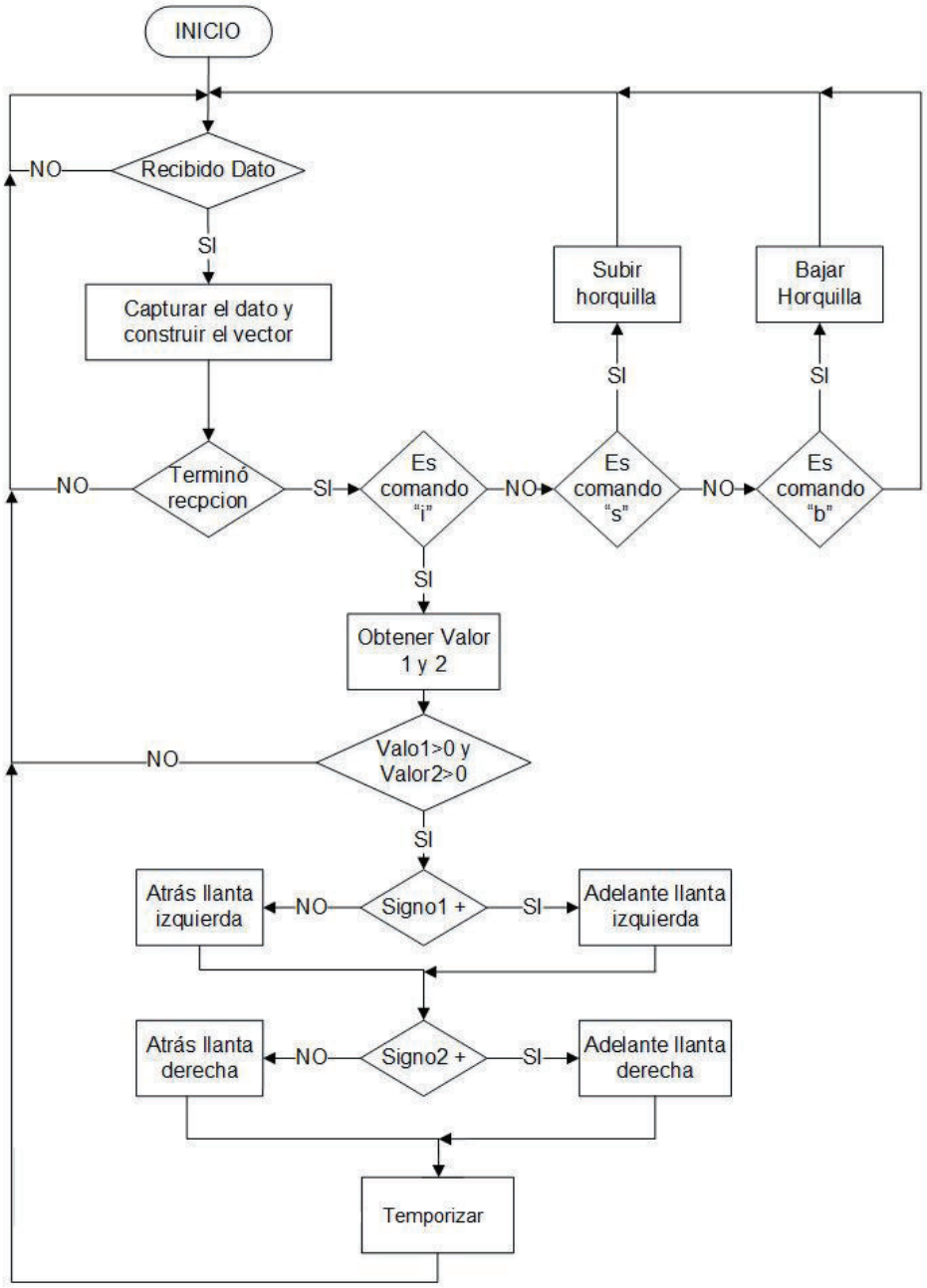


Figura 3-40 Algoritmo de robots esclavos.

Para la utilización de este sistema se puede referir al anexo 1, en donde se encuentra el manual de usuario.

CAPÍTULO 4

4.1. PRUEBAS Y RESULTADOS

Para la validación de este trabajo se han realizado varias pruebas de operación, las mismas que se describen en este capítulo, así como también se realiza un análisis de los resultados obtenidos. Hay que recordar que para la operación del sistema se tienen varias fases de forma general: movimiento de los agentes solos, elevación del objeto a transportar y por último el traslado del objeto de forma cooperativa.

4.2. PRUEBA 01: MOVIMIENTO LIBRE.

La primera prueba realizada fue para avaluar el movimiento realizado por los robots de manera independiente, así como también el comportamiento del sistema de control. Para este fin se crearon rutas virtuales a lo largo de todo el entorno para que las sigan los agentes, obteniendo los siguientes resultados:

Durante la realización de esta prueba se detectó un primer problema con el sistema, esto se dio debido a la altura en la cual se encontraban las marcas de orientación de los robots. Estas se encontraban a diferente nivel que generaban un problema de paralaje con la imagen adquirida por la cámara.

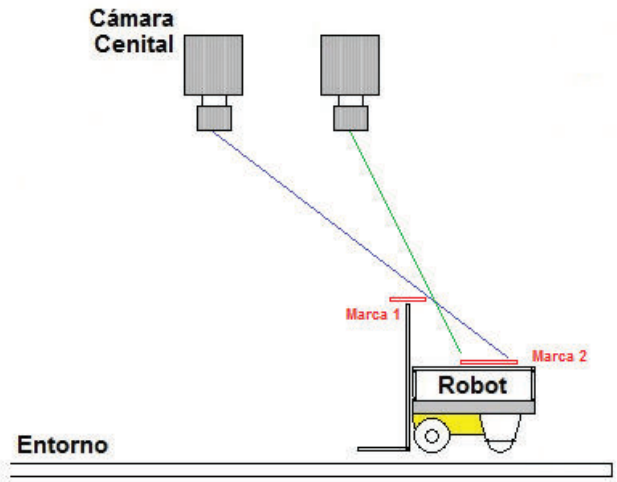


Figura 4-1 Toma a diferente posición.

Esta condición de falla en el hardware hacía que las imágenes adquiridas por la cámara entreguen ángulos que estaban totalmente fuera de la real orientación de

los robots y como consecuencia no se tenía un buen control del sistema, generando errores muy grandes, incluso se dio el caso que la imagen adquirida por la cámara fusionaba a las dos marcas y por consiguiente perdiendo toda la información de orientación.

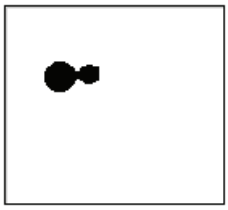
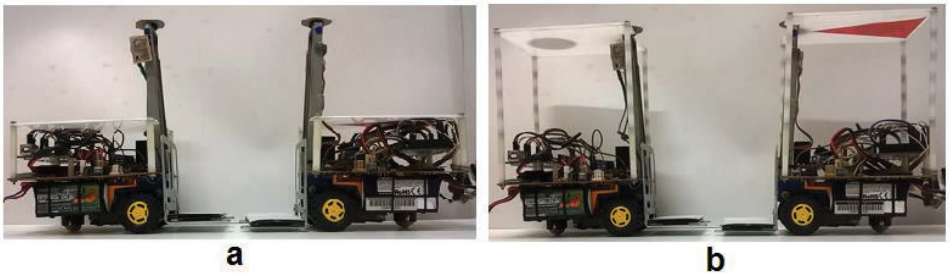


Figura 4-2 Imagen fusionada.

La solución a este problema fue ubicar las dos marcas de orientación a la misma altura, para un mejor detalle del análisis realizado, se puede referir al anexo 2.



a) Altura original de las marcas de orientación. b) Modificación a la altura de las marcas

Figura 4-3 Modificación a la altura de marcas.

4.2.1. RUTA HORIZONTAL ROBOT_A.

Para un primer análisis se ha generado una ruta virtual horizontal, se denomina virtual debido a que no toma valores del entorno para su trazado.

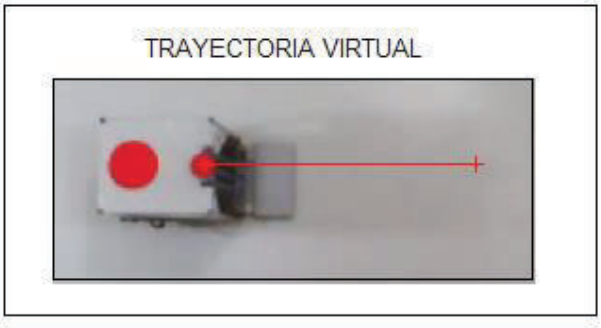


Figura 4-4 Ruta virtual.

Con las primeras configuraciones del sistema se tiene el movimiento mostrado en la Figura 4-5:

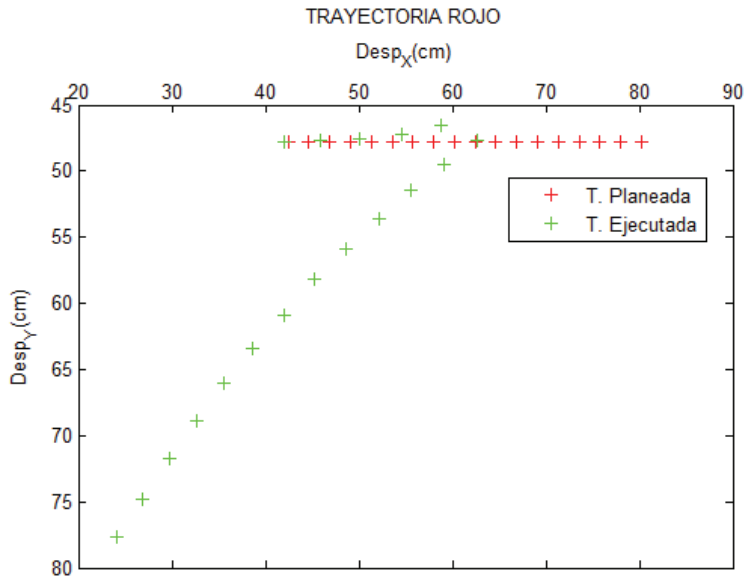


Figura 4-5 Ruta inicial del robot rojo.

Como se observa con los valores iniciales planteados para el controlador el comportamiento del robot es totalmente errático, hasta llega al punto invertir su movimiento con respecto a la ruta. Por consiguiente, el error generado en el ángulo también será enorme, esto se puede apreciar en las Figuras 4-6 y 4-7.

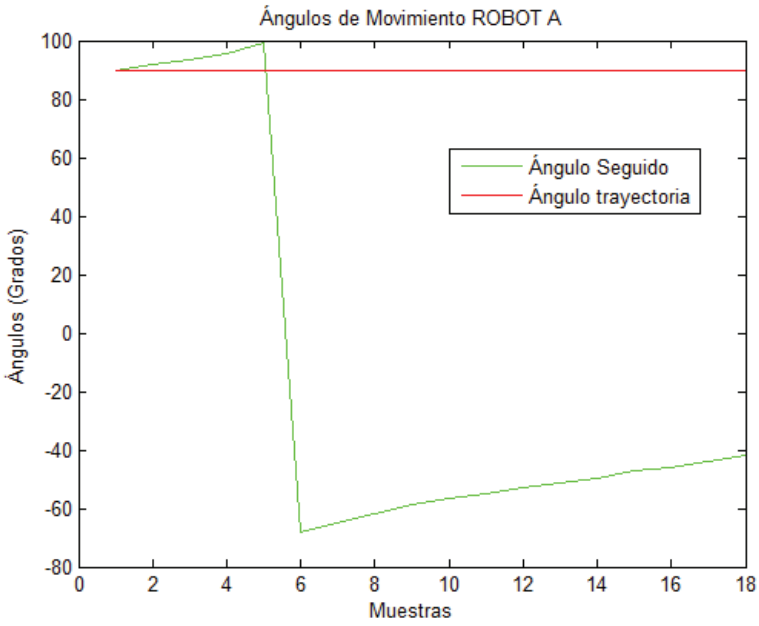


Figura 4-6 Ángulos generados robot rojo.

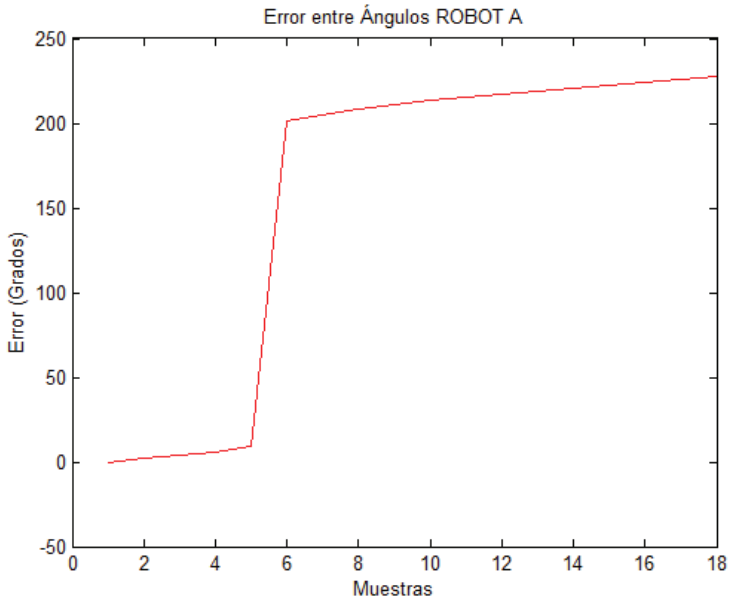


Figura 4-7 Error generado entre los ángulos.

Por lo sucedido en esta primera experimentación se tienen que realizar cambios al algoritmo, lo primero será garantizar que siga la línea recta de la ruta, para esto se hicieron varias experimentaciones modificando los valores del controlador difuso para minimizar el error del ángulo, con el cambio que más se pudo apreciar la mejora fue cuando se alteraron los valores de la variable lingüística error central y en específico el conjunto difuso error central, los nuevos valores se tiene en la tabla 4-1.

Tabla 4-1 Variable lingüística del error.

Términos lingüísticos	Rango	Función de membresía
EC (Original)	[-10 -5 5 10]	Trapezoidal
EC (Corregido)	[-1 -0.5 0.5 1]	Triangular

De la Tabla 4-1, se puede observar que el rango para el error al momento del movimiento, cuando el error es cero, es de tipo trapezoidal y con valores de [-10 -5 5 10], de esto se entiende que si el error se encuentra entre -10° y 10° al momento que el agente va a moverse, razón por la cual se tiene como consecuencia un error grande, por consiguiente, será necesario minimizar este rango para que el error sea más pequeño.

Después de varias pruebas con rangos más pequeños se ha obtenido un valor de [-1 -0.5 0.5 1] para cuando el error es cero y cambiando a una función de membresía de tipo triangular.

La prueba realizada con el nuevo controlador entrega la gráfica de la Figura 4-8 para la ruta.

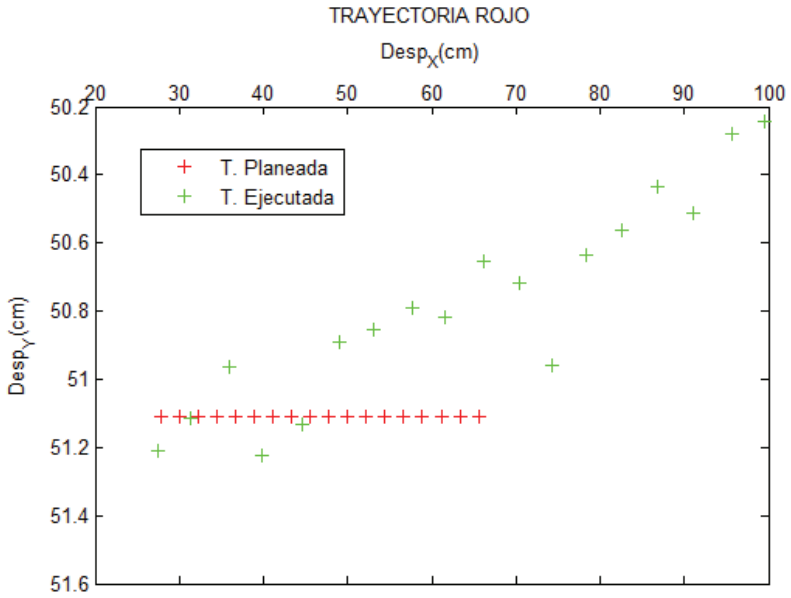


Figura 4-8 Ruta corregida del controlador.

De esta gráfica se aprecia que existe un error en la posición, se puede notar que los valores de desplazamiento realizado por el robot son mayores a los esperados para que cubra la ruta, por esta razón se deberá estimar el tiempo de acción de los actuadores para que dependiendo del mismo el desplazamiento sea mayor o menor.

En la Figura 4-9, se observa que la ruta tiene un ángulo constante de 90° a lo largo de todo su recorrido, existe una diferencia aproximada de ±1° con respecto a las rutas seguidas, mejorando totalmente con respecto al controlador anterior.
Para ratificar esto se tiene la gráfica 4-10, en la que se aprecia que el error no supera del ±1°, esta corrección hace que el sistema se encuentre con un controlador sintonizado.

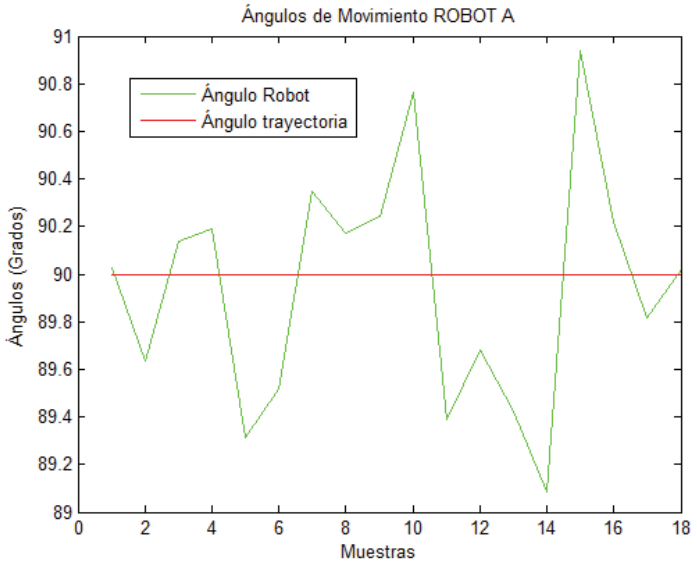


Figura 4-9 Ángulos de movimiento.

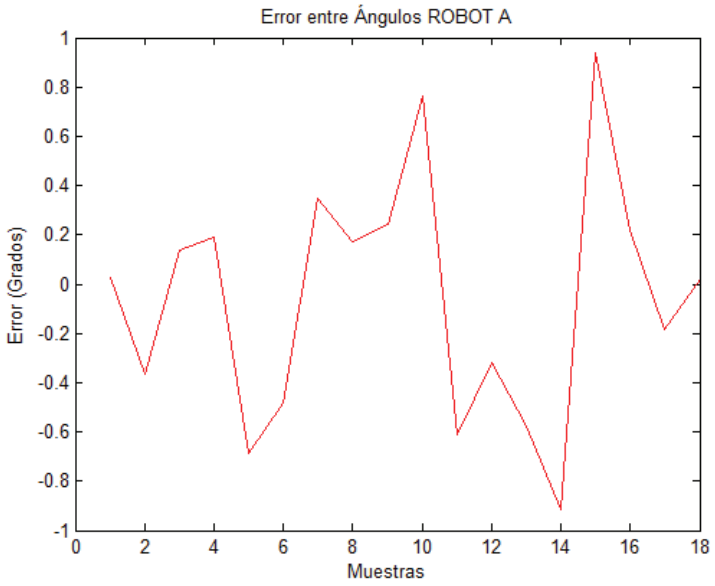


Figura 4-10 Error entre ángulos.

Como se mencionó anteriormente el problema ahora es de posición, se presume debido a que el tiempo del movimiento o accionamiento de los motores no coincide con el valor que tienen que moverse dentro de la ruta. Los valores de tiempo de acción de los motores han sido fijados en cada uno de los robots en base de la experiencia, pero estos valores están errados, para lo cual es necesario hacer un

estudio del desplazamiento de los robots con respecto al tiempo, es decir calcular una ecuación que describa su velocidad. Se han realizado varios ensayos apoyándose en la visión artificial, obteniéndose la Tabla 4-2 que relaciona tiempo de acción de los motores con la distancia en píxeles recorrida, de está tabla se deduce la Figura 4-11, la misma que ha sido obtenida mediante la utilización de la herramienta de ajuste de curvas de Matlab.

Tabla 4-2 Recolección de datos Robot_A.

Consolidación de datos Robot_A							
Tiempo	**Dist. Pixel**	**Tiempo**	**Dist. Pixel**	**Tiempo**	**Dist. Pixel**	**Tiempo**	**Dist. Pixel**
0	0	30	4	60	10	90	18
5	0	35	5	65	12	95	19
10	1	40	6	70	13	100	20
15	1	45	7	75	15		
20	2	50	8	80	15		
25	3	55	9	85	16		

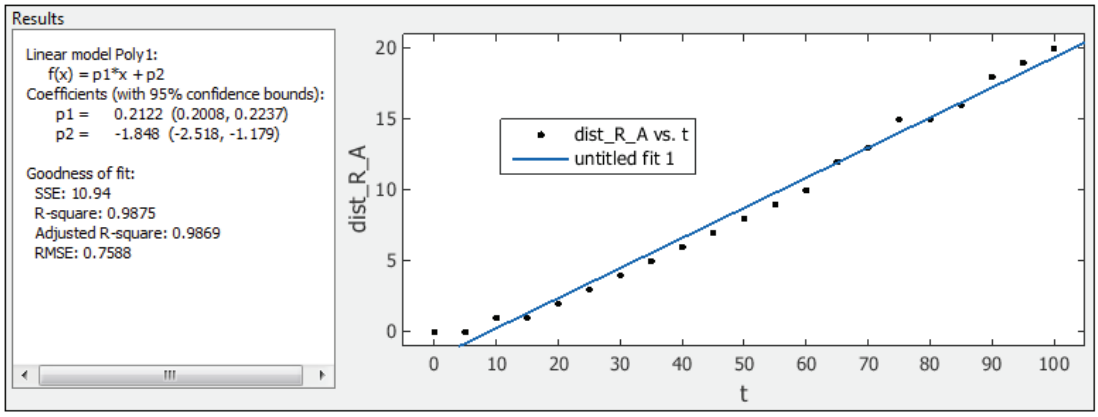


Figura 4-11 Gráfica relación tiempo – distancia Robot_A.

Esta misma experimentación se ha realizado para el Robot_B. obteniendo la Tabla 4-3 y utilizando la misma herramienta, se ha conseguido la Figura 4-12.

Tabla 4-3 Recolección de datos Robot_B.

Consolidación de datos Robot_A							
Tiempo	**Dist. Pixel**	**Tiempo**	**Dist. Pixel**	**Tiempo**	**Dist. Pixel**	**Tiempo**	**Dist. Pixel**
0	0	30	5	60	12	90	18
5	0	35	6	65	13	95	19
10	1	40	7	70	14	100	20
15	1	45	8	75	15		
20	2	50	9	80	17		
25	3	55	11	85	17		

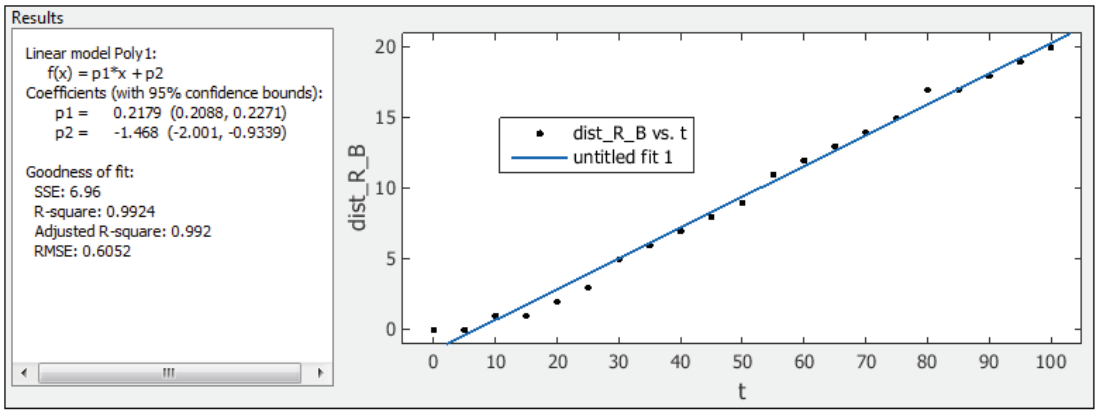


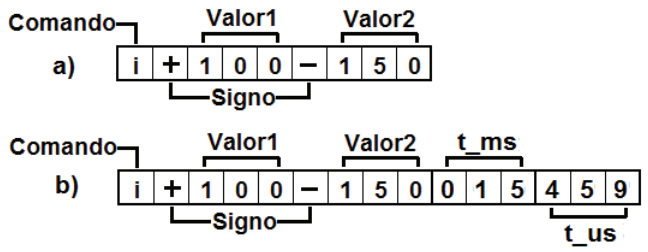
Figura 4-12 Gráfica relación tiempo – distancia Robot_B.

En las Figuras 4-11 y 4-12 se observa que el comportamiento para cada uno de los robots es muy similar debido a que es lineal, la herramienta utilizada permite obtener las ecuaciones de las dispersiones de puntos para cada caso, encontrándose que para cada uno de los robots son muy similares las curvas obtenidas. En cuanto a las pendientes son las mismas, pero varían en el punto de corte con el eje de las ordenadas. Las siguientes ecuaciones son las que relacionan tiempo con distancia, las ecuaciones de la velocidad de los agentes.

$$\boldsymbol{distancia = 0.2122 * tiempo - 1.848} \qquad (4.1)$$

$$\boldsymbol{distancia = 0.2179 * tiempo - 1.468} \qquad (4.2)$$

Con esta ecuación se puede calcular el tiempo necesario para que el robot recorra una distancia dada desde un punto1 hasta un punto2, este dato será enviado al robot vía bluetooth, por consiguiente, deberá modificarse la trama de transmisión ya que originalmente se transmitía únicamente valores de PWM y dirección para el movimiento de los motores, este cambio se aprecia en la Figura 4-13.



a) Trama inicial b) Trama modificada

Figura 4-13 Trama de comunicación enviada a los esclavos.

Con los cambios realizados en el sistema se ha obtenido la Figura 4-14 de la ruta planteada y realizada por el Robot_A. recuerde que se sigue manteniendo la misma ruta virtual de las experimentaciones anteriores.

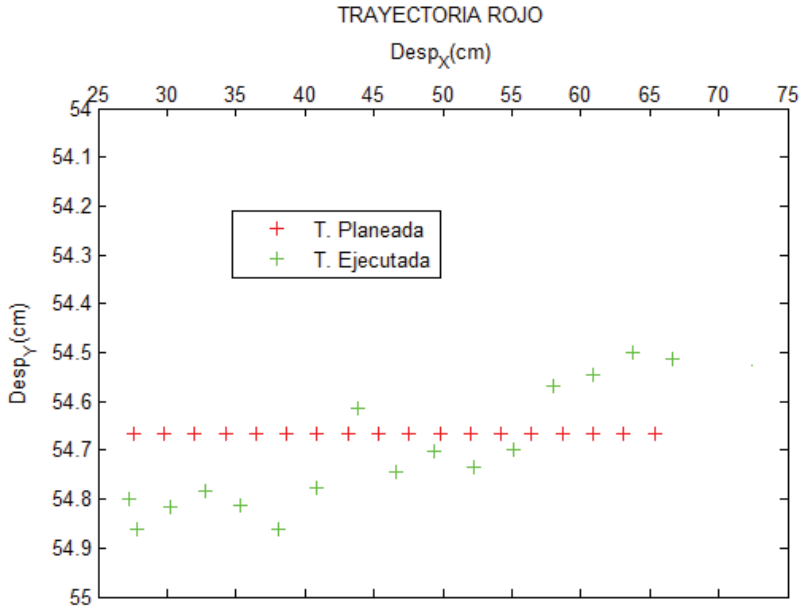


Figura 4-14 Ruta horizontal Robot_A.

En la Figura 4-14 se tiene la gráfica de la ruta propuesta, color rojo, y la ruta realizada, color verde, de la misma que se precia que el error en distancia recorrida,

eje x, ya está minimizada, esto indica que la corrección realizada en la trama para que el tiempo de acción en los motores sea calculado y transmitido desde el maestro ha surtido efecto. Ahora con respecto al eje y, se puede apreciar que el error también es mínimo, por consiguiente, esto ratifica que el cambio realizado a la trama ha surtido un buen efecto, no se llega a cero por características mecánicas del sistema. Con respecto al error de ángulo, el valor sigue sin superar el ±1°, esto se puede verificar en la Figura 4-15.

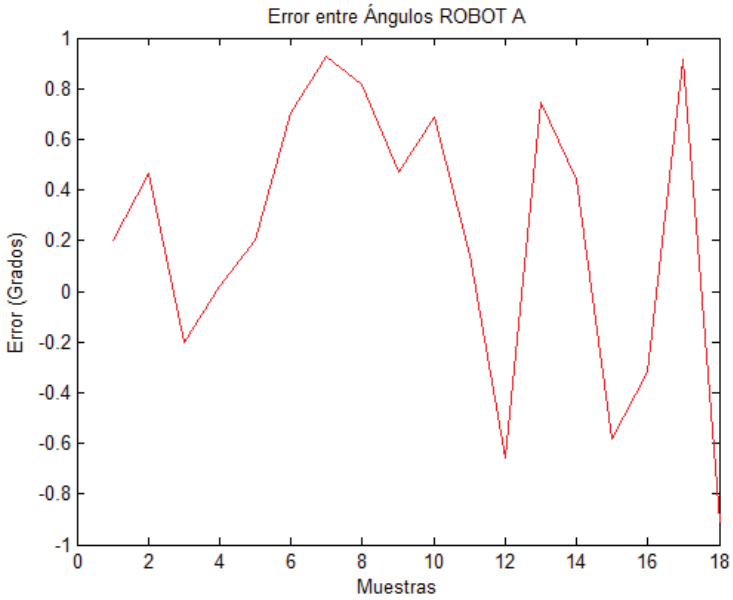


Figura 4-15 Error de ángulo.

Para ratificar el correcto funcionamiento del sistema, se plantean nuevas rutas para analizar el funcionamiento de los robots en su desplazamiento por las mismas, esto permitirá validar los cambios realizados al algoritmo de operación del sistema o realizar nuevos cambios para mejorar su desempeño.

4.2.2. RUTA VERTICAL ROBOT_A.

La ruta establecida es vertical, la misma que irá desde arriba hacia abajo del entorno.

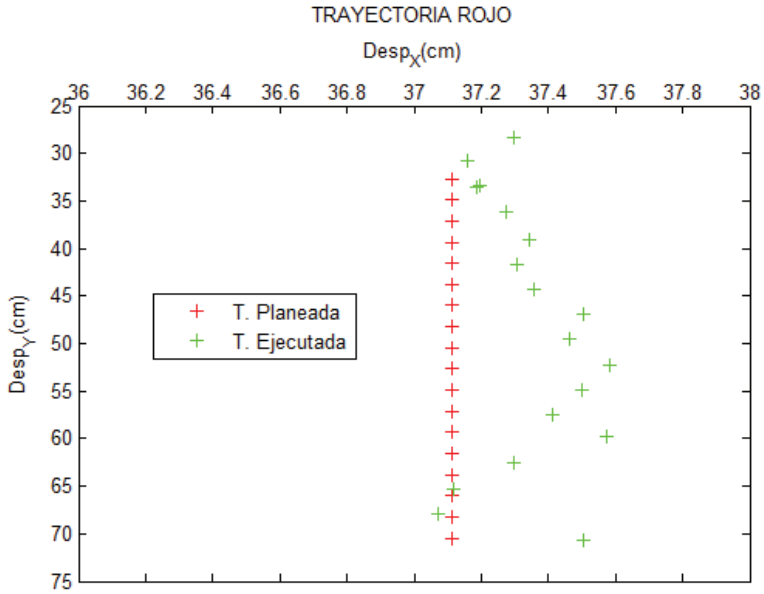


Figura 4-16 Ruta vertical Robot_A.

En la Figura 4-16 se aprecia que el error en la posición sigue manteniéndose en valores pequeños, no supera de 0.4 cm, ahora para el error de ángulo, este se observa en la Figura 4-17 que se mantiene sin superar del ±1°.

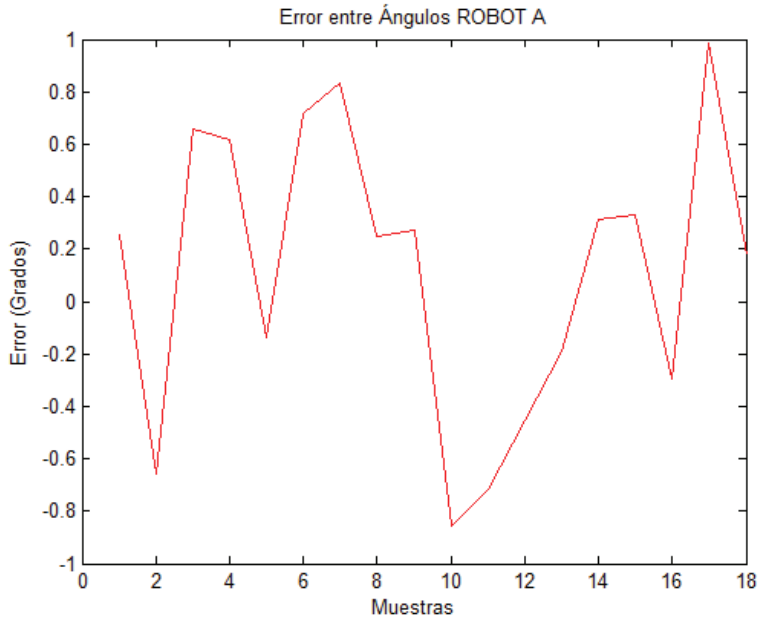


Figura 4-17 Error entre ángulos ruta vertical Robot_A.

4.2.3. RUTAS A MÚLTIPLOS DE 45° ROBOT_A

Para esta prueba se han planteado varias rutas como múltiplos de 45° para la ruta, la primera es a 135° dentro del entorno, esto se aprecia en las Figura 4-18. En esta primera ruta diagonal se puede apreciar que el error de posición es pequeño, Figura 4-19 y el error de ángulo no supera de ±1°, Figura 4-20.

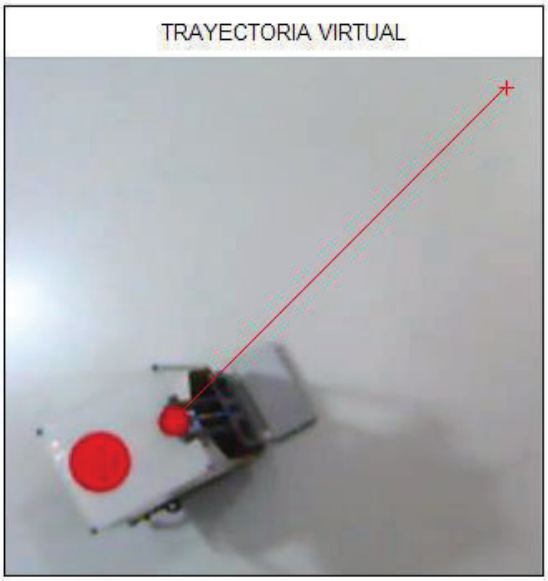


Figura 4-18 Ruta virtual a 135°.

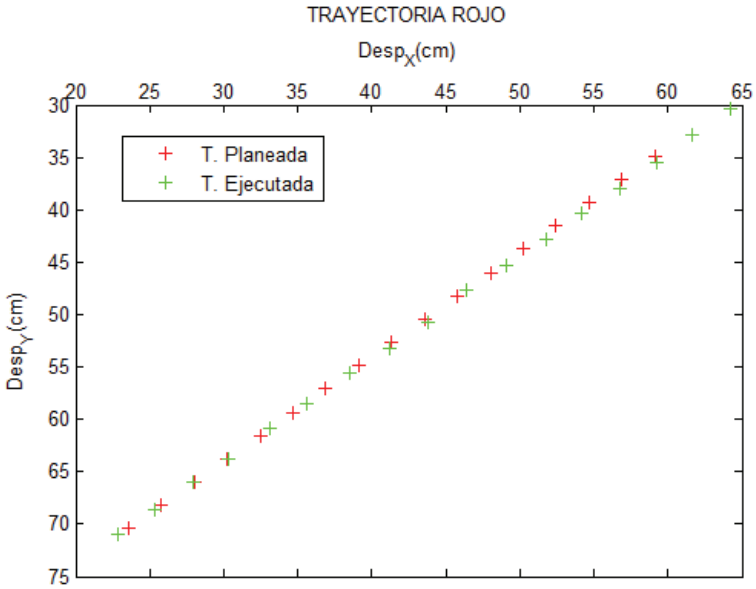


Figura 4-19 Posiciones con ruta a 135°.

Para una mejor apreciación se grafica el error entre los ángulos.

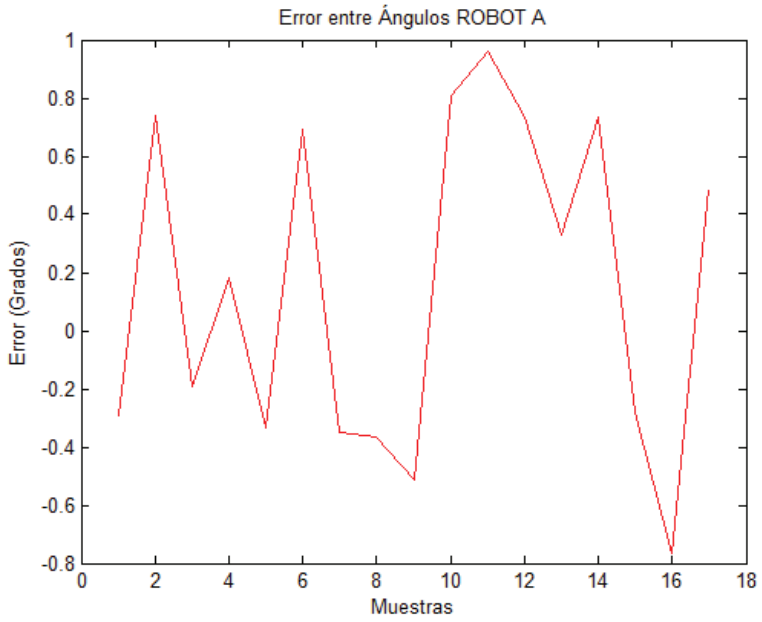


Figura 4-20 Error de ángulo con ruta a 135°.

La otra ruta propuesta es a 45°, Figura 4-21, de esta ruta se puede apreciar que la respuesta es muy similar a la anterior, es decir un pequeño error en la posición, Figura 4-22 y un error entre ángulos generados que no supera del ±1°, Figura 4-23.

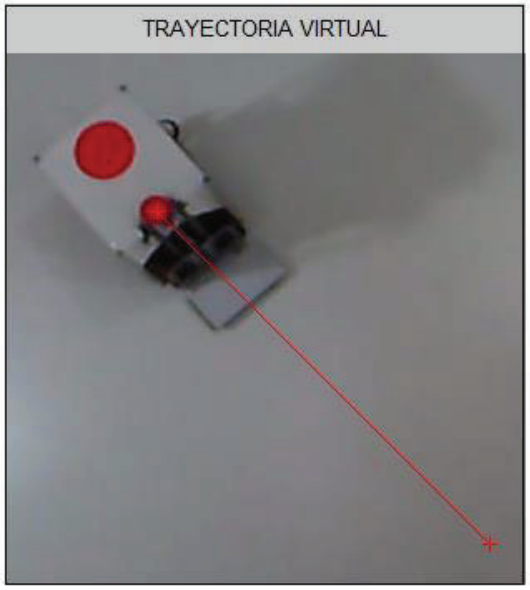


Figura 4-21 Ruta virtual a 45°.

Se grafica el movimiento realizado por el robot.

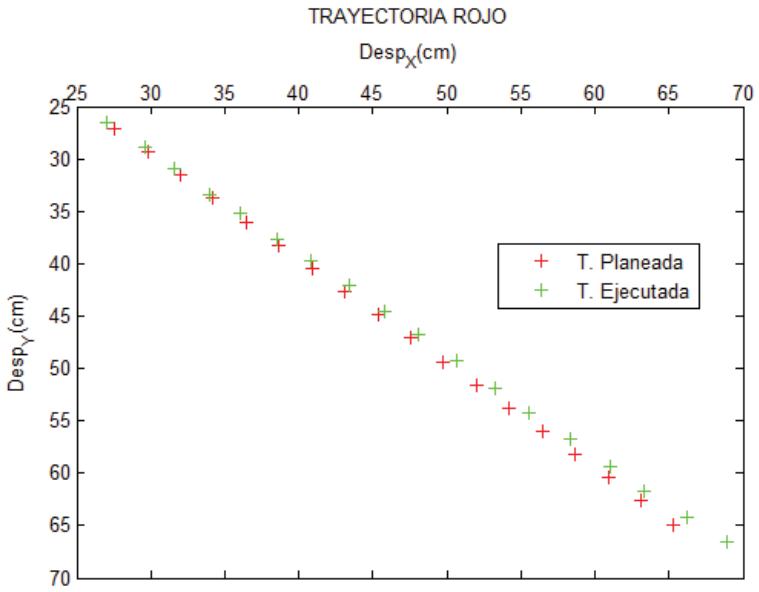


Figura 4-22 Posiciones con ruta a 45°.

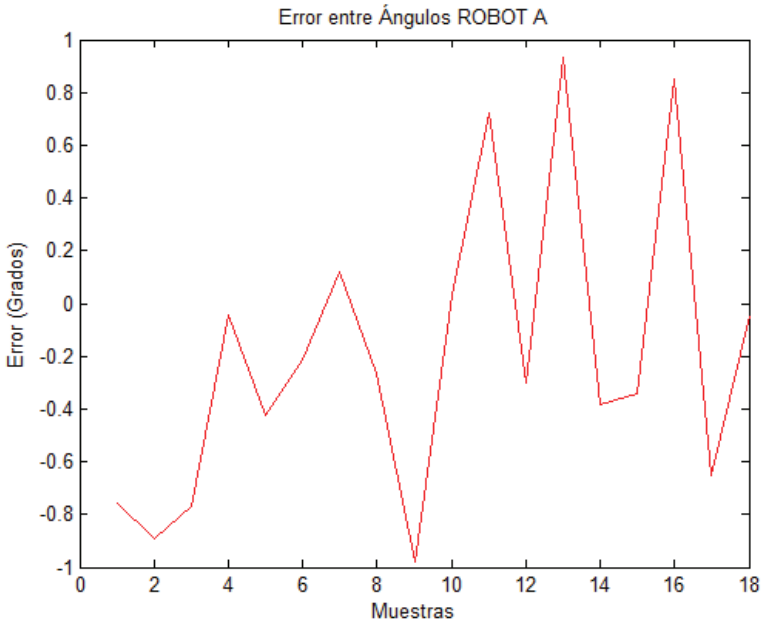


Figura 4-23 Error de ángulo con ruta a 45°.

La siguiente ruta corresponde al valor de -45°, al igual que en las dos diagonales anteriores los resultados son similares, errores en la posición pequeños y un error en los ángulos generados por el controlador que no supera el ±1°.

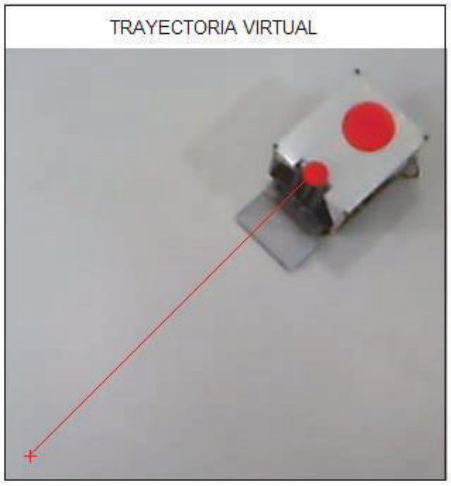
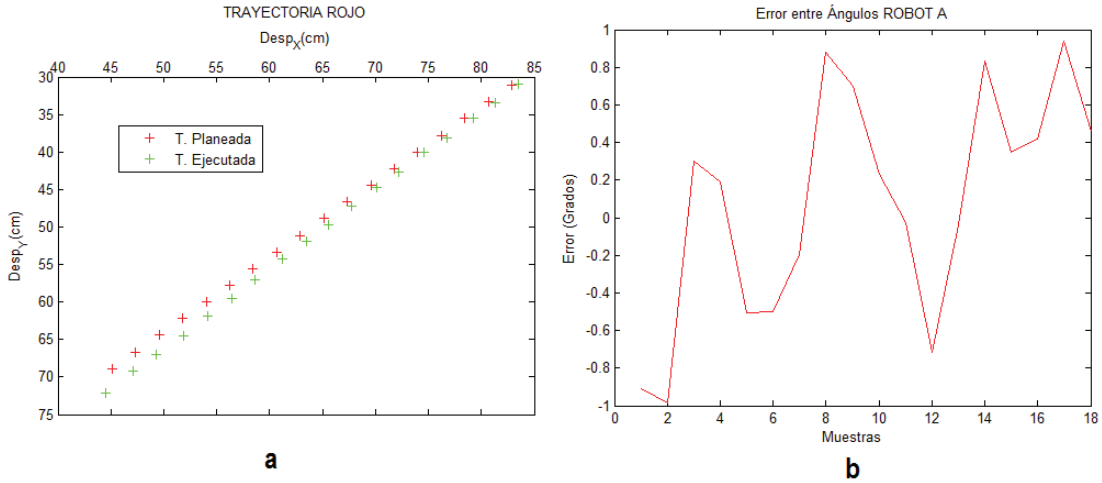


Figura 4-24 Ruta virtual -45°.



a) Gráfico de posición b) Grafico de error de ángulos

Figura 4-25 Posición y error en la ruta -45°.

Se han minimizado el tamaño de los gráficos por optimización del espacio, debido a que las respuestas del sistema son las mismas que en experimentaciones anteriores.

La siguiente ruta corresponde a -135°. Nuevamente los resultados son similares a las experimentaciones anteriores, en lo que respecta a los errores de posición y de ángulos.

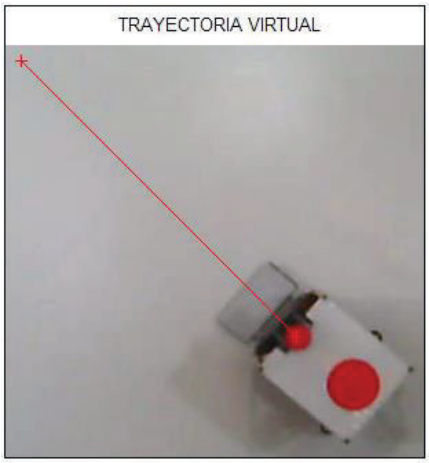
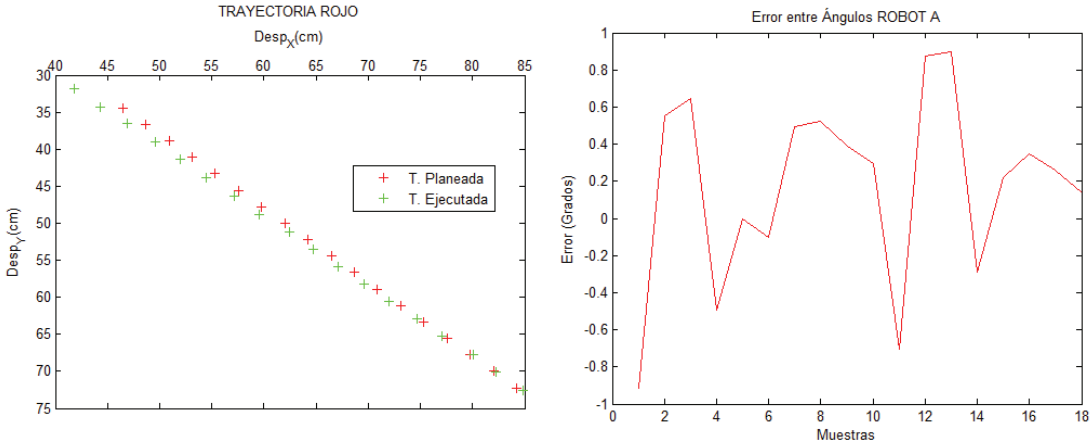


Figura 4-26 Ruta virtual a -135°.



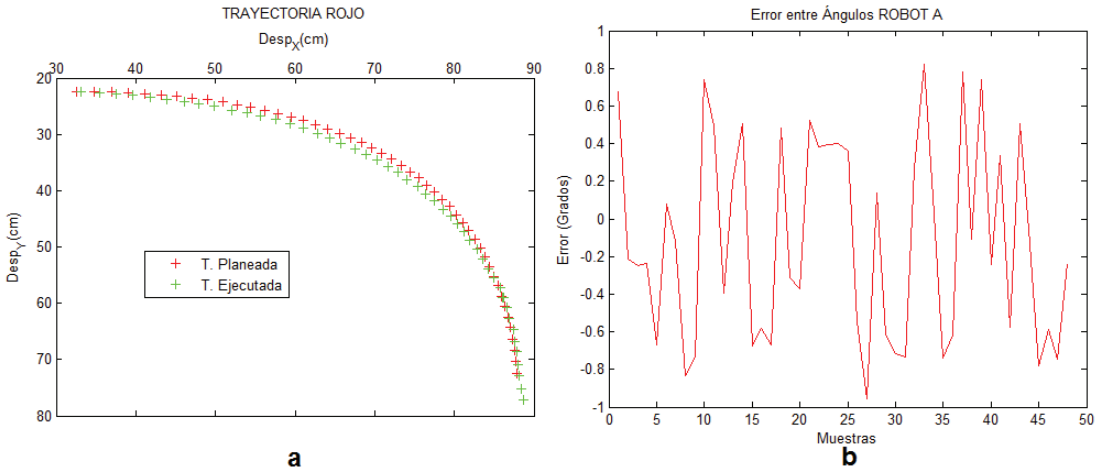
a **b**

a) Gráfico de posición b) Grafico de error de ángulos

Figura 4-27 Posición y error ruta -135°.

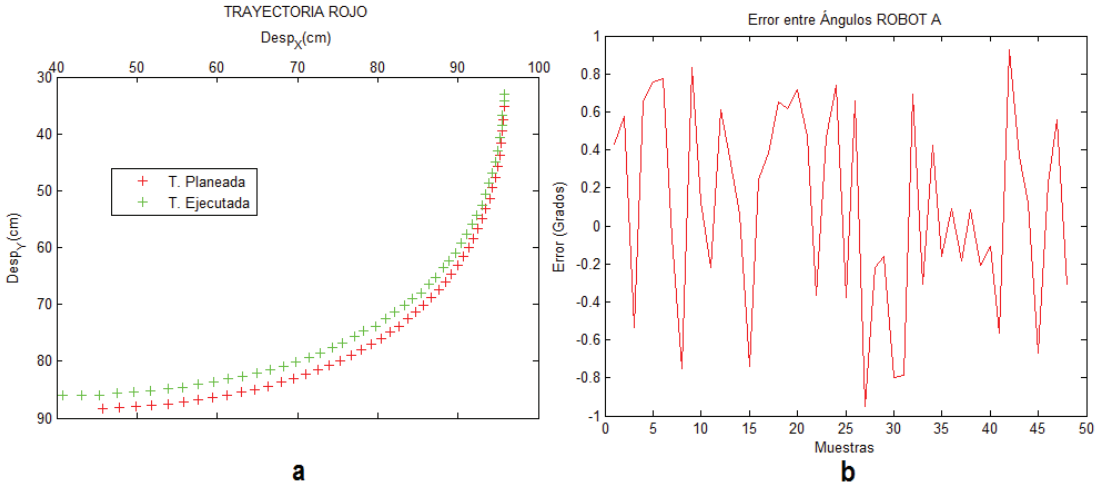
4.2.4. RUTAS CURVAS ROBOT_A.

Estas rutas están formadas por cuadrantes de circunferencia para verificar el comportamiento del robot para este tipo de movimiento:



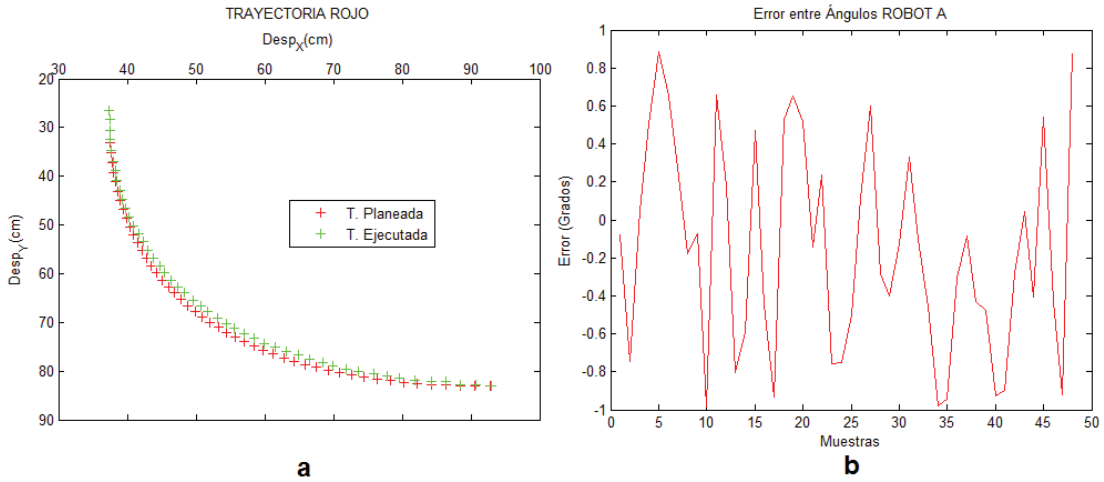
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-28 Movimiento primer cuadrante Robot_A.



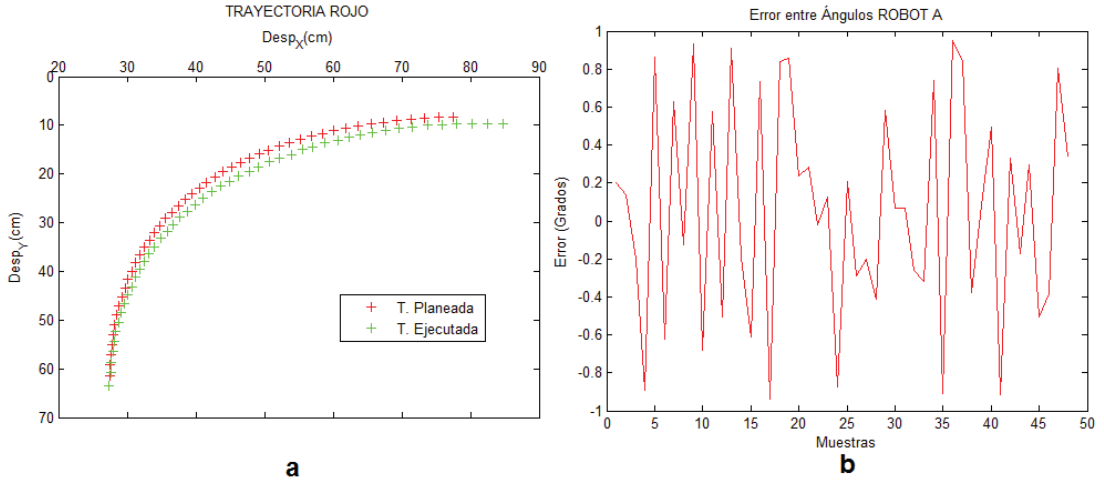
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-29 Movimiento segundo cuadrante Robot_A.



a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-30 Movimiento tercer cuadrante Robot_A.



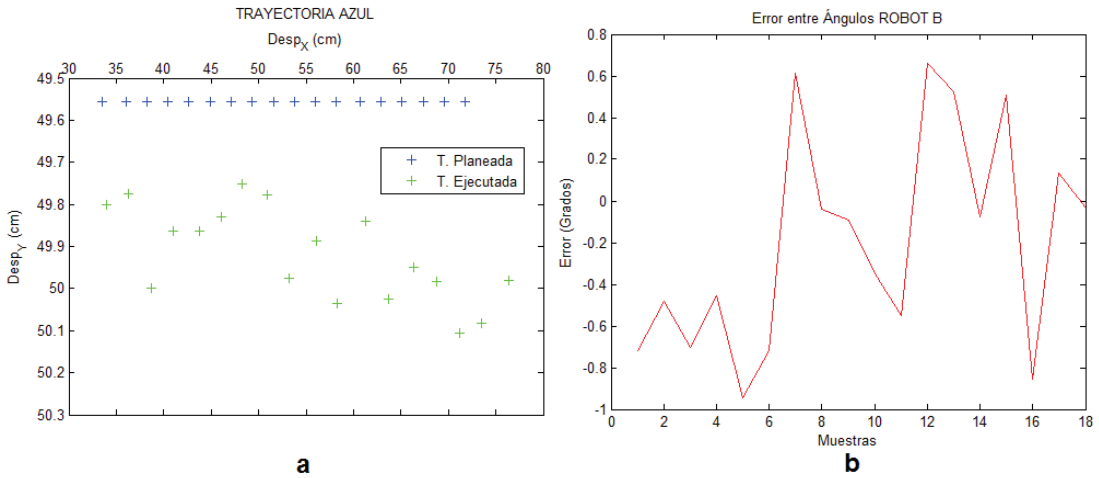
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-31 Movimiento cuarto cuadrante Robot_A.

En las Figuras 4-28, 4-29, 4-30 y 4-31, se tiene los movimientos en cada uno de los cuatro cuadrantes de una circunferencia para el Robot_A, en cada una de ellas se puede apreciar que al momento del inicio del movimiento el robot sigue la ruta con un error de posición mínimo, pero conforme va avanzando este se va alejando. A lo largo de toda la experimentación se ha observado que los valores transmitidos desde el maestro, siempre son los mismos esto pone a pensar que existe algún error adicional pero ya se puede deslindar que es debido a la programación.

Como el sistema se encuentra formado por dos robots, se van a realizar las mismas pruebas con el Robot_B, robot con marcas de guía azul.

4.2.5. RUTA HORIZONTAL ROBOT_B.



a) Rutas propuesta y seguida b) Error entre ángulos

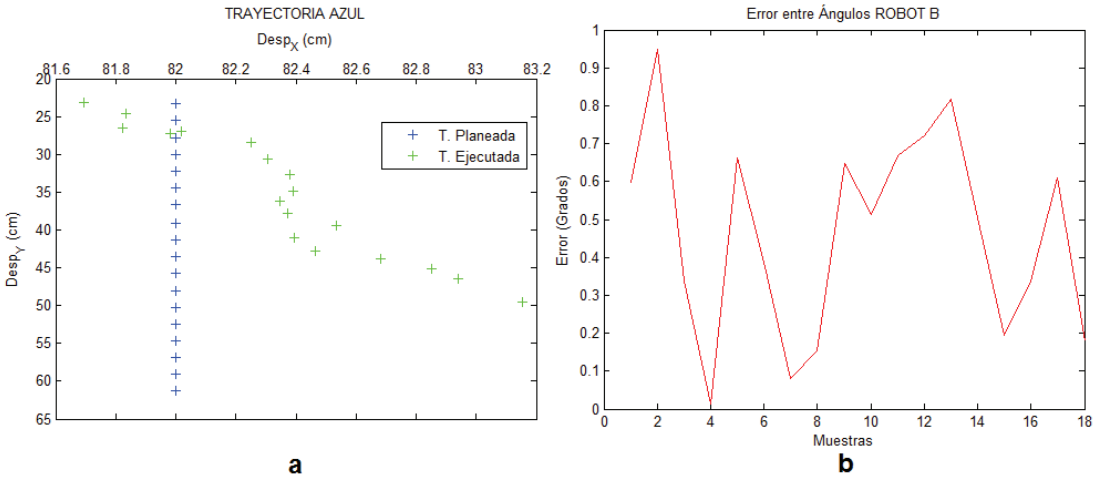
Figura 4-32 Ruta horizontal Robot_B.

En la Figura 4-32a, se aprecia que el avance es correcto, persiste el error de posición, pero se mantiene en valores pequeños, obsérvese la diferencia entre el la marca más lejana y la ruta planificada, no supera los 0.5 cm, distancia muy aceptable. En la Figura 4-32b se puede ver que el controlador está calculando muy bien las acciones de control, debido a que no existe error entre los ángulos mayor de ±1%.

4.2.6. RUTA VERTICAL ROBOT_B.

En la Figura 4-33a se aprecia la ruta realizada de forma vertical por el Robot_B, nótese de la misma forma que la ruta horizontal error de posición no supera de 1.2 cm, además se confirma que para este tipo de movimiento el controlador genera las acciones de control para que el error entre los ángulos no sea mayor del ±1°.

En las experimentaciones posteriores se seguirá analizando los problemas relacionados con la posición, debido a que con respecto a los ángulos las respuestas del sistema de control se siguen manteniendo.

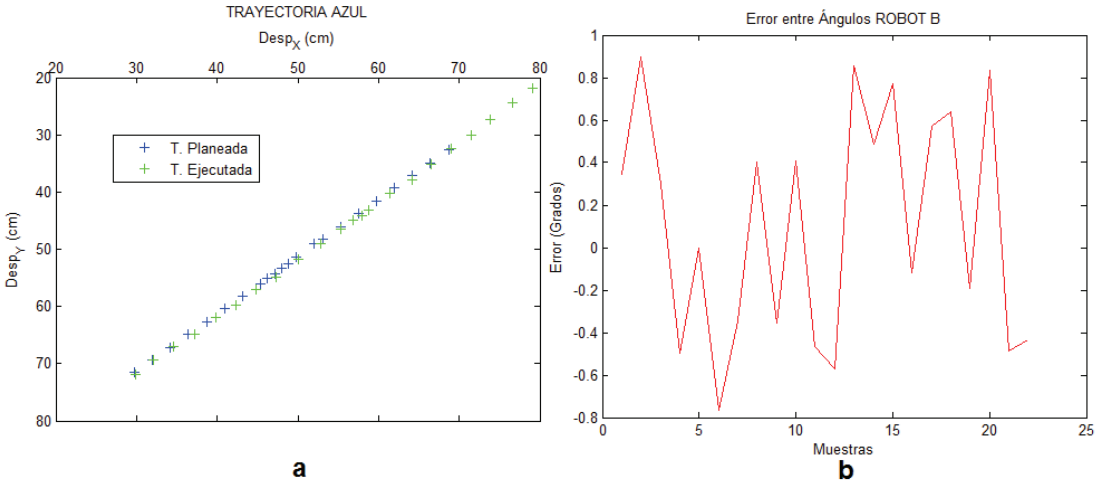


a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-33 Ruta vertical del Robot_B.

4.2.7. RUTAS A MÚLTIPLOS DE 45° DEL ROBOT_B.

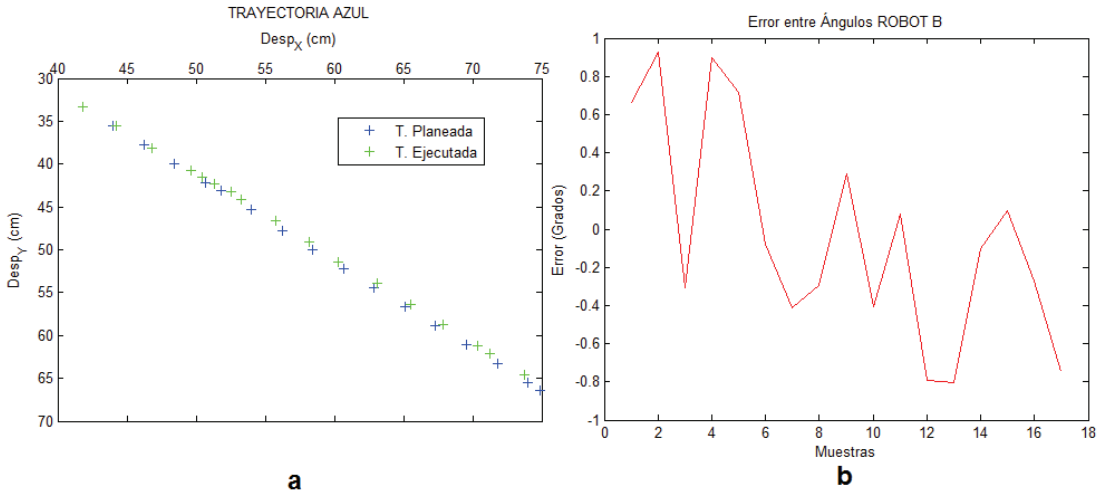
La primera prueba es a 135°, de esta experimentación se han obtenido los gráficos en las Figura 4-34.



a) Rutas propuesta y seguida b) Error entre ángulos

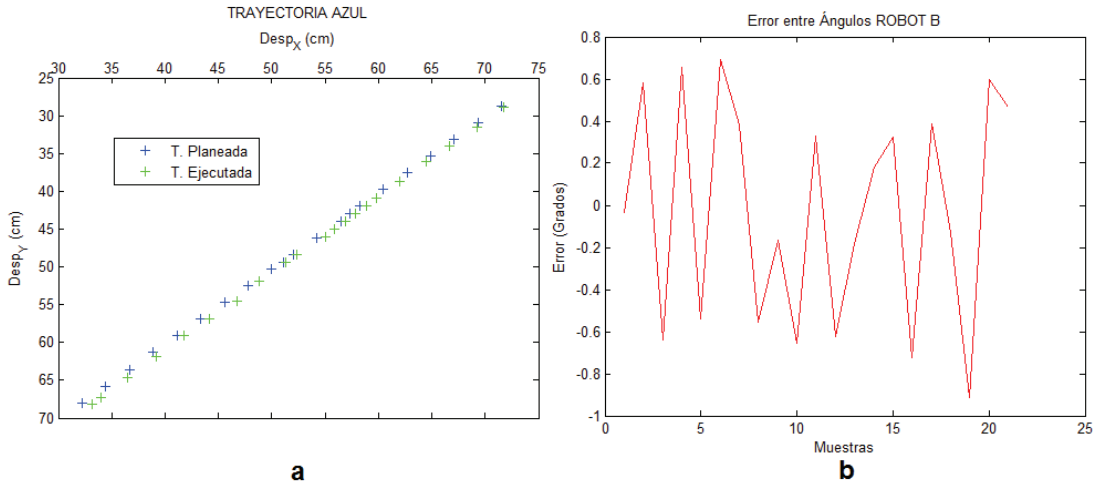
Figura 4-34 Ruta 135° Robot_B.

En la Figura 4-34a, se aprecia que la ruta seguida va a la par de la ruta planificada, se aprecian errores en la posición que son pequeños. En lo que respecta al error generado por los ángulos se tiene que los valores no superan de ± 1°, esto da a comprender que el controlador diseñado para este robot cumple con las expectativas de control esperadas. Ahora se plantea una ruta a 45°, para esto se obtiene la Figura 4-35.



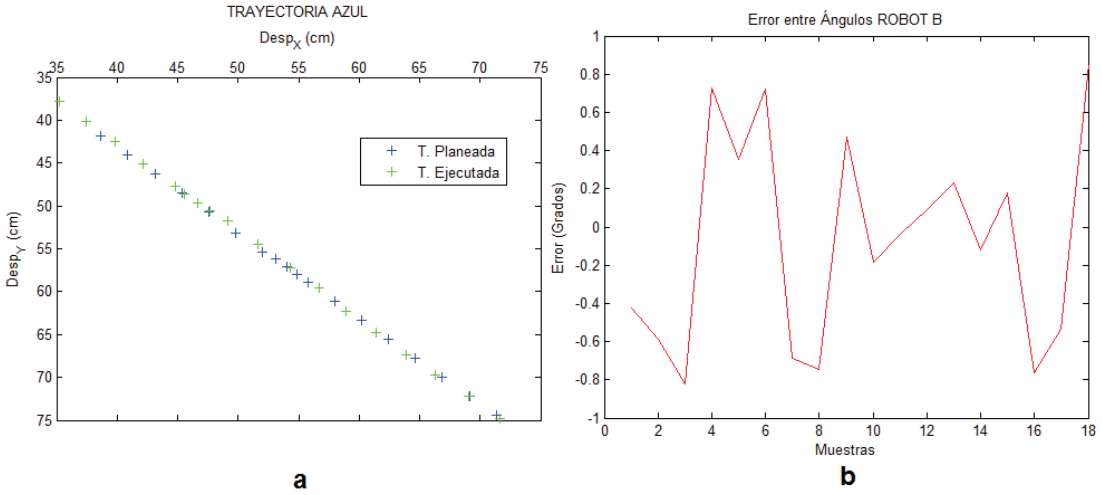
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-35 Ruta 45° Robot_B.



a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-36 Ruta a -45° Robot_B.



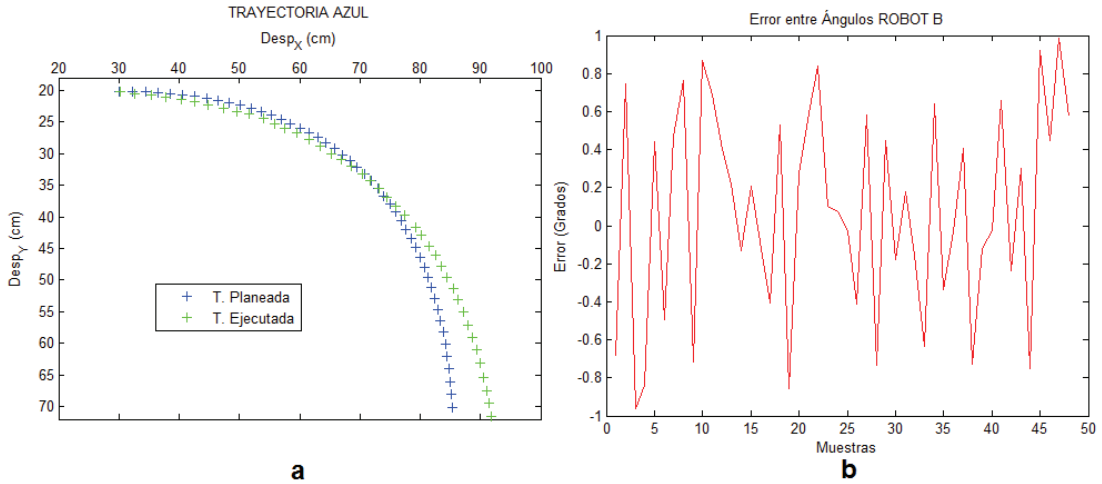
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-37 Ruta -135° Robot_B.

En las Figuras 4-34, 4-35, 4-36, 4-37 se observan resultados similares a los obtenidos por el Robot_A, errores pequeños de posición y además errores pequeños entre ángulos.

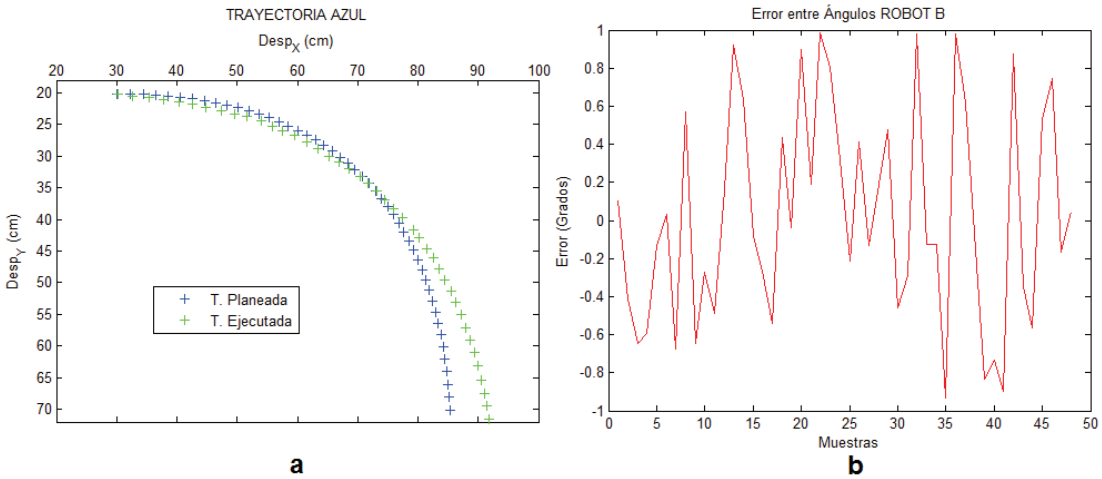
4.2.8. RUTAS CURVAS ROBOT_B.

El espacio ha sido dividido en cuatro cuadrantes para cubrir todas las posibilidades de movimiento que podrían realizar cada uno de los robots.



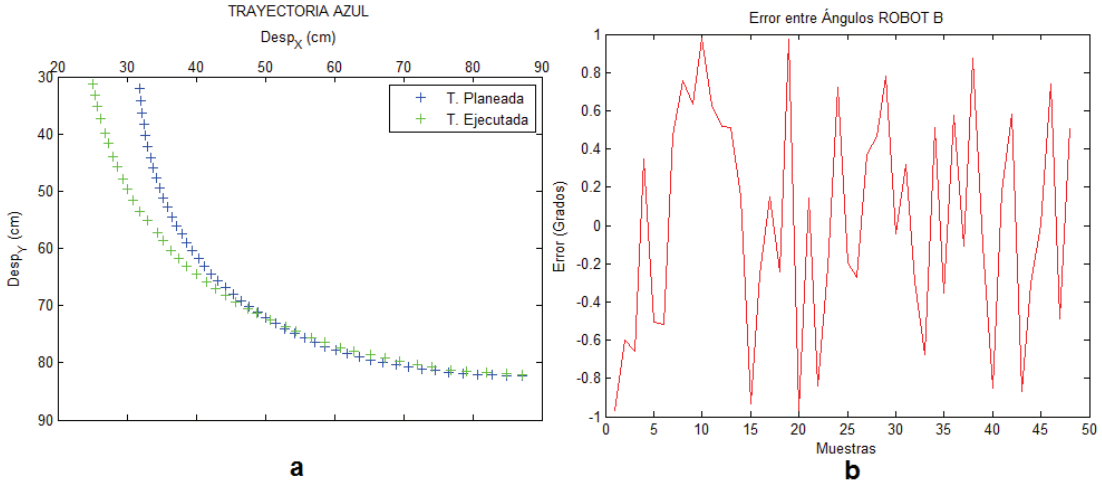
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-38 Movimiento primer cuadrante Robot_B.



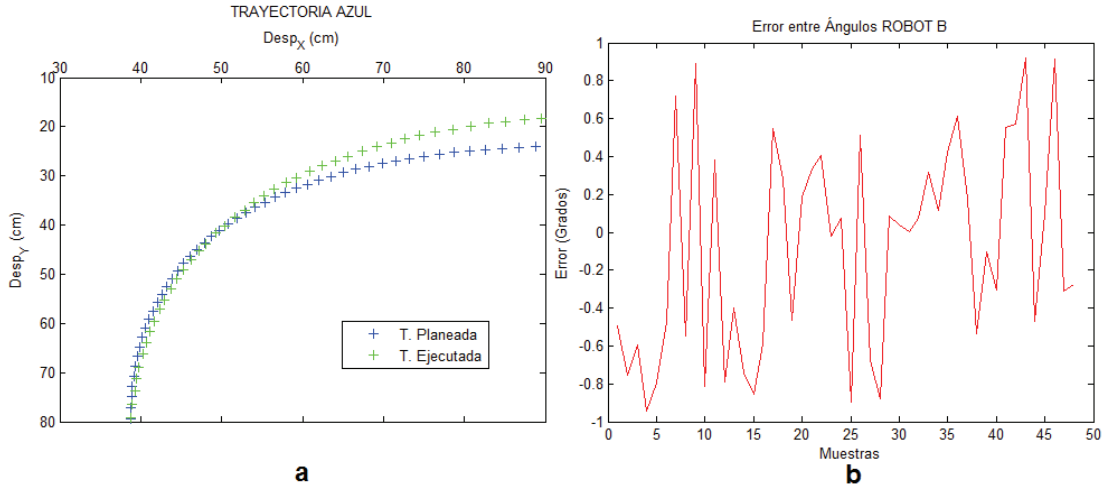
a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-39 Movimiento segundo cuadrante Robot_B.



a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-40 Movimiento tercer cuadrante Robot_B.



a) Rutas propuesta y seguida b) Error entre ángulos

Figura 4-41 Movimiento cuarto cuadrante Robot_B.

En las Figuras 4-38, 4-39, 4-40 y 4-41, se tiene los movimientos del Robot_B en los cuatro cuadrantes del entorno, se precia que al inicio del movimiento es seguido muy bien por el robot, pero conforme va avanzando se va alejando de la misma, el mismo comportamiento se pudo apreciar en el movimiento del Robot_A.

4.3. PRUEBA 02: FUENTE.

En vista que el error de posición no ha podido der reducido en mayor grado, se ha analizado el funcionamiento de los robots, en donde: los elementos encargados del movimiento son los motores a los cuales se les aplica una señal modulada en ancho de pulso directo de la fuente, es decir sin ningún sistema de regulación. Esta señal por definición es de tipo cuadrada de frecuencia fija y que varía su tiempo en alto para modificar su valor DC.

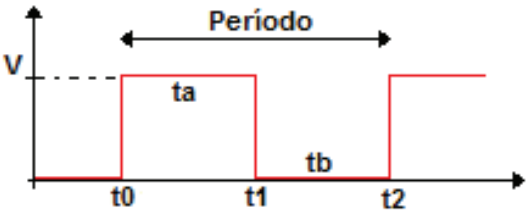


Figura 4-42 Señal PWM.

Pero los valores aplicados a los motores son de corriente continua, razón por la que aplicando la ecuación del voltaje DC, de tiene:

$$v_{dc=}\frac{1}{T}\int_{a}^{b} f(t)dt \qquad (4.3)$$

Aplicado a la señal PWM se tiene:

$$v_{dc=}\frac{1}{T}\int_{t0}^{t2} vdt \qquad (4.4)$$

Resolviendo y aplicando simplificación se tiene.

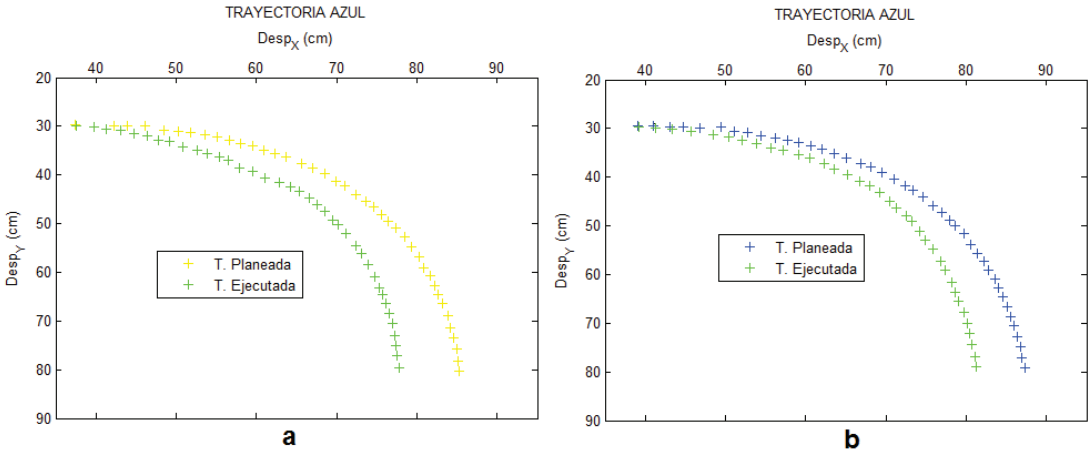
$$v_{dc=}Kv \qquad (4.5)$$

En donde K corresponde al ciclo de trabajo, por lo tanto, los valores de voltaje DC aplicados a los motores estarán en función de la fuente.

Para comprobar esto se ha realizado una prueba de seguimiento de ruta con dos condiciones: la primera cuando la batería que alimenta el sistema de potencia se encuentra recién cargado y la segunda cuando la batería se encuentra cerca de su límite inferior.

Para la alimentación del sistema de potencia se ha utilizado una batería de Lipo de dos celdas la misma que tiene como voltaje máximo 8.43 V_{DC} y un voltaje mínimo de 6 V_{DC}. Por lo tanto, a lo largo del tiempo de operación de los robots los voltajes aplicados a los motores irán disminuyendo, por consiguiente, disminuyendo su desplazamiento a lo largo de la ruta.

Para validar esto se obtuvieron las gráficas de la Figura 4-43.



a) Ruta con batería a 6 V_{DC} b) Ruta con batería a 8.43 V_{DC}.

Figura 4-43 Ruta con diferentes valores de fuente.

En la Figura 4-43b se puede apreciar que el desplazamiento es mayor con la batería totalmente cargada, con respecto a la operación con la batería con desgaste.

Un caso particular detectado durante esta prueba son los falsos positivos, esto se refiere a posibles puntos que pueden ser identificados por el sistema de visión artificial como marcas de orientación. Esto se dio debido a que para el Robot_B el color filtrado azul y los agentes en la parte superior de la horquilla tiene dos puntos de sujeción del eje con soportes de color azul, los mismos que durante la operación fueron detectados como marca de orientación pequeña, círculo pequeño.

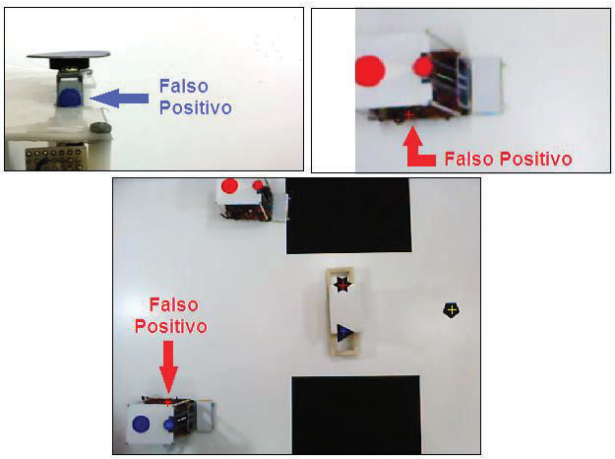


Figura 4-44 Falso positivo color azul.

Para solucionar este problema se realizaron las siguientes acciones: primero se le realizó una cubierta de color blanco para el sistema electrónico, de esta forma se pudo aislar posibles marcas. La segunda acción fue realizar la eliminación de áreas en base de dos límites, esto se refiere a que originalmente se eliminaban áreas mayores a la de la marca pequeña, de allí la razón de la prencia de los falsos positivos, marcas más pequeñas, de esta manera en el entono únicamente quedó la marca pequeña de orientación de Robot_B. Forma final de los robots se la puede apreciar en la Figura 4-45.

Figura 4-45 Robot con cubierta.

4.4. PRUEBA 03: MOVIMIENTO COOPERATIVO 1.

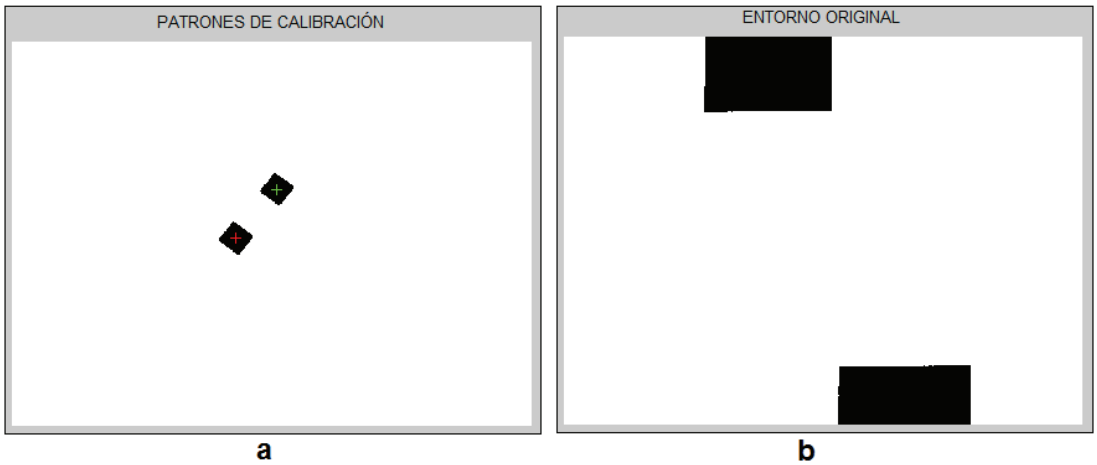
Una vez que se ha analizado la operación de los robots de forma libre con rutas virtuales, ahora se somete a los robots al entorno semi estructurado, en el cual las rutas ya son generadas en base del análisis de las imágenes y la aplicación de algoritmos de visión artificial.

Como se menciona en el capítulo 3, el sistema va cumpliendo varias fases para su operación, razón por la que se tienen acciones que son comunes para todas las pruebas. Estas acciones comunes se mencionan en esta prueba, pero para las siguientes por ser las mismas no se las mencionará debido a que entregarán resultados similares.

4.4.1. FASES COMUNES.

La fase de seteo determina la creación del objeto para la cámara para la adquisición de las imágenes, así como también la creación de los objetos para la comunicación bluetooth con cada uno de los robots.

Otra fase común es la de patrones, esta determina en el cálculo de una variable llamada lado, la misma que es utilizada posteriormente para la compresión de la imagen del entorno y demás cálculos, esto se realiza para que la altura a la cual se encuentra la cámara no sea un valor fijo. De esta fase se obtiene la Figura 4-46a.



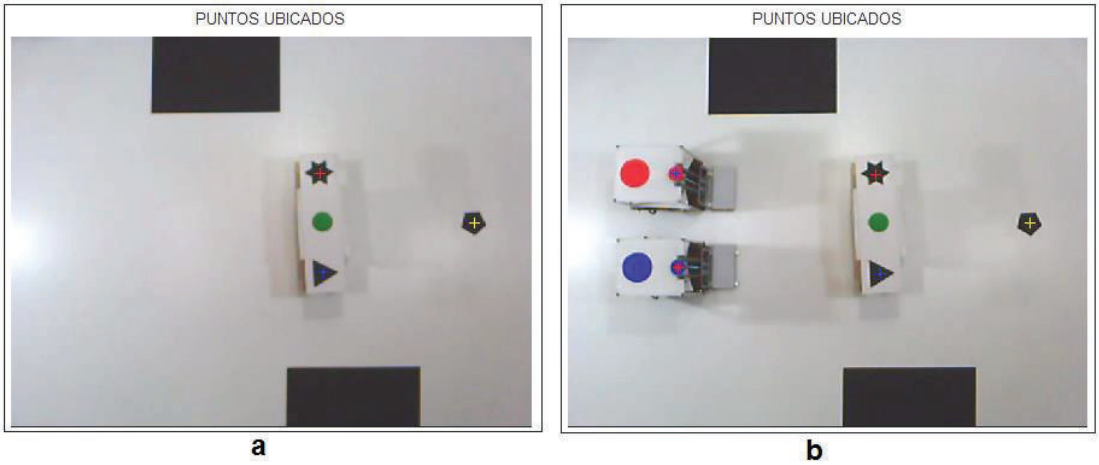
a) Patrones de calibración b) Entorno Original

Figura 4-46 Fases Comunes.

En la Figura 4-46b se tiene una toma del entorno, la misma que será binarizada y sirve como base de los cálculos posteriores del entorno. Hasta aquí las fases que son comunes para cada una de las pruebas.

4.4.2. OBTENCIÓN DE PUNTOS.

Para la ejecución del algoritmo es necesaria la ubicación de puntos base y con estos el sistema calcula las rutas y demás acciones que ejecutarán los agentes para cumplir con su función. Inicialmente se ubica el punto final en donde será transportado el objeto, para el caso se lo identifica como el pentágono, Figura 4-47a, así como también los puntos de finales de movimiento para cada uno de los robots, estos se identifican con la estrella de 6 puntas para el Robot_A y con el triángulo para el Robot_B.



a) Marcas de puntos finales de movimiento. b) Puntos iniciales de movimiento en los robots.

Figura 4-47 Puntos de Movimiento prueba 03.

Una vez identificados los puntos finales del movimiento se tienen que ubicar los puntos iniciales para los movimientos, estos se encuentran sobre cada uno de los robots con forma de círculos pequeños de dos colores, rojo para el Robot_A y azul para el Robot_B, esto se aprecia en la Figura 4-47b.

Con toda la información recopilada se tiene que reducir el entorno, para lo cual se aplica el algoritmo de compresión por áreas explicado en el capítulo 3.

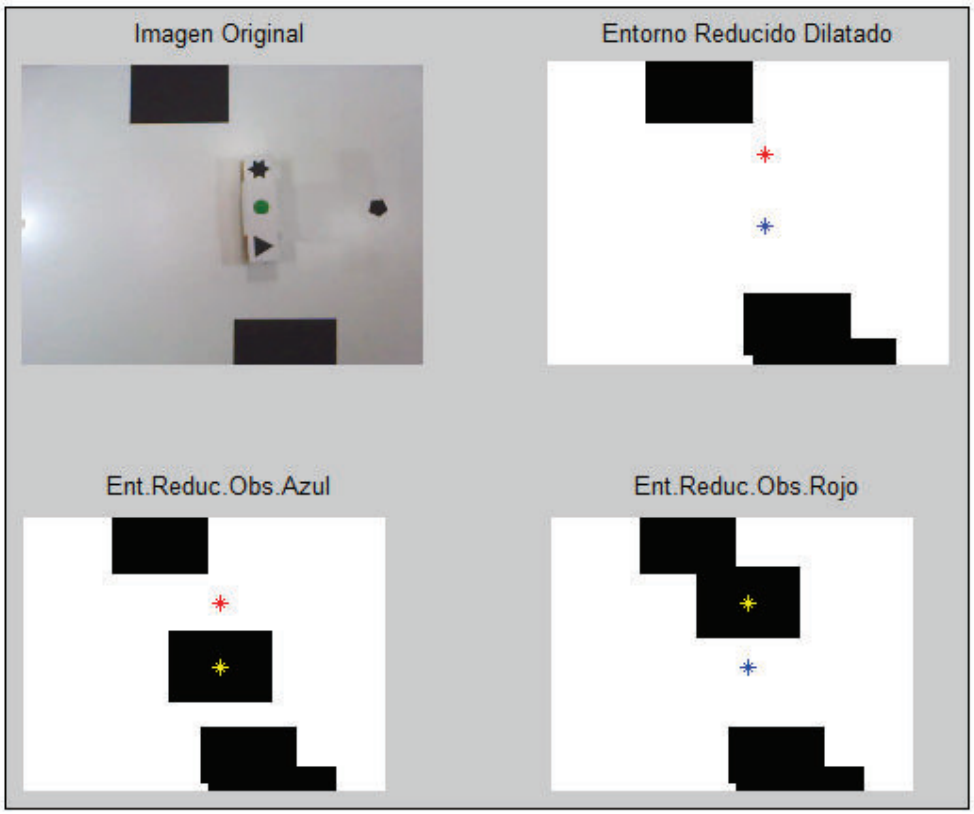
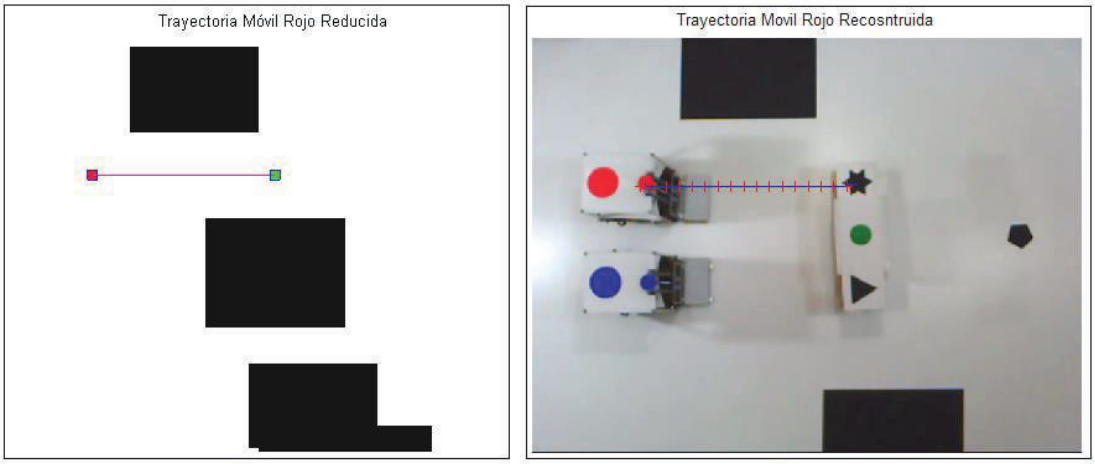


Figura 4-48 Compresión del Entorno prueba 03.

Una vez que se ha comprimido el entorno, se tienen que ubicar sobre él los puntos para el movimiento. Hay que analizar el caso particular de los puntos finales, estos son únicamente puntos dentro del entorno, pero para el caso del análisis serían áreas en las cuales estarán ubicados cada uno de los robots, por lo tanto, se tienen que crear áreas que representen la presencia de estos, lo que se puede apreciar en las Figura 4-48.

Además, se tiene que dilatar la imagen para que exista holgura entre la ruta para el movimiento y el entorno, es decir crear un margen de espacio para que la herramienta del robot, horquilla, no choque con elementos dentro del entorno.

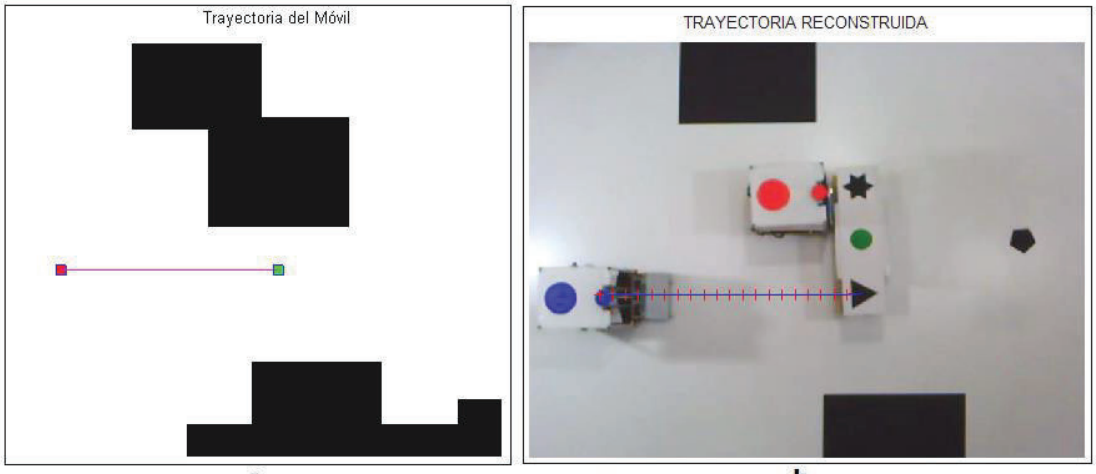
Con lo hasta aquí realizado se buscan las rutas obteniéndose la Figura 4-49 para el Robot_A y la Figura 4-50 para el Robot_B.



a b

a) Ruta encontrada b) Ruta reconstruida

Figura 4-49 Ruta Robot_A prueba 03.



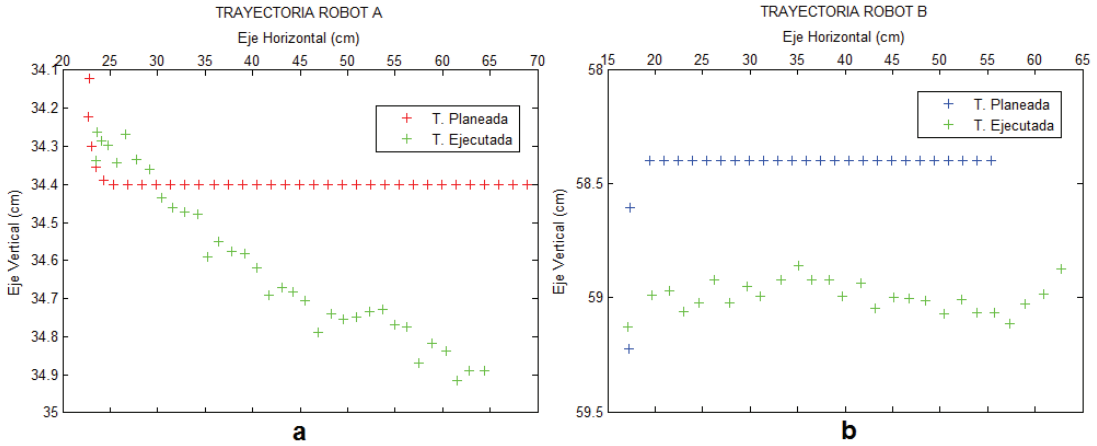
a b

a) Ruta encontrada b) Ruta reconstruida

Figura 4-50 Ruta Robot_B prueba 03.

4.4.3. DESPLAZAMIENTO DE LOS ROBOTS INDIVIDUALES.

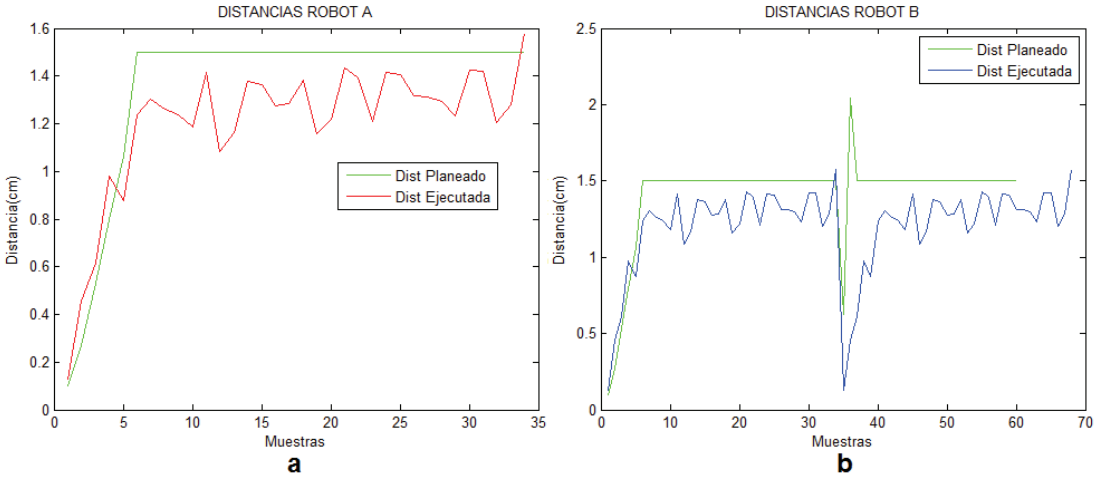
Con las rutas calculadas y ejecutado el algoritmo de movimiento individual para este robot, se obtienen las gráficas de la Figura 4-51 en donde se aprecia el movimiento realizado por el Robot_A.



a) Ruta Robot_A b) Ruta Robot_B

Figura 4-51 Movimientos independientes de robots prueba 03.

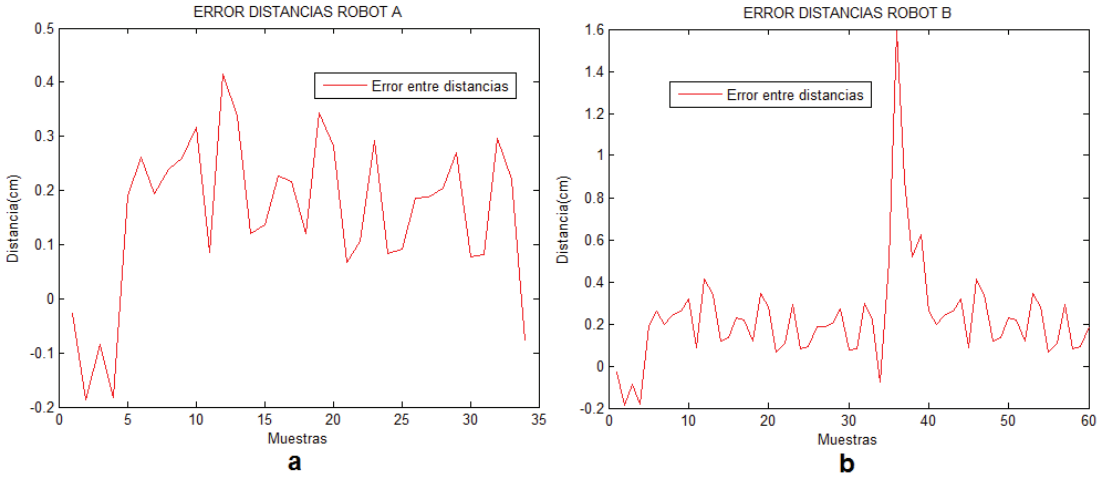
En la Figura 4-51 del desplazamiento de los robots se aprecia que con respecto al eje x los dos agentes llegan a la posición, en cambio con respecto al eje y, tiene el error esperado y detectado en pruebas anteriores. Realizando un análisis con respecto a las distancias se obtiene las siguientes gráficas mostradas en la Figura 4-52.



a) Distancias Robot_A b) Distancias Robot_B

Figura 4-52 Distancias generadas por robots individuales prueba 03.

Para una mejor apreciación del movimiento de los robots se han calculado las distancias absolutas entre los puntos iniciales y los puntos finales de cada movimiento, de esta manera en la Figura 4-52a se muestran las distancias que deben ser ejecutadas por el Robot_A y las generadas por el mismo agente, en donde se aprecia que las distancias se encuentran muy próximas, el mismo análisis se puede realizar observando la Figura 4-52b. Para tener una mejor apreciación del error de las distancias se tiene la Figura 4-53.



a) Error de distancia Robot_A solo b) Error de distancia del Robot_B solo.

Figura 4-53 Errores de distancia de los robots individuales prueba 03.

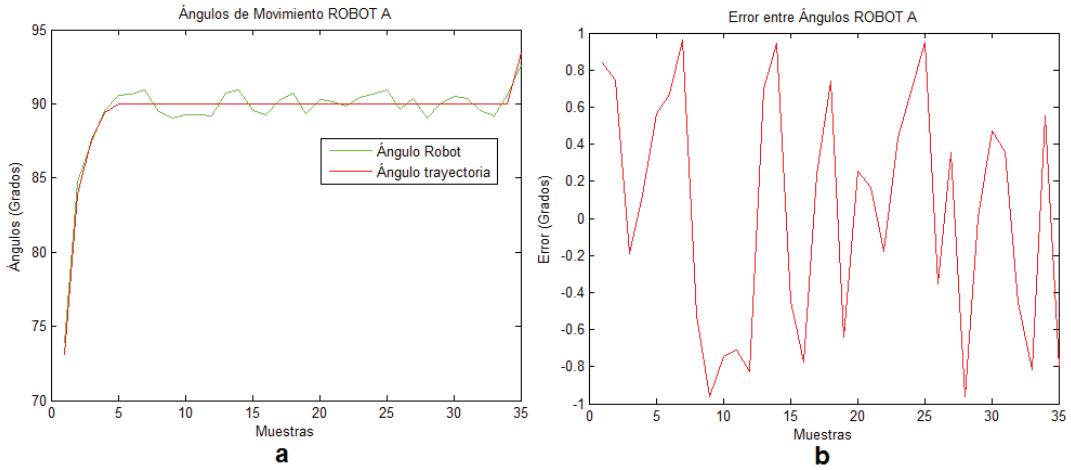
En las Figuras 4-53a y 4-53b se detallan los errores en las distancias, de tal forma que los errores de las distancias son muy pequeños, el detalle de estos se muestra en la Tabla 4-4.

Tabla 4-4 Resumen de errores de distancia robots solos prueba 03.

Robot_A Solo		**Robot_B Solo**	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	0.40	Error máximo	1.6
Error mínimo	0.1	Error mínimo	0.2

En la Tabla 4-4 se puede apreciar que los valores de errores medidos en centímetros son pequeños esto permite entender que el sistema se encuentra siguiendo bien la ruta planificada por el maestro y que las acciones de control realizadas han minimizado los errores de distancia absoluta entre los puntos de

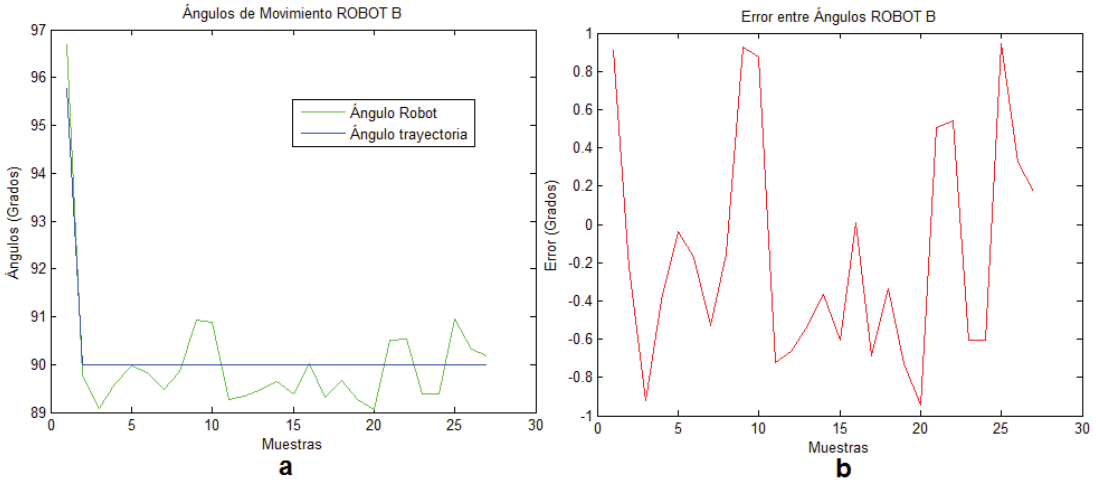
movimiento. Hay de la figura 4-53 hay que entender que si el error es positivo significa que el robot le ha faltado para llegar al punto esperado y caso contrario si es negativo, significa que se ha pasado del punto, este razonamiento queda establecido para todos los posteriores análisis de errores en la distancia. Con relación a los ángulos generados para el movimiento se tiene la Figura 4-54 y la Figura 4-55.



a) Ángulos generados Robot_A b) Error entre ángulos del Robot_A

Figura 4-54 Ángulos del Robot_A, prueba 03.

En la figura 4-54a, se aprecia que los ángulos propuestos son muy próximos a los ángulos realizados por el Robot_A, esto indica que el controlador tiene un muy buen desempeño y para ratificar esto en la Figura 4-54b se observa que el error generado entre los ángulos no supera el ±1°.



a) Ángulos generados Robot_B b) Error entre ángulos del Robot_B

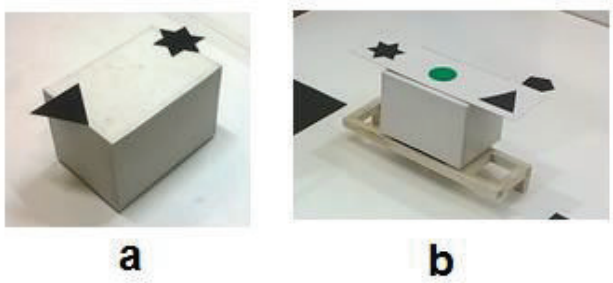
Figura 4-55 Ángulos del Robot_B, prueba 03.

En la figura 4-55a, se aprecia que los ángulos propuestos son muy próximos a los ángulos realizados por el Robot_B, esto indica que el controlador tiene un muy buen desempeño y para ratificar esto en la Figura 4-55b se observa que el error generado entre los ángulos no supera del ±1°.

4.4.4. DESPLAZAMIENTO COOPERATIVO.

Para verificar el movimiento del objeto se han implementado varios cambios: el primero es una separación mayor entre las marcas de orientación esto es debido a que por el ancho de los robots y la presencia de su herramienta genera colisiones entre los agentes al momento de posicionarse frente al objeto.

El segundo cambio es una marca adicional de color verde sobre el mismo para ubicarlo dentro del entorno, se ha utilizado este color ya que es el restante de la gama de información de la imagen, hay que recordar que se está trabajando con imágenes RGB y resulta sencillo el realizar un filtrado de color verde.



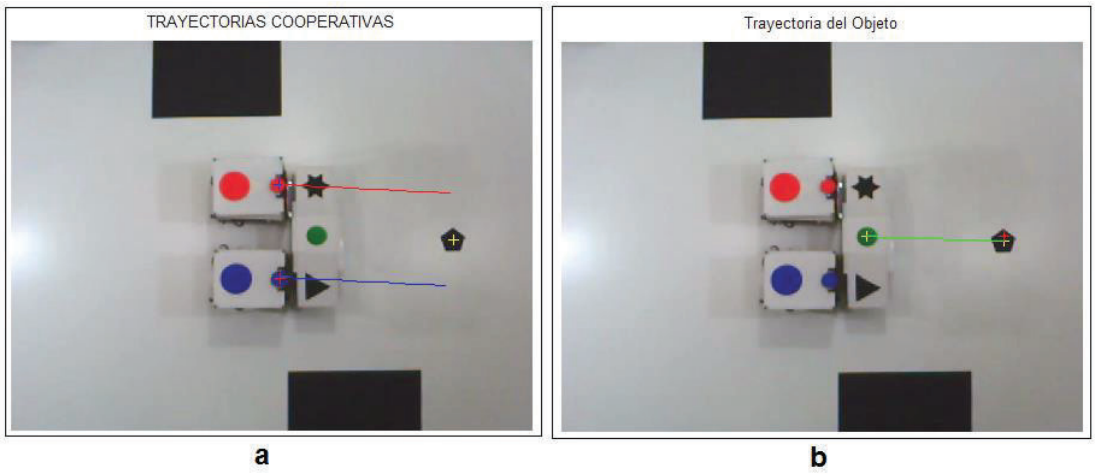
a) Objeto Inicial b) Objeto modificado

Figura 4-56 Cambios en el Objeto.

Una vez que los dos robots ya están ubicados frente al objeto, se lo eleva y se traza la ruta cooperativa, así como también la ruta para el objeto, esto se aprecia en la Figura 4-57.

Esta ruta es recta debido a que el punto final se encuentra dentro de los límites de los puntos de ubicación de los dos robots, por consiguiente, al realizar una ruta

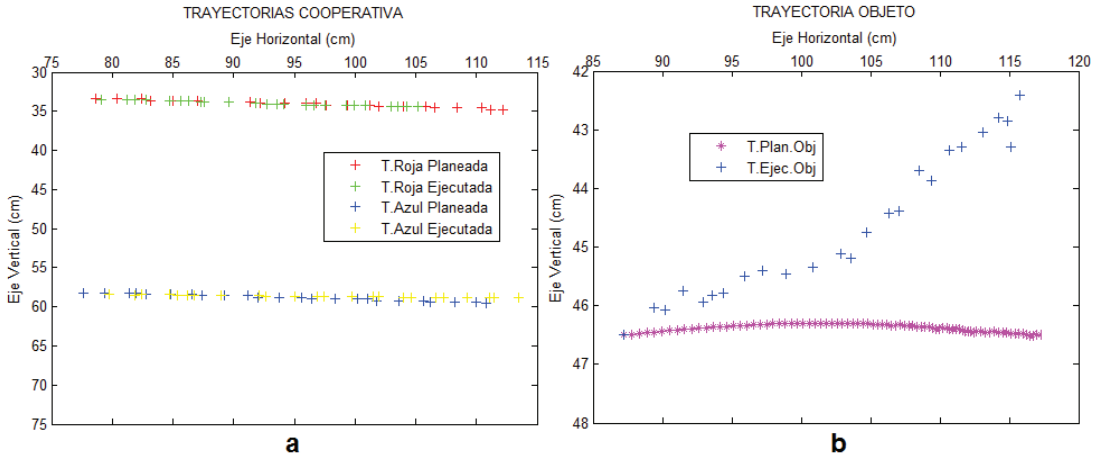
curva, los radios de curvatura de los robots son muy cerrados que no permiten un movimiento bien coordinado.



a) Rutas para los robots b) Ruta para el objeto.

Figura 4-57 Rutas cooperativas prueba 03.

Con las rutas construidas se ejecuta el algoritmo de operación cooperativa, obteniéndose como resultado del movimiento en las gráficas de la Figura 4-58.

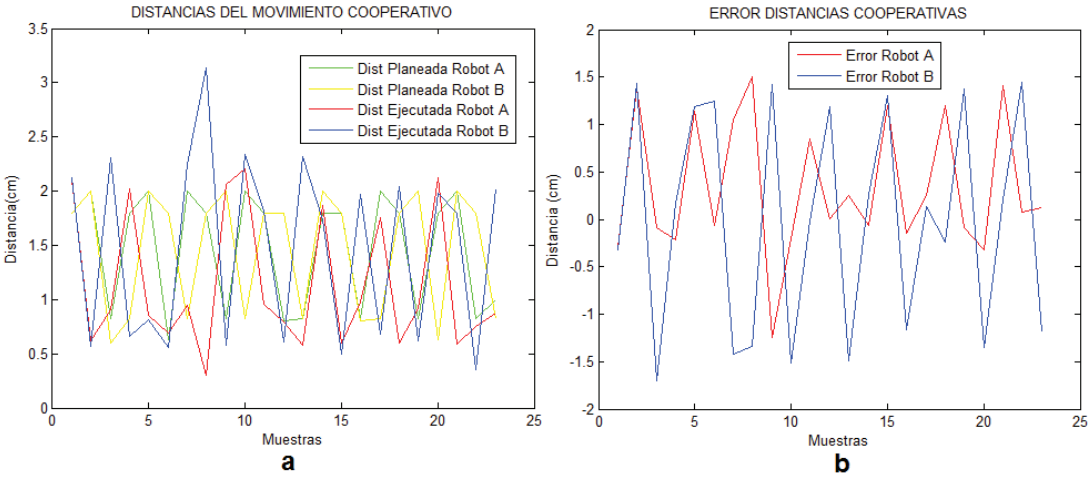


a) Rutas cooperativas b) Ruta del Objeto.

Figura 4-58 Movimiento Cooperativo prueba 03.

En la Figura 4-58a se observa que las rutas planteadas para cada uno de los robots son seguidas en buena forma y con un error de posición pequeño. En lo que respecta al objeto este llega al punto objetivo con un error de posición debido a los errores acumulados y generados por cada uno de los agentes.

Para entender de manera clara las distancias ejecutadas por cada uno de los agentes se realiza el cálculo de las distancias absolutas generadas por los agentes, esto se muestra en la Figura 4-59.



a) Distancias Cooperativas de los Robots b) Error de las distancias

Figura 4-59 Distancias Cooperativas de los Robots prueba 03.

En la Figura 4-59a, se aprecian las distancias absolutas propuestas por el maestro y las realizadas por los esclavos, en donde se valora que las diferencias son pequeñas dando a notar que las rutas son bien seguidas por parte de los robots. Para tener una mejor apreciación de las diferencias entre las distancias se grafican los errores, esto se muestra en la Figura 4-59b, nótese que el error no supera de los 1.4 cm como máximo. Para más detalle se tiene la Tabla 4-5.

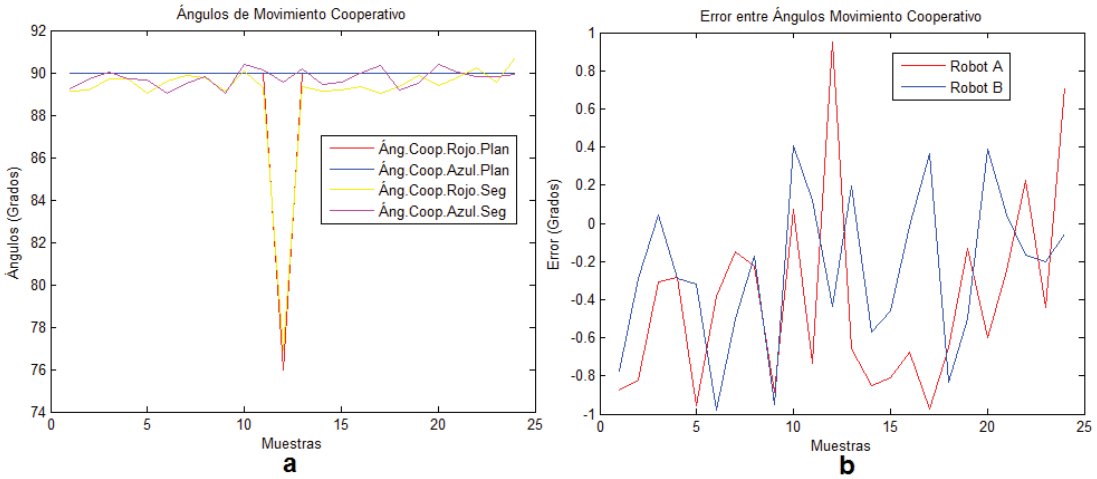
Tabla 4-5 Resumen de Errores de Distancia Cooperativos prueba 03

Robot_A		Robot_B	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	1.40	Error máximo	1.40
Error mínimo	0.10	Error mínimo	-0.15

En la Tabla 4-5 se puede observar los valores de errores medidos en centímetros son pequeños esto permite entender que el sistema cooperativo se encuentra siguiendo, mediante un buen accionar, las rutas planificadas por el maestro por

cada uno de los agentes y que las acciones de control realizadas han minimizado los errores de distancia absoluta entre los puntos de movimiento.

Los ángulos generados por los esclavos y comprados con los propuestos por el sistema maestro son los mostrados en la Figura 4-60.



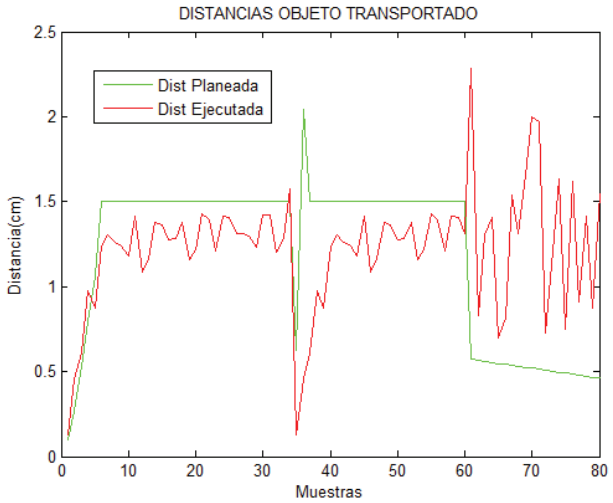
a) Ángulos de los robots b) Errores de los ángulos

Figura 4-60 Ángulos de los robots en el sistema cooperativo prueba 03.

En la Figura 4-60a, se aprecia que los ángulos propuestos son muy similares a los ángulos del movimiento, indicando el buen funcionamiento de los controladores. Esto se ratifica con la figura 4-60b en la que se aprecia que el error generado con respecto al ángulo no supera el ±1°.

Para finalizar esta prueba se realiza el cálculo de la distancia absoluta generada por el objeto durante su traslado, esto se parecía en la Figura 4-61, en donde se aprecia que la diferencia entre las distancias propuestas y realizadas son pequeñas, pero mayores que las generadas por los robots, esto es debido a que los movimientos realizados por el objeto no es totalmente las acciones realizadas por los agentes debido a que el objeto no está fijo a los agentes, es decir como únicamente fue levantado del piso, durante el movimiento este sufre desplazamientos de sus posiciones en las herramientas, horquillas, que hacen que

el error de posición sufra incrementos considerables, el resumen de estas gráficas se tiene en la Tabla 4-6.



a) Distancias absolutas del Objeto b) Error de la distancia.

Figura 4-61 Distancias ejecutadas por objeto prueba 03.

Tabla 4-6 Resumen Movimiento del Objeto prueba 03.

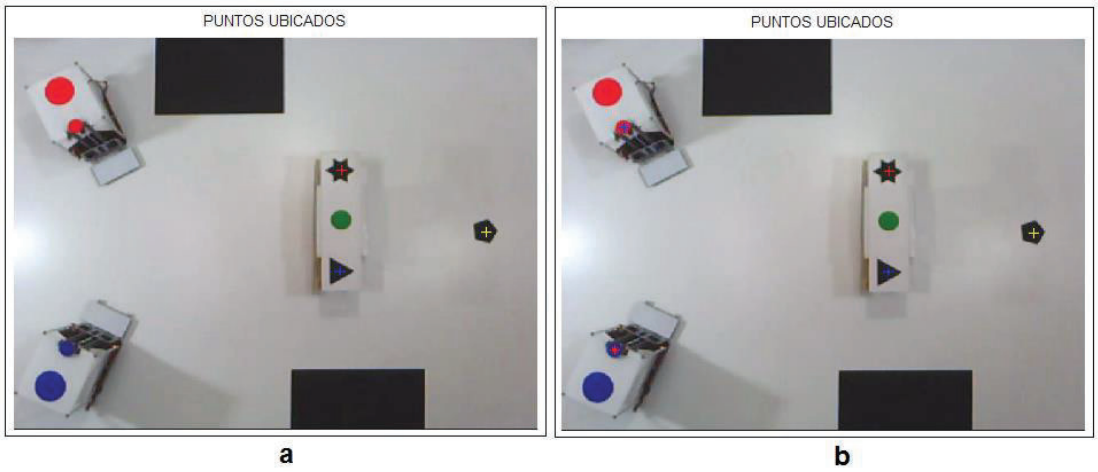
Movimiento del Objeto	
Tipo	**Valor (cm)**
Error máximo	2.40
Error mínimo	0.10

4.5. PRUEBA 04: MOVIMIENTO COOPRATIVO 2

Para esta prueba se ha modificado la posición inicial de los robots, así como también algunas líneas de programación dentro de los algoritmos del sistema maestro. Como se mencionó en la prueba 03 existen varias fases que son comunes razón por la que no se entrará en mayores detalles al respecto.

4.5.1. OBTENCIÓN DE PUNTOS.

Las nuevas posiciones son las mostradas en la Figura 4-62.



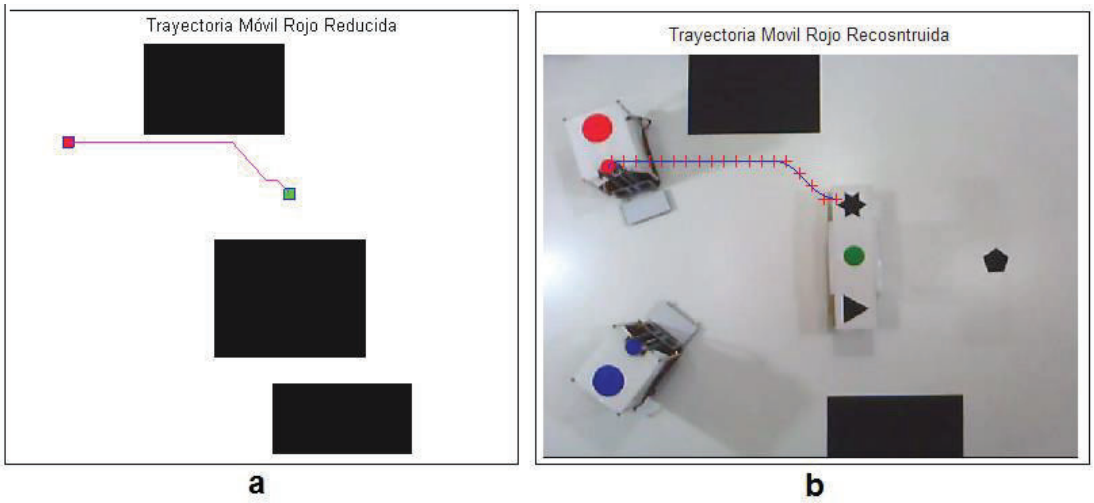
a) Puntos finales b) Puntos Finales e Iniciales

Figura 4-62 Puntos obtenidos prueba 04.

Figura 4-63 Operaciones de compresión y ubicación de puntos prueba 04.

4.5.2. DESPLAZAMIENTO DE LOS ROBOTS INDIVIDUALES

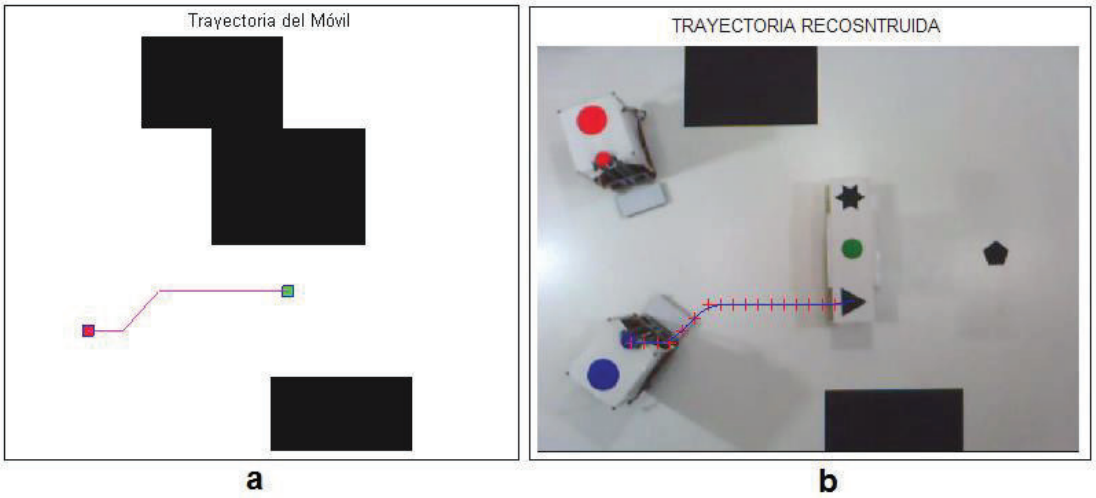
Para el Robot_A se ha determinado la ruta mostrada en la figura 4-64.



a) Ruta comprimida Robot_A b) Ruta reconstruida Robot_A.

Figura 4-64 Ruta Robot_A prueba 04.

Para el Robot_B se han determinado la ruta mostrada en la figura 4-65.

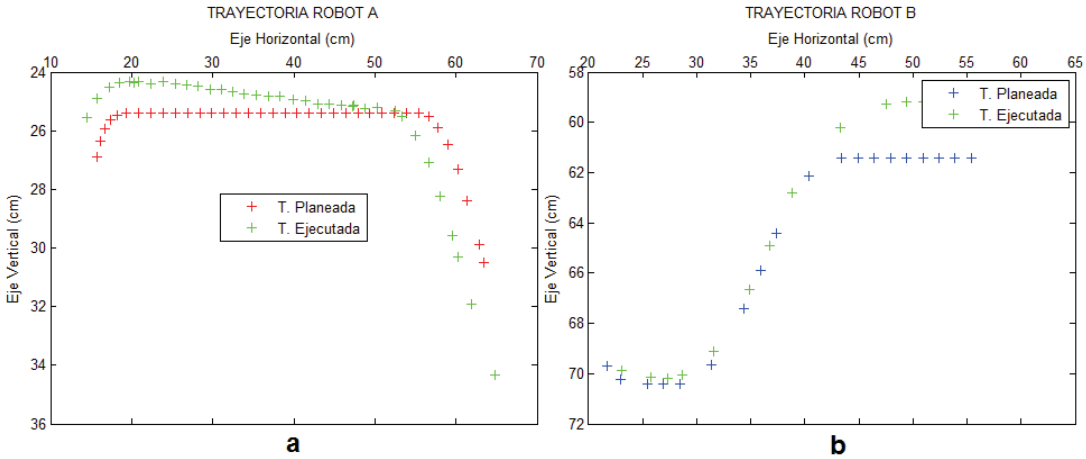


a) ruta comprimida Robot_B b) Ruta reconstruida Robot_B.

Figura 4-65 Ruta Robot_B prueba 04.

Con las rutas calculadas se ejecutan los algoritmos de movimiento para cada uno de los robots de forma individual, obteniéndose las gráficas de la Figura 4-66, en

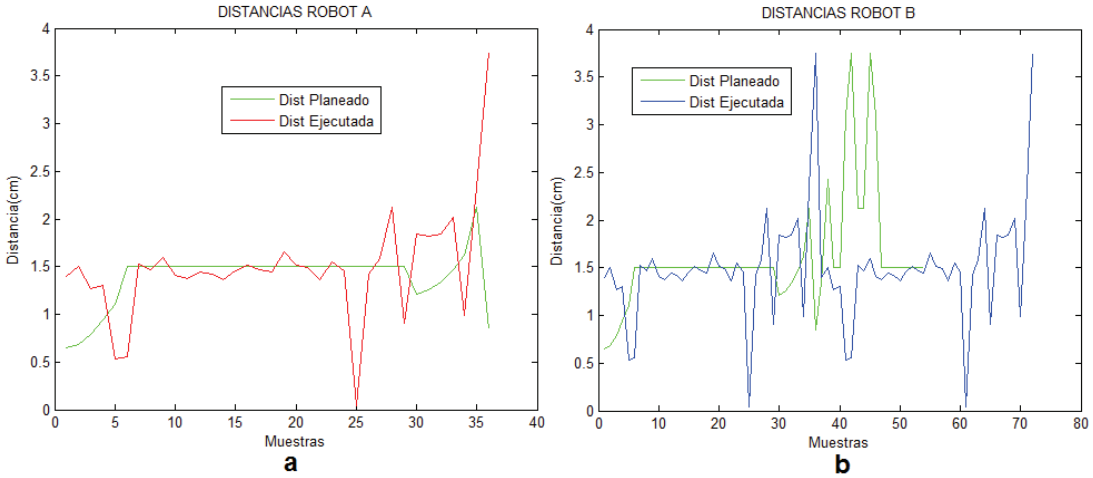
las que se puede apreciar que los robots siguen las rutas planificadas con un buen grado de apego a las mismas tanto para el Robot_A como para el Robot_B.



a) Rutas del Robot_A b) Rutas del Robot_B.

Figura 4-66 Rutas individuales de los robots prueba 04.

Se calcula las distancias absolutas del movimiento, obteniéndose la Figura 4-67 para los movimientos individuales de cada uno de los robots.

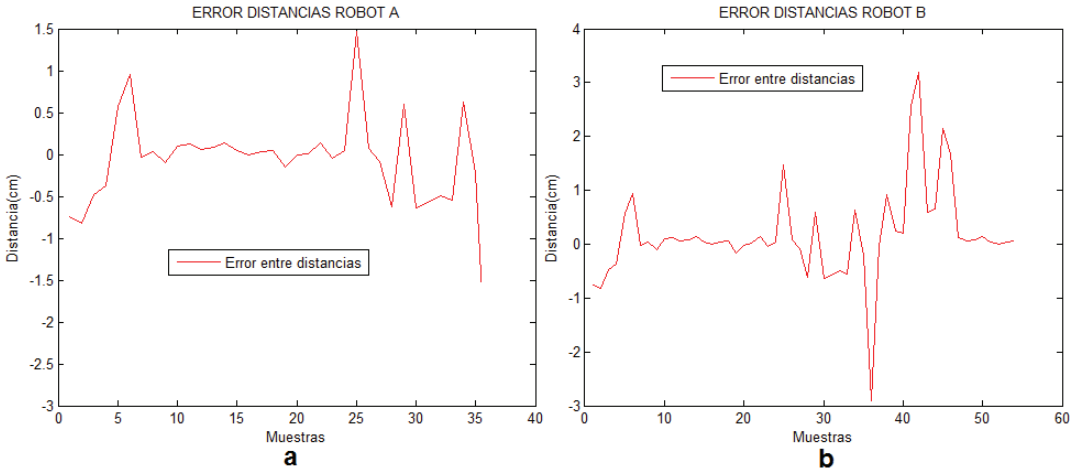


a) Distancias Robot_A solo b) Distancia Robot_B solo.

Figura 4-67 Gráficas de distancias generadas movimiento individual prueba 04.

Con los cambios realizados se puede apreciar que las distancias generadas hora son más similares, esto se aprecia en la Figura 4-67a y Figura 4-67b. Observese bien que las distancias son mucho más cercanas, por consiguiente los errores

generados también se van a minimizar, para lo cual se grafican los errores obteniendo la Figura 4-68.



a) Error de distancia Robot_A Solo. b) Error de distancia Robot_B.

Figura 4-68 Gráfico de los errores de distancia movimiento solos prueba 04.

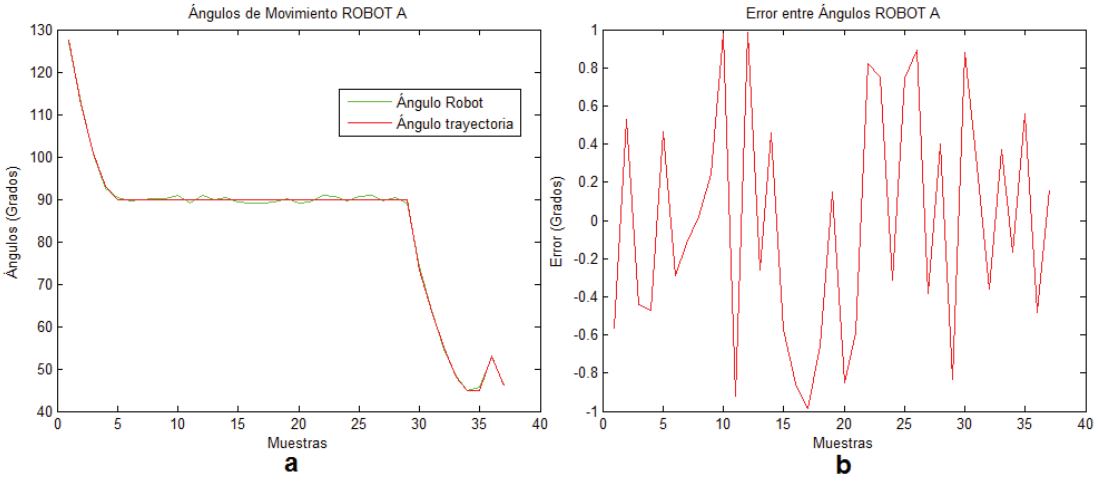
En efecto los errores se han minimizado, permitiendo que la ruta ejecutada sea más similar a la ruta planificada.

Para vizualizar esta reducción se muestra en la Tabla 4-7.

Tabla 4-7 Tabla Resumen Movimiento Individual prueba 04.

Robot_A		Robot_B	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	1.50	Error máximo	3.00
Error mínimo	-0.80	Error mínimo	-0.10

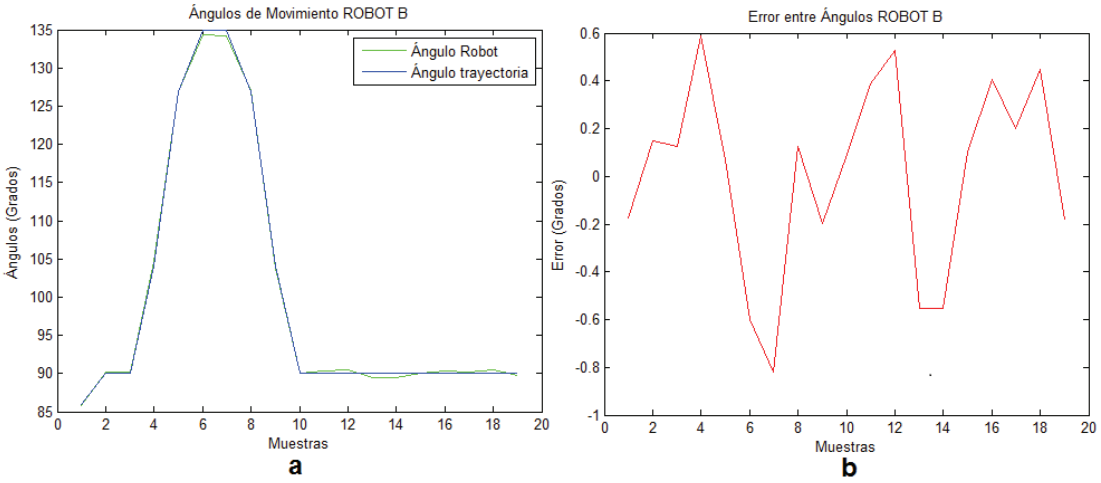
Para identificar el comportamiento con respecto a los controladores difusos se realizan las gráficas de los ángulos, para lo cual se tiene la Figura 4-69.



a) Ángulos del Robot_A b) Error del Robot_A.

Figura 4-69 Ángulos del Robot_A prueba 04.

Se aprecia que la diferencia entre los ángulos generados y los ángulos propuestos para el Robot_A son cada vez mucho menores, de esto se tiene que el error es mucho más pequeño, esto se aprecia en la Figura 4-69b en la cual se ratifica el buen funcionamiento del controlador. Para el caso del Robot_B se tiene la Figura 4-70, obteniéndose resultados similares que para el Robot_A.

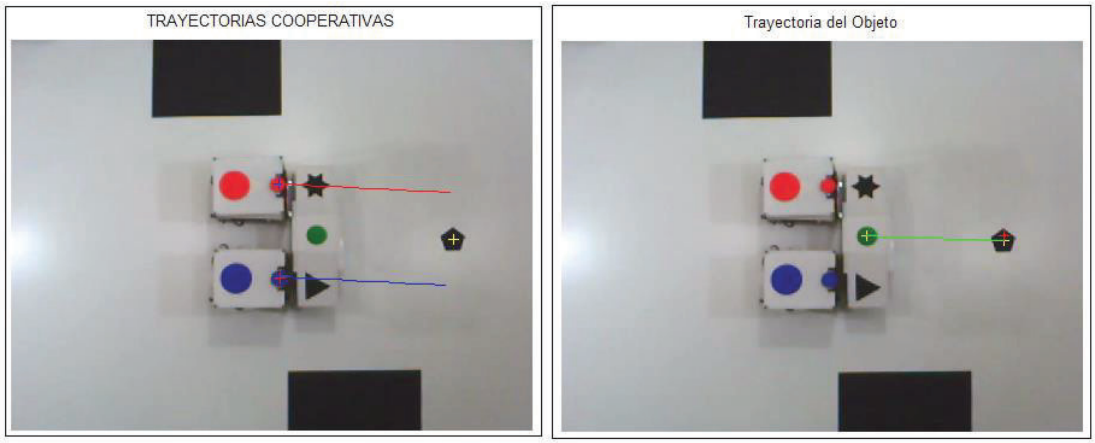


a) Ángulos del Robot_B b) Error del Robot_B.

Figura 4-70 Ángulos del Robot_B prueba 04.

4.5.3. DESPLAZAMIENTO COOPERATIVO.

Se ejecuta el algoritmo para el cálculo de las rutas cooperativas para los robots y se obtienen las imágenes mostradas en la Figura 4-71.

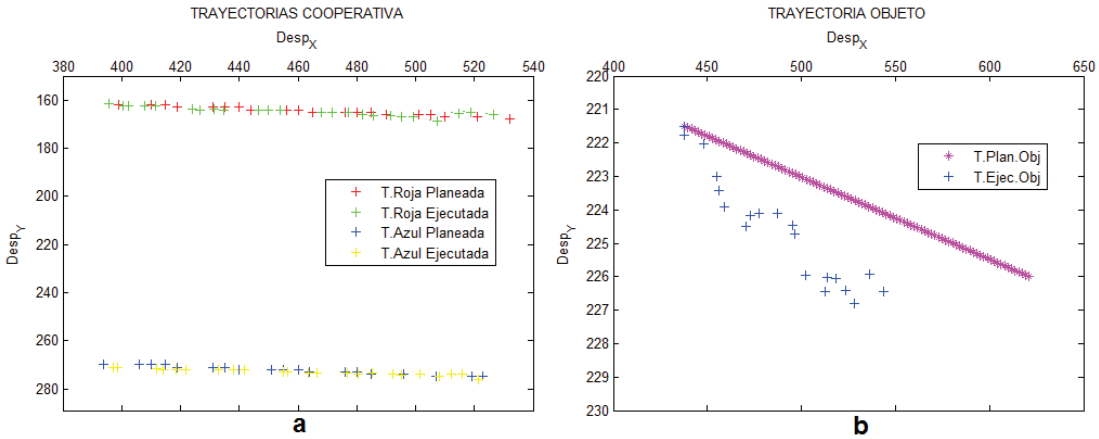


a b

a) Rutas para los robots. b) Ruta para el objeto.

Figura 4-71 Rutas cooperativas prueba 04.

Una vez obtenidas las rutas del algoritmo, se ha ejecutado el algoritmo de movimiento cooperativo y de la ejecución del mismo se han obtenido las gráficas mostradas en la Figura 4-72.



a b

a) Ruta de los robots b) Ruta del objeto.

Figura 4-72 Rutas cooperativas prueba 04.

En la Figura 4-72a se aprecia que las rutas cooperativas son muy similares, pero lo que hay a destacar de esta experimentación es que el objeto no ha llegado al punto

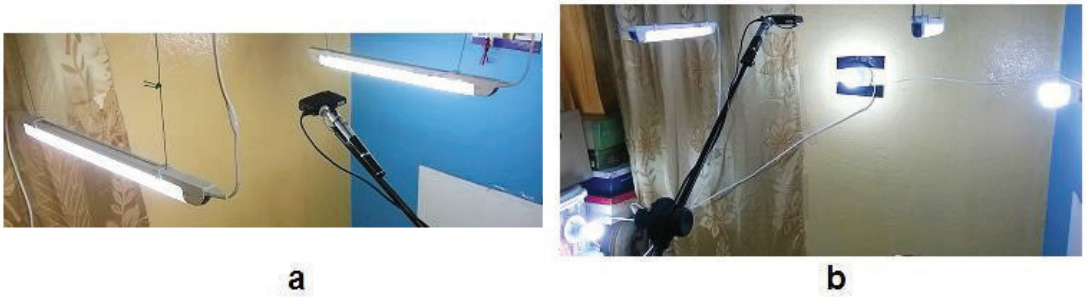
final esto se aprecia en la Figura 4-72b, esto es debido a que el sistema entregó un error el mismo que indica:

Error using imaqdevice/getsnapshot (line 72)
A timeout occurred during GETSNAPSHOT.

Se buscó información relacionada con este tipo de error, de lo cual fue imposible encontrarla, incluso en la página de apoyo de Matlab, ante lo cual se empezó a ensayar con los tiempos y demás opciones que se pueden hacer a nivel de software. No hubo una buena respuesta ante lo cual se revisó información sobre las cámaras y formas de adquisición de las imágenes.

De esta información se determinó que la falta de iluminación puede causar este tipo de errores debido a que el sensor que capta la imagen se demora en hacer la captura y transmitirlo vía USB al computador, esto genera el Timeout.

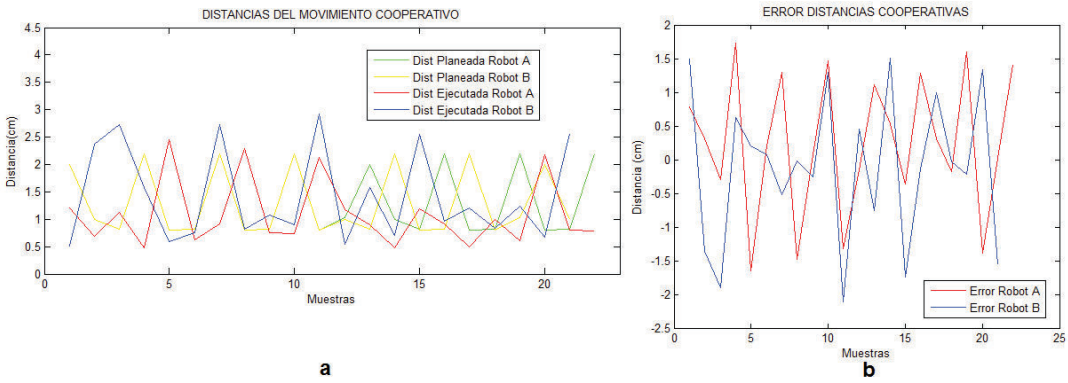
Para lo cual se ha modificado el sistema de iluminación dando resultados algo alentadores debido a que: en las experimentaciones previas este tipo de falla se presentaba a con mucha más frecuencia, una vez realizado el cambio en el sistema de iluminación, la falla persiste, pero de forma esporádica. El cambio realizado al sistema de iluminación se lo puede apreciar en la Figura 4-73.



a) Iluminación Inicial b) Iluminación modificada

Figura 4-73 Cambio sistema de iluminación.

A pesar del problema generado por la falla de la cámara, se recolectaron varios datos, con los cuales se realiza el análisis de distancias absolutas para continuar con la validación del sistema, obteniéndose las gráficas de la Figura 4-74.



a) Distancias Cooperativas de Robots b) Error distancia movimiento cooperativo.

Figura 4-74 Distancias cooperativas prueba 04.

Se observa en la Figura 4-74 que la diferencia entre las distancias propuestas y generadas por el sistema son pequeñas, indicando de esta forma que las rutas ejecutadas por los dos robots son cada vez más similares a las planificadas por el sistema maestro.

Para apreciar la minimización de errores que se da en el sistema cooperativo se tiene la Tabla 4-8, en base de la gráfica de la Figura 4-74b.

Tabla 4-8 Resumen Distancia Cooperativa prueba 04.

Robot_A		Robot_B	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	1.65	Error máximo	1.50
Error mínimo	-0.30	Error mínimo	-0.7

De igual manera se ha obtenido datos de distancia del desplazamiento del objeto, para lo cual se tiene la gráfica en la Figura 4-75.

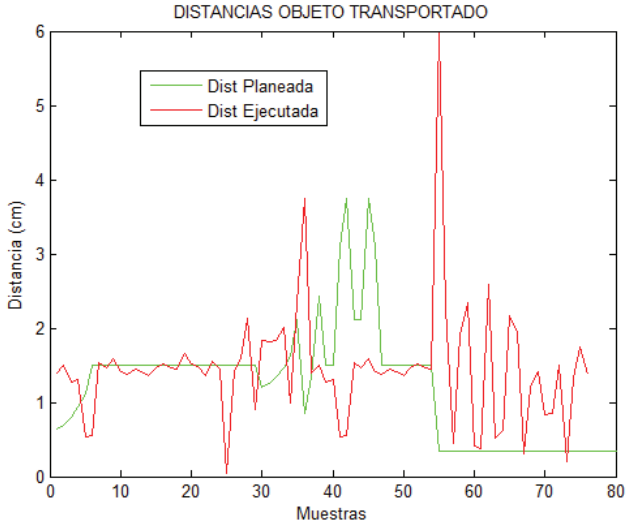


Figura 4-75 Distancia del objeto prueba 04.

En la figura 4-75 se aprecia que hasta el punto de ejecución del sistema se tenía picos de error de 6cm como máximo y -0.5cm como mínimo, un error muy aceptable para el movimiento, debido a que la mayoría de los errores se mantienen dentro de una banda entre 2.3 y -0.5cm.

4.6. PRUEBA 05: MOVIMIENTO COOPRATIVO 3

Para esta prueba se han colocado a los dos agentes, uno junto al otro, esto se aprecia en la Figura 4-76, en donde se observan también los puntos encontrados para el movimiento.

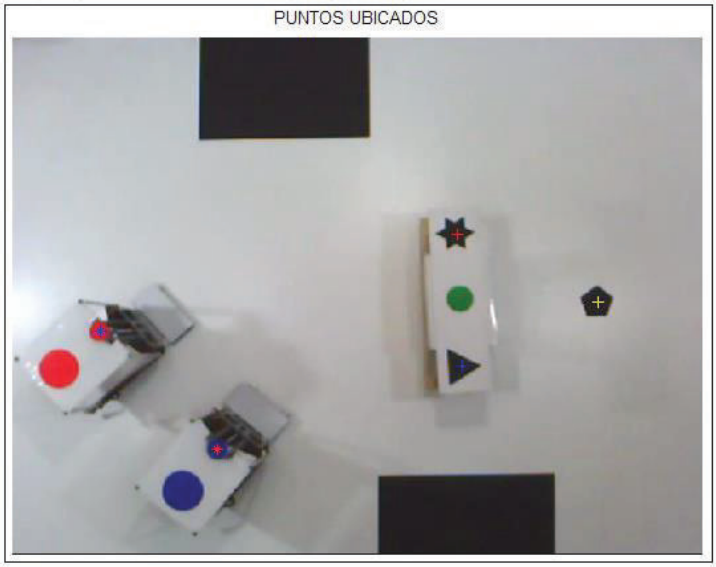


Figura 4-76 Puntos ubicados prueba 05.

Con los puntos encontrados se ha tratado al entorno para obtener la Figura 4-77, con los nuevos puntos dentro del mismo.

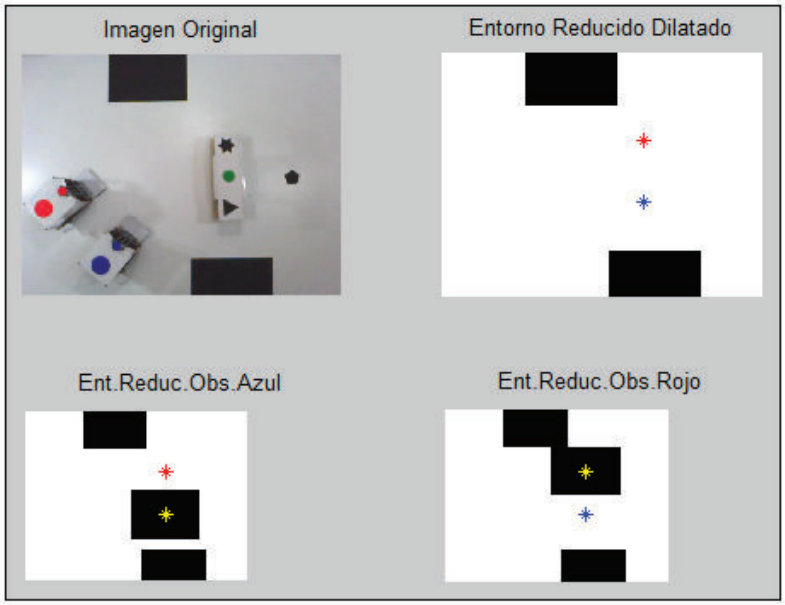
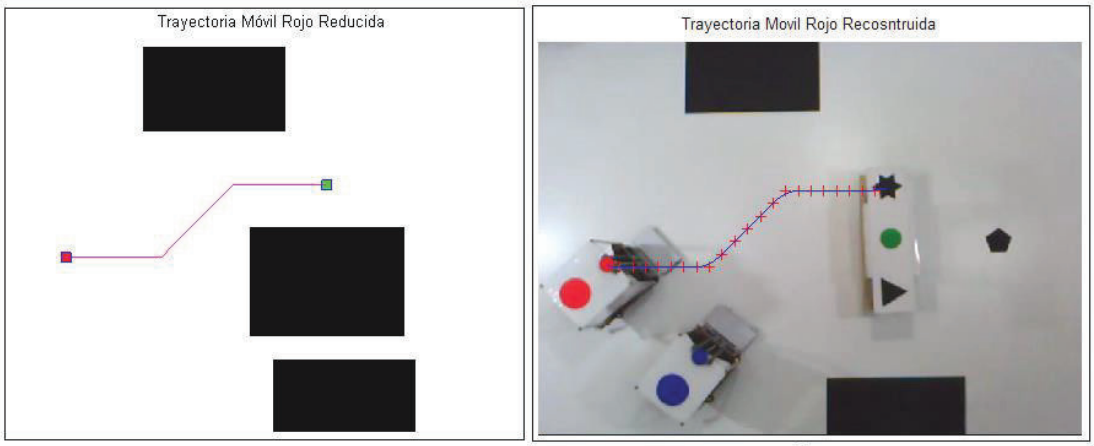


Figura 4-77 Entorno tratado con puntos ubicados prueba 05.

4.6.1. DESPLAZAMIENTO DE LOS ROBOTS INDIVIDUALES.

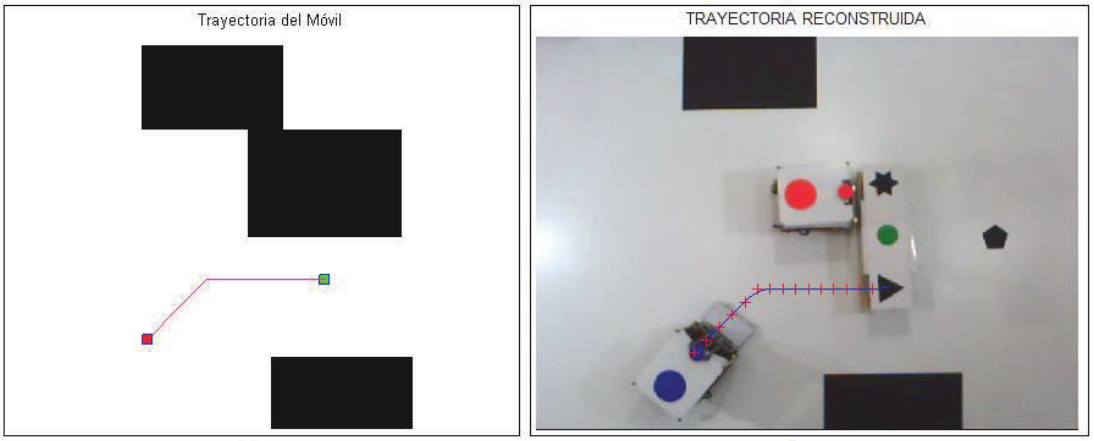
Para el movimiento de los robots se han obtenido las imágenes mostradas en la Figura 4-78 para el Robot_A y Figura 4-79 para el Robot_B.



a b

a) Ruta reducida b) Ruta reconstruida.

Figura 4-78 Rutas Robot_A prueba 05.



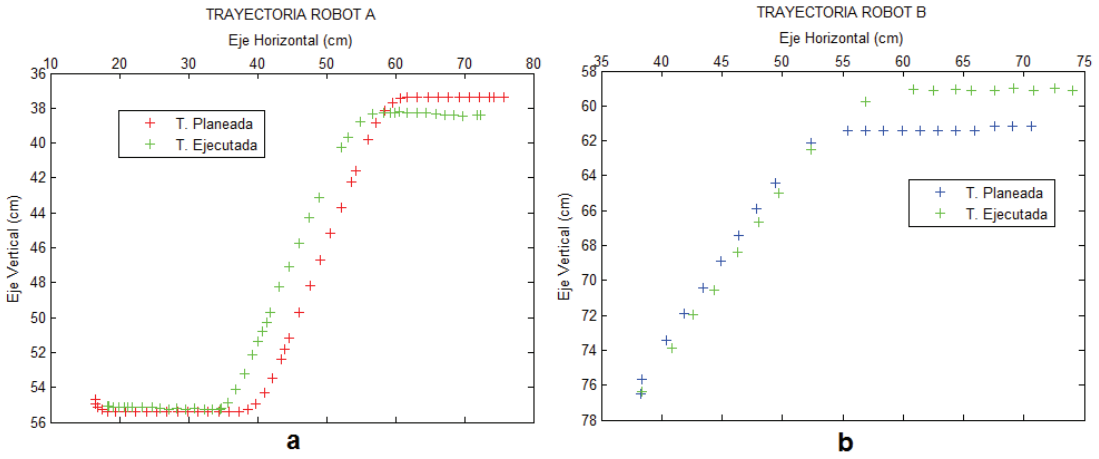
a b

a) Ruta reducida b) Ruta reconstruida.

Figura 4-79 Ruta Robot_B prueba 05.

Con las rutas obtenidas se realizan los movimientos de cada uno de los robots, obteniéndose las gráficas de la Figura 4-80 para cada uno de ellos, hay que observar en particular las rutas son muy próximas, no se han dado colisiones, pero

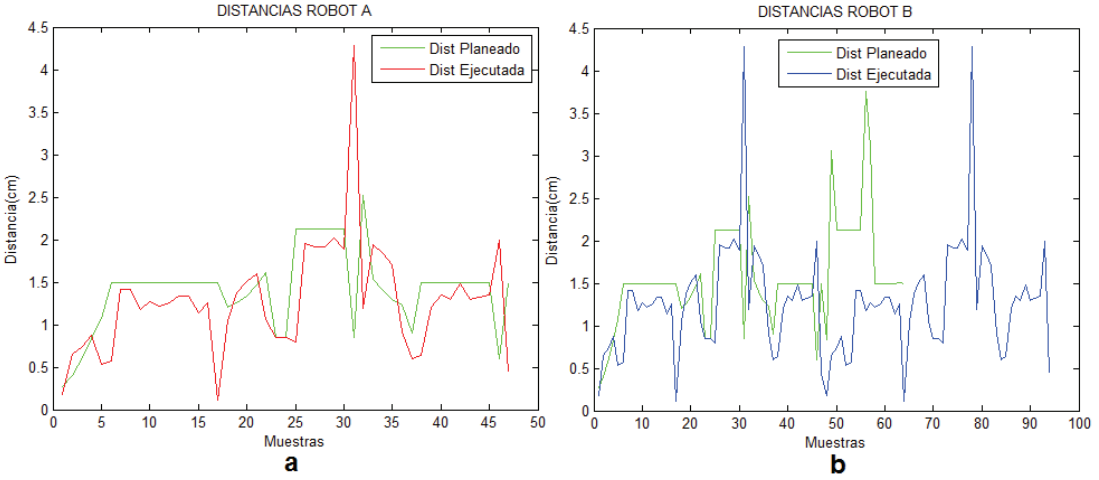
esto da a pensar que el espacio para el entorno en bastante reducido para los dos robots.



a) Ruta Robot_A b) Ruta Robot_B.

Figura 4-80 Rutas robots prueba 05.

Se calcula las distancias absolutas de movimiento, encontrando las gráficas de la Figura 4-81.

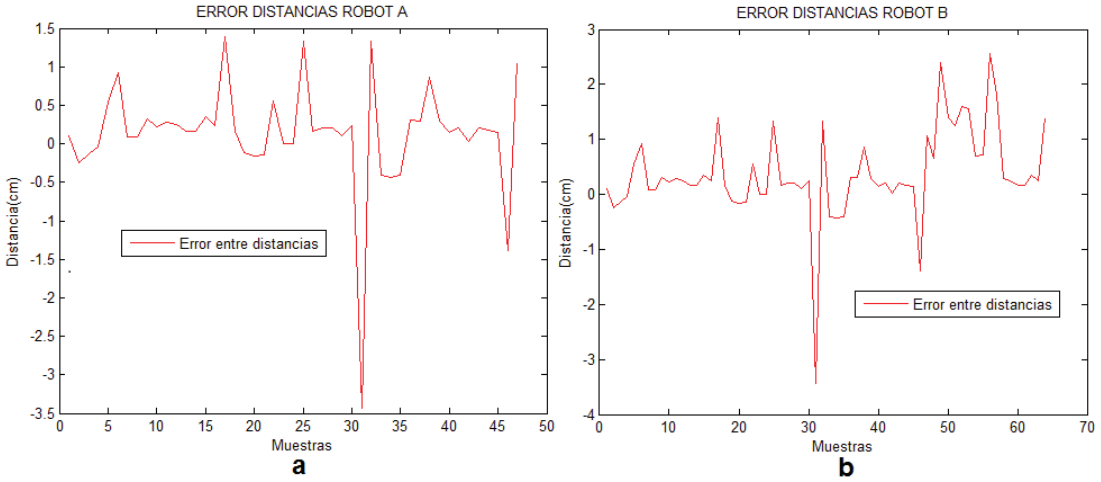


a) Distancia absoluta Robot_A solo b) Distancia absoluta Robot_B solo.

Figura 4-81 Distancia absoluta robots prueba 05.

En la Figura 4-81 se aprecia que las distancias están muy cercanas para los dos robots, de tal forma que el error cometido en distancia será pequeño de esta manera

se obtiene el error en distancia absoluta del movimiento, obteniéndose la Figura 4-82, en la misma que se puede apreciar que el error no sobrepasa de los 2cm.



a) Error del Robot_A Solo. b) Error del Robot_B solo.

Figura 4-82 Errores absolutos de robots prueba 05.

Del análisis de estas gráficas se obtiene la Tabla 4-9 como resumen de los datos obtenidos con respecto a las distancias absolutas.

Tabla 4-9 Tabla Resumen Movimiento Individual prueba 05.

Robot_A		Robot_B	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	-1.50	Error máximo	-3.30
Error mínimo	-0.50	Error mínimo	0.1

En lo respecta a los ángulos de control de los robots, los resultados obtenidos dan valores muy similares a las pruebas anteriores, de tal forma que se puede asegurar que el sistema continúa funcionando adecuadamente.

Para ratificar esto se tiene la Figura 4-83. en la que se puede apreciar que los errores de los ángulos para los dos robots no superan del ±1°, un error muy

pequeño, considerando que las mediciones de ángulos están entre rangos de 0° a ±180°.

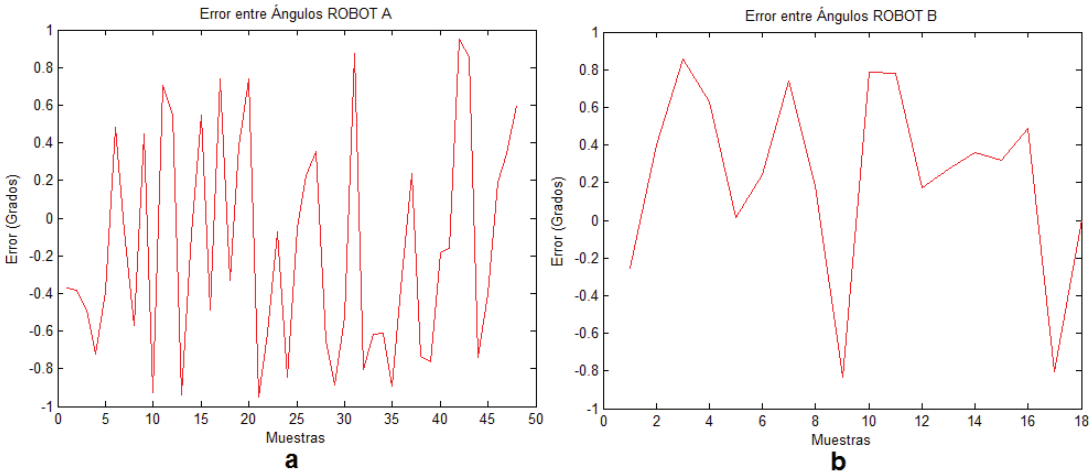
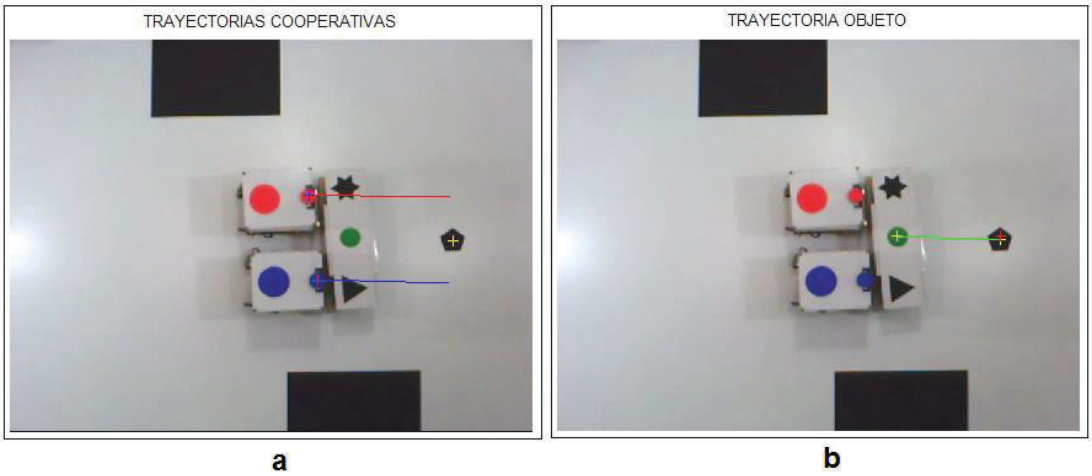


Figura 4-83 Error entre ángulos de los robots prueba 05.

4.6.2. DESPLAZAMIENTO COOPERATIVO.

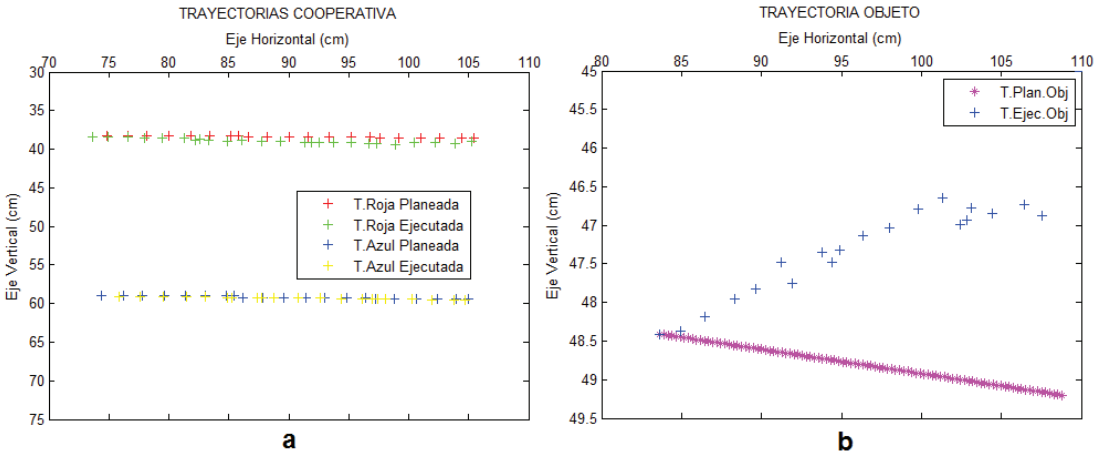
Para el desplazamiento cooperativo se obtuvieron las imágenes mostradas en la Figura 4-84, en donde se aprecia la ruta de cada uno de los robots, así como también la ruta que deberá seguir el objeto.



a) Ruta cooperativa para los robots. b) Ruta para el Objeto.

Figura 4-84 Rutas cooperativas prueba 05.

Con las rutas calculadas se ejecuta el algoritmo de movimiento cooperativo, obteniendo las imágenes de la Figura 4-85.

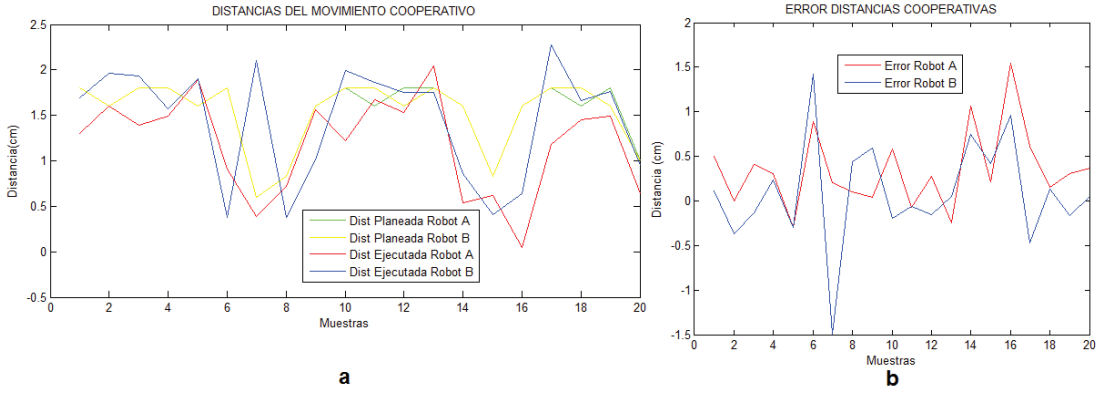


a) Movimiento cooperativo de los Robots b) Movimiento del objeto.

Figura 4-85 Movimiento cooperativo prueba 05.

De la Figura 4-85a, se aprecia que los movimientos de los agentes cooperativos van muy juntos cumpliendo con su función, ahora con respecto del movimiento del objeto, Figura 4-85b, los puntos finales del movimiento se encuentran muy cercanos, pero persiste un error de posición, es decir el objeto llaga hasta el punto de destino llevado por los robots cooperativos.

Para verificar el movimiento con respecto a distancia recorrida, se calcula la distancia absoluta, obteniendo la gráfica de la Figura 4-86.



a) Distancia Absoluta Robot_A b) Distancia Absoluta Robot_B.

Figura 4-86 Distancias absolutas cooperativas prueba 05.

En la figura 4-86 se aprecia que las distancias absolutas son muy próximas, por consiguiente, se entiende poco error, para ratificar esto se tiene la Figura 4-86b, en donde se tiene los errores.

El error es muy pequeño, no supera de los pocos. Para una mejor apreciación de estos datos se tiene la Tabla 4-10 en la que se resumen los valores más significativos de estas gráficas.

.

Tabla 4-10 Resumen Errores Cooperativos prueba 05

Robot_A		Robot_B	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	1.40	Error máximo	-1.50
Error mínimo	-0.30	Error mínimo	-0.10

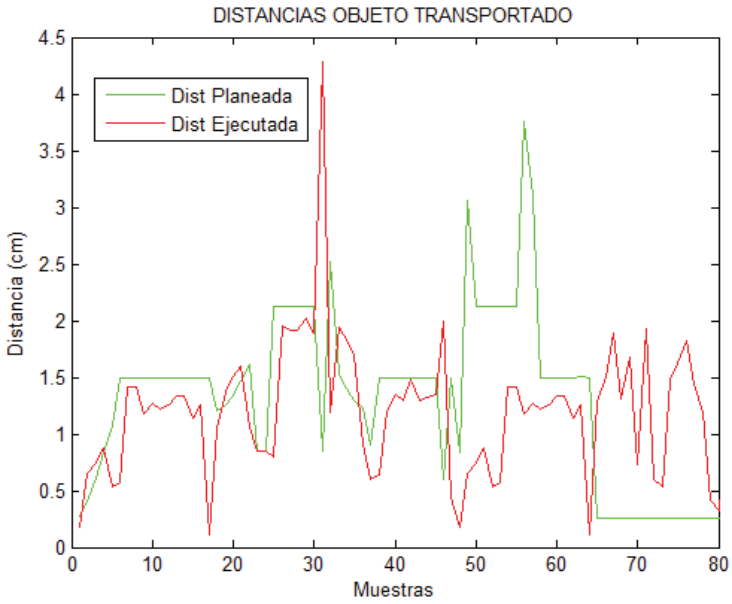


Figura 4-87 Distancia Movimiento cooperativo prueba 05.

4.7. PRUEBA 06: MOVIMIENTO COOPRATIVO 4

Para esta prueba se ha cambiado la ubicación del punto final para que los robots realicen un movimiento de forma curva con el objeto. Figura 4-88.

4.7.1. DESPLAZAMIENTO DE LOS ROBOTS INDIVIDUALES.

Figura 4-88 Puntos Obtenidos para movimiento prueba 06.

Con los puntos obtenidos se procesa el entorno, obteniendo las imágenes de la Figura 4-89 de las ubicaciones de los puntos para los movimientos dentro del entorno reducido y con los espacios que evitan colisiones.

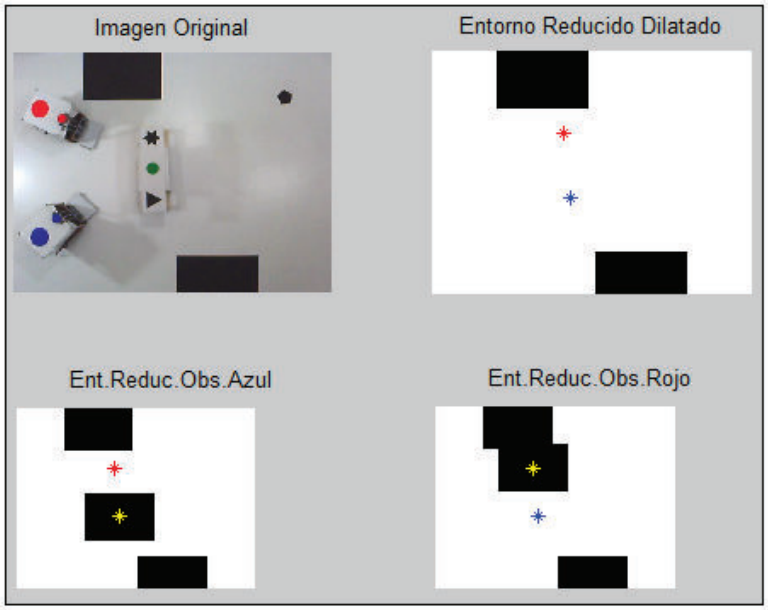
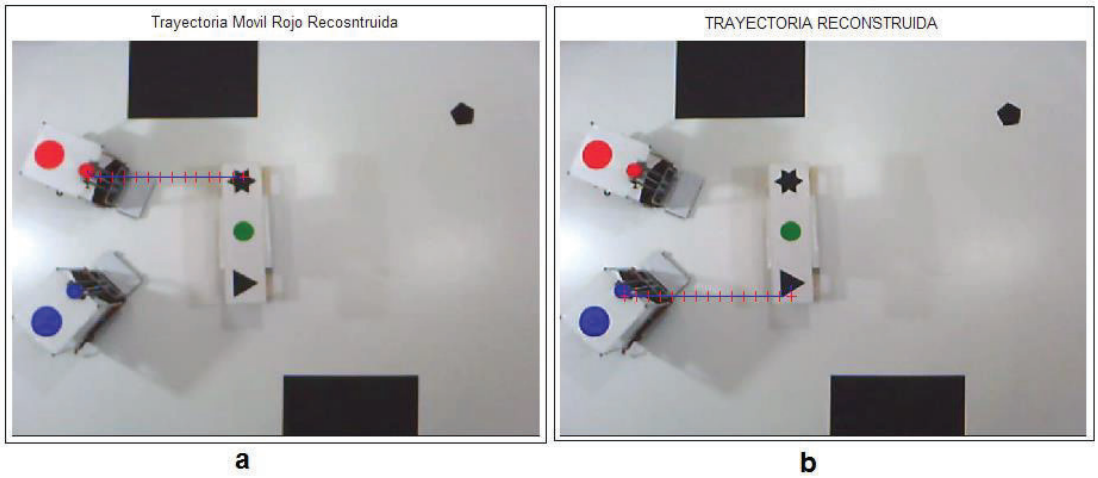


Figura 4-89 Puntos en entorno reducido prueba 06.

4.7.2. DESPLAZAMIENTO DE LOS ROBOTS INDIVIDUALES.

Con los puntos ubicados dentro del entorno obtiene las rutas tanto para el Robot_A como para el Robot_B, esto se aprecia en la Figura 4-90 para los dos agentes.



a) Ruta reconstruida Robot_A b) Ruta reconstruida Robot_B

Figura 4-90 Rutas reconstruidas para robots prueba 06.

Se ejecuta el algoritmo de movimiento individual de cada agente, obteniéndose la Figura 4-91, en donde se observa que el error de posición es pequeño, similar a los obtenidos en las experimentaciones anteriores.

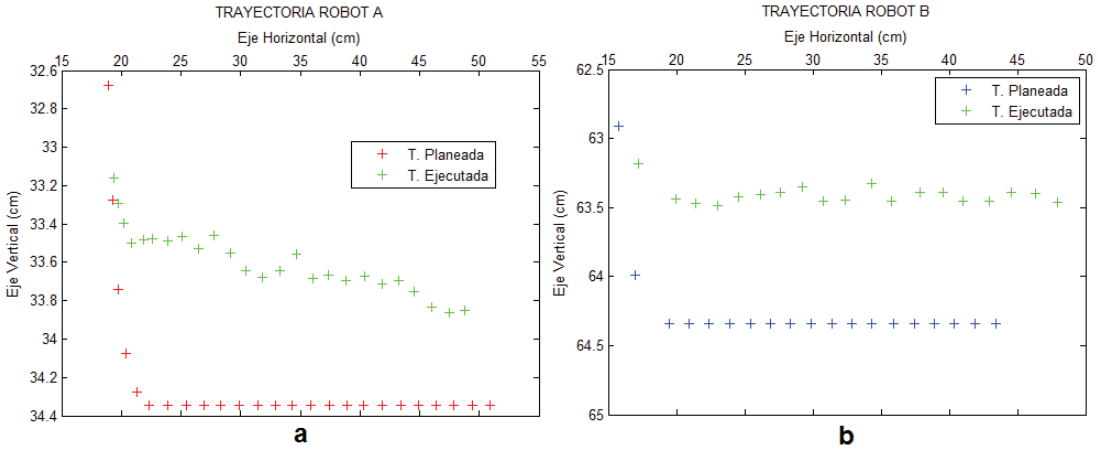
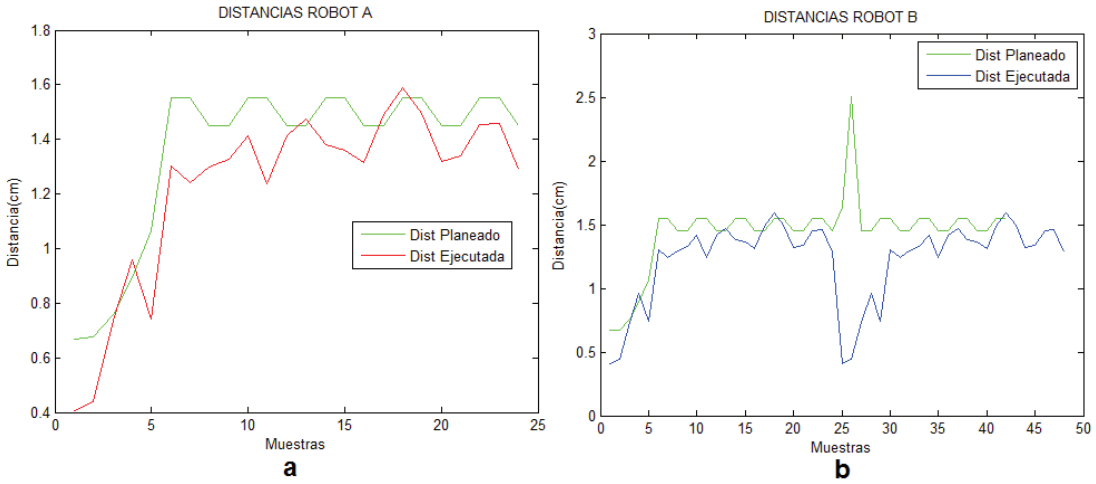


Figura 4-91 Rutas individuales de los robots prueba 06.

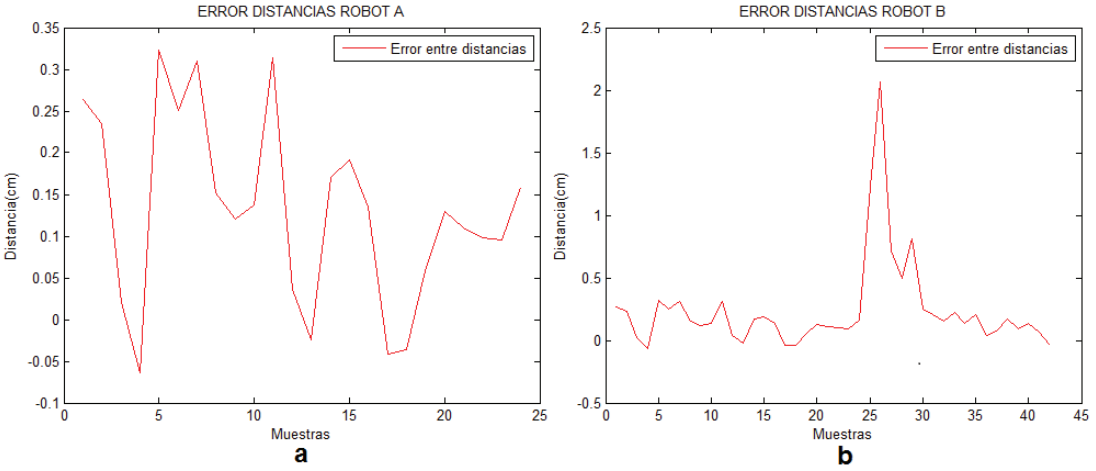
Se calcula la distancia generada por las Rutas, obteniendo las gráficas en la Figura 4-92.



a) Distancia del Robot_A solo b) Distancia del Robot_B solo.

Figura 4-92 Distancias Individuales prueba 06.

Se aprecia que las distancias generadas son muy similares a las producidas por los robots, para ratificar esto se calcula los errores, esto se aprecia en la Figura 4-93. En donde los errores producidos son pequeños.



a) Error Distancia Robot_A b) Error Distancia Robot_B

Figura 4-93 Error Rutas prueba 06.

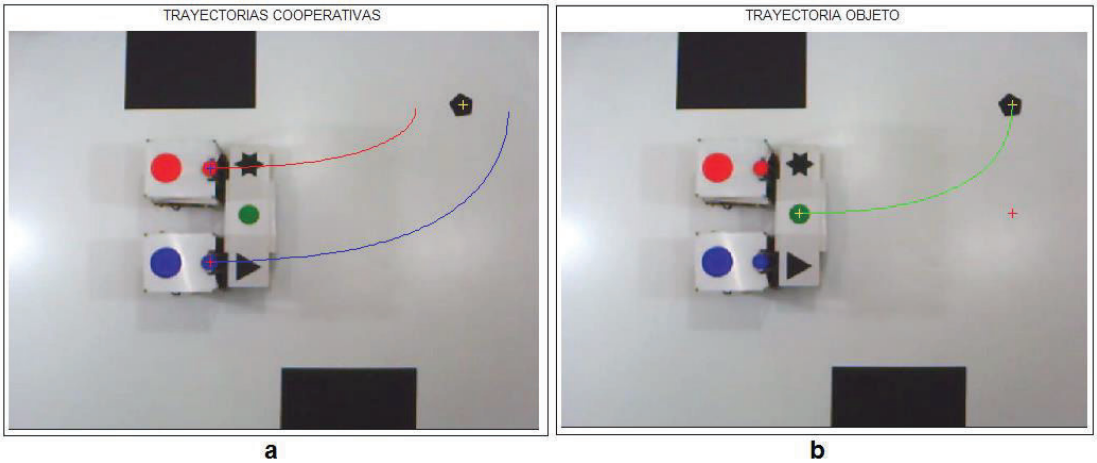
Para resumir la Figura 4-86 que corresponde a los errores cometidos en la distancia absoluta individual, se tiene la Tabla 4-10.

Tabla 4-11 Resumen Errores Movimientos Individuales prueba 06.

Robot_A		Robot_B	
Tipo	**Valor (cm)**	**Tipo**	**Valor (cm)**
Error máximo	-0.35	Error máximo	2.00
Error mínimo	-0.04	Error mínimo	-0.10

4.7.3. DESPLAZAMIENTO COOPERATIVO.

Las Rutas generadas para el movimiento cooperativo de los robots es el mostrado en la Figura 4-94, hay que notar que para este caso son rutas curvas para evaluar el comportamiento del sistema ante este tipo de movimientos.



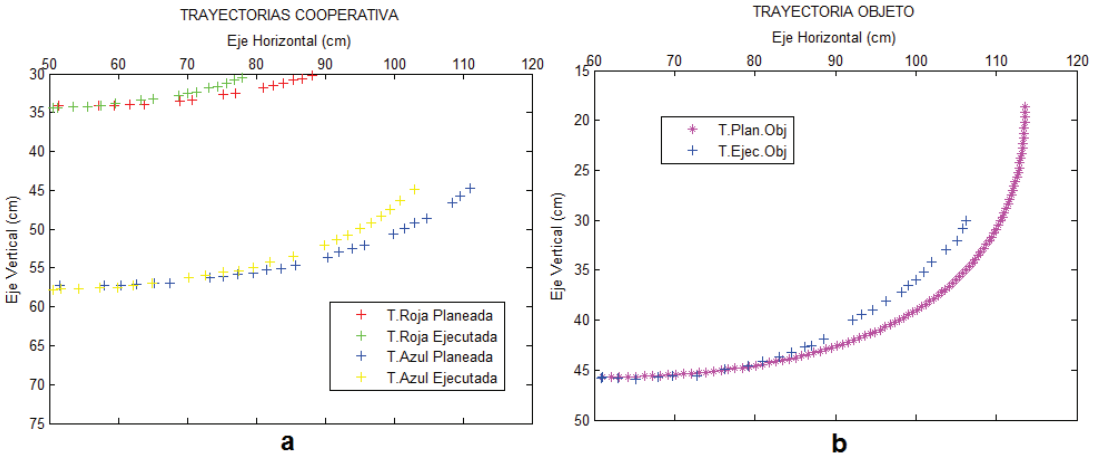
a) Rutas Cooperativas b) Ruta Objeto

Figura 4-94 Rutas Cooperativas prueba 06.

Aplicado el algoritmo de movimiento cooperativo se obtiene la gráfica de movimiento mostrado en la Figura 4-95 para cada uno de los robots, en la misma que se observa que las rutas planteadas para cada uno de los robots son seguidas de forma muy cercanas tanto para el Robot_A como para el Robot_B.

En lo que respecta al objeto este también va muy cercano a su ruta, pero en el caso de esta experimentación nuevamente se volvió a presentar el problema con la cámara al momento de la adquisición de la imagen.

Se realizaron varias pruebas, pero se ha tomado para este documento la que ha recorrido la máxima distancia el objeto, esto se puede visualizar en la Figura 4-95b.



a) Rutas de los robots b) Ruta del objeto.

Figura 4-95 Rutas cooperativas ejecutadas de los robots prueba 06.

Para un mejor análisis se calculan las distancias, obteniéndose la Figura 4-96. En esta figura se aprecia que las distancias van variando significativamente esto es debido a la forma de la ruta que tienen que ejecutar los dos robots, para identificar si esto es un problema de calculan los errores de distancia obteniéndose la Figura 4-95b en la misma que se tienen errores pequeños para el Robot_A y errores grandes para el Robot_B, esto se debe a que el Robot_A ejecuta un radio de curvatura menor al que ejecuta el Robot_B, de esto se desprende que las distancias ejecutadas por este robot son muy diferentes entre: la planteada y la ejecutada.

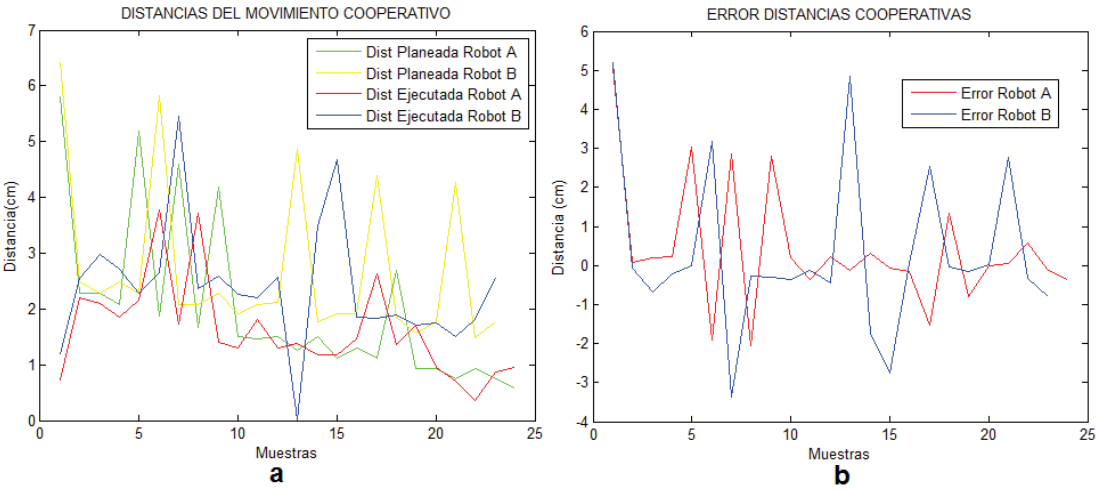


Figura 4-96 Distancia cooperativa prueba 06.

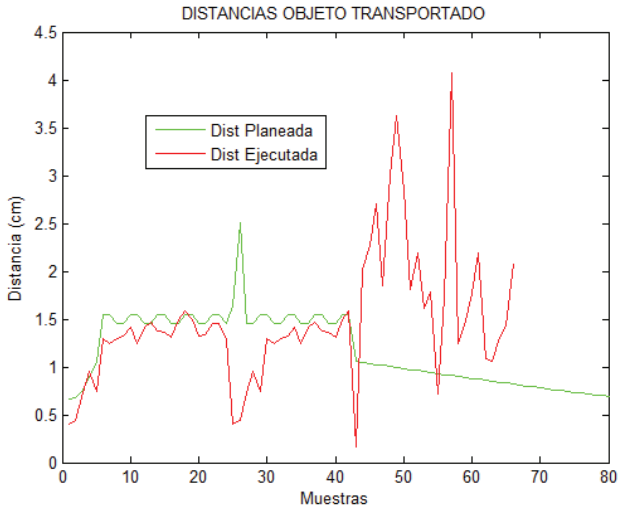


Figura 4-97 Error distancia cooperativa prueba 06.

De la Figura 4-97 se puede analizar que el objeto seguía muy bien en la distancia planteada con respecto a la ejecutada, pero llega un momento en que se vuelve muy errática esto es debido a que, al no tener un sistema de sujeción del objeto, al momento del transporte el objeto se mueve de las horquillas, haciendo que la distancia que se mueve sea modificada.

CAPÍTULO 5

5.1. CONCLUSIONES

Del trabajo realizado y explicado a lo largo de esta tesis se han obtenido varias conclusiones y se han generado recomendaciones las mismas que se enumeran a continuación:

- Se ha creado un sistema de visión artificial que permite controlar la navegación de un sistema robótico móvil cooperativo conformado por dos agentes robóticos dentro de un entorno semi estructurado, porque los agentes desde cualquier posición y con cualquier orientación que tengan son capaces de ubicarse frente al objeto a movilizar, elevarlo y desde allí de forma cooperativa transportarlo hasta el lugar definido como punto final del movimiento del objeto, esto se certifica en cada una de las pruebas realizadas para la validación del sistema.

- Del estudio realizado del estado del arte en relación a la visión artificial y los sistemas robóticos cooperativos se ha determinado que los sistemas existentes se apoyan en sistemas sensoriales adicionales como por ejemplo la realimentación de las llantas, que para el caso de este trabajo no ha sido utilizada de tal forma que el lazo cerrado de control del sistema se lo cierra mediante la visión artificial y dando como resultado el aporte principal de este trabajo a pesar que los resultados no son muy alentadores, el sistema cumple con su función del movimiento del objeto.

- Se automatizaron los dos montacargas de juguete y se los convirtió en dos mini-robots móviles que son capaces de recibir instrucciones vía comunicación inalámbrica para realizar acciones ordenadas por el sistema maestro, esto se puede evidenciar en la forma de la trama que es enviada desde el sistema maestro la cual está estructurada de tal forma que lleva información de movimiento, valor de PWM y tiempo de acción, eso para el caso de movimiento, en cambio para el caso de la elevación del objeto solamente recibe la orden de elevación y el robot cierra su lazo de control mediante la interrupción que es generado en la columna de la horquilla, de esta manera en cada uno de los

robots se tienen acciones de control en lazo cerrado en la horquilla y como un actuador simple para la navegación, con estas dos características de operación se ha logrado que los robots trabajen de manera cooperativa.

- Se han desarrollado una serie de algoritmos basados en el comportamiento de análisis del cerebro humano el cual se basa en establecer puntos de marca para ubicar los elementos, realizar las acciones de movimiento, comparar si está en el punto, de no ser así continúa con el mismo proceso una y otra vez hasta llegar al punto esperado. Todas estas acciones las realiza el sistema de visión artificial por consiguiente es quien está dotado de inteligencia y determina las acciones de control, por lo tanto, se constituye en el maestro del sistema y por el contrario como los robots únicamente ejecutan instrucciones dadas por el maestro, eso los convierte en esclavos que no están provistos de decisión e inteligencia. Este sistema en conjunto al mover el objeto desde el punto A hasta el punto B conjuntamente se forma el sistema cooperativo. Un punto importante a notar acá es que los puntos pueden ser colocados en cualquier lugar del entono semi estructurado, ya que las marcas que identifican los puntos de movimiento son buscadas dentro del entorno.

- La búsqueda del camino óptimo para cada uno de los robots permite que el sistema pueda encontrar una ruta libre de obstáculos y que esta se encuentre dentro del espacio de trabajo basado en la inteligencia artificial y el análisis de las imágenes, esto se logra mediante la aplicación del algoritmo A* y el sistema de dilatación de las imágenes que agrega holgura para el movimiento de los robots para evitar colisiones entre ellos y con el entorno.

- Al momento de la ejecución de las rutas determinadas por el sistema de visión artificial, los puntos por los que pasan cada uno de los robots se van registrando de tal forma que una vez terminado el movimiento se puede realizar el análisis de los movimientos realizados por cada uno de ellos, esto se puede evidenciar en el capítulo 4 de pruebas y resultados en el cual para cada una de las pruebas se tienen gráficas de los movimientos ejecutados obteniendo resultados bastante aceptables ya que el sistema presenta errores pequeños.

- De los análisis realizados a cada una de las gráficas del movimiento de los robots se puede asegurar que el sistema funciona cumpliendo la condición de mover el objeto desde el punto A hasta el punto B, obteniéndose como resultado que las rutas tienen fallas de posición las cuales medidas en píxeles parecerían errores grandes pero una vez convertidos a valores numéricos generan valores máximos de pocos centímetros, un error bastante tolerable considerando que no se dispone de realimentación de las llantas para identificar el desplazamiento realizado por las mismas y además las dimensiones del espacio son bastante considerables con respecto al error generado.

- La técnica de filtrado por color es un método de filtrado bastante aplicable pero tiene el problema que como las imágenes tienen en cada uno de los píxeles, que la conforman, valores de cada uno de los componentes de RGB, es esto se deduce que existe un alto grado de variabilidad con respecto a la iluminación, esto fue ampliamente comprobado durante las pruebas iniciales de operación se presentaban confusiones del sistema entre las tonalidades de verde y azul, razón por la que se tuvo que probar con distintas tonalidades de color verde para la operación de algoritmo de filtrado por color.

- El algoritmo A* tiene un buen desempeño, pero la carga computacional que demanda para su operación es bastante alta ya que tiene que barrer todos los espacios libres de la imagen, por lo tanto, en el caso de espacios pequeños la ejecución es rápida pero cuando se habla de una matriz 640x480 píxeles se está referenciando a gran cantidad de interacciones que hacen que el tiempo de búsqueda de la ruta tarde mucho. A este problema se lo ha podido solucionar mediante la reducción de la imagen en base de áreas que ha permitido que este tiempo se reduzca ya que la matriz de imagen es más pequeña, además hay que mencionar que como el tamaño de los robots es bastante grande es necesaria esta reducción de la imagen, a pesar que al momento de la reconstrucción al tamaño real de la imagen se genere un error al momento de la ubicación de los puntos originales.

- El plano en el cual sean colocadas las marcas de orientación tiene que ser el mismo ya que de no ser así el problema de paralaje que se genera hace que los ángulos calculados por el sistema maestro estén totalmente diferentes de los valores reales de orientación que tienen los agentes dentro del entorno.

- La iluminación juega un papel muy importante para un sistema de visión artificial debido a que desde el inicio de las pruebas de operación el sistema presentaba comportamientos extraños, es así que en una fase inicial se tuvo que aislar una ventana que está junto al entorno, con eso se mejoró en parte pero continuaban las fallas, posterior se daban los problemas de Timeout que genera Matlab, ante lo cual se modificó el sistema de iluminación dando como resultado la eliminación casi total de este problema.

- La utilización de cámaras de buena calidad es determinante para este tipo de trabajo, ya que al inicio de este proyecto se adquiría la imagen con una web cam común, pero la calidad de la imagen era muy deplorable y generaba confusiones por ejemplo en la búsqueda de patrones, al cambiar por la cámara fotográfica Premier DS-5060S se mejoró notablemente, pero a pesar de eso el problema de Time out registrado por Matlab evidencia el problema con la cámara.

- El espacio de trabajo es muy pequeño para rutas curvas prolongadas y los robots no pueden navegar, esto se verificó en la prueba 06, pero el hacer que el espacio sea mayor disminuiría la calidad de la imagen y de esta manera produciendo mayor cantidad de fallas por la calidad de la cámara, con esto se ratifica que la calidad de la cámara juaga un papel muy importante para este tipo de trabajo.

5.2. RECOMENDACIONES

- Un punto a ser muy considerado en el entorno es que este sea perfectamente plano, esto es debido a que durante las pruebas en muchas ocasiones el movimiento cambiaba repentinamente, analizando el funcionamiento de los robots, se encontró que existía una imperfección en el terreno el cual hacía que la ruta ejecutada se mueva cuando el agente pasaba por esa sección del entorno ya que la ruta estaba establecida y no es recalculada.

- Las baterías que son utilizadas para el sistema de comunicación tiene que ser de gran cantidad de unidades de flujo de corriente (amperios/hora), debido a que durante las fases de pruebas las baterías que estaban alimentando al sistema de control sufrían más descarga que el sistema de potencia y al tener los dos baterías de similares características se determinó que el sistema de comunicación posee un mayor consumo, por consiguiente para mejorar el tiempo de operación del sistema son necesarias baterías que posean este tipo de características.

- Para trabajos futuros como este es muy recomendable que la calidad de la cámara sea muy buena y sobre todo que esta tenga un buen tiempo para la comunicación con el computador personal, sobre todo que esté con la capacidad de autoenfoque automático para que este provea de imágenes mucho más definidas.

- Con respecto al elemento a ser movido o desplazado cooperativamente, éste tiene que ser sujetado por los agentes para garantizar que la ruta por la que se mueva esté lo más similar posible a la planificada por el sistema de búsqueda de rutas.

- Para los trabajos con procesamiento de imágenes hay que tener en cuenta que la iluminación tiene que ser uniforme a lo largo de todo el entorno para así garantizar que las imágenes adquiridas sean similares al entorno real.

REFERENCIAS BIBLIOGRÁFICAS.

6.1. BIBLIOGRAFÍA

[1] T. Arai, P. Enrico y L. E. Parker, «Advances in Multi-Robots Systems,» *IEEE Transactions on Robotics and Automation,* pp. 665-661, 2002.

[2] U. Y. Cao, A. S. Fukunaga y A. Kahng, Cooperative Movile Robotics: Antecedents and Directions., Boston: R.C. Arkin and G.A. Bekey, 1997.

[3] J. P. Desai, J. P. Ostrowski y R. Vijay Kumar, «Modeling and Control of Formations of Nonholonomic Movile Robots,» *IEEE Transacttions,* 2001.

[4] T. Kar-Han y M. Lewis, «Virtual structures for high precision cooperative movile robotic control,» *IEEE International Conference,* pp. 4-8, 1996.

[5] M. Mataric, M. Nilsson y K. Simarin, «Cooperative multi-robot box-pushing, in Proc.,» *IEEE Conference,* pp. 556-561, 1995.

[6] E. A. Beltrán Andrés, «Transporte Cooperativo de Objetos con una plataforma móvil en un entorno estructurado,» *Redes de ingeniería,* pp. 12-22, 2012.

[7] «Robo Cup,» 2015. [En línea]. Available: http://www.robocup2015.org/. [Último acceso: 22 11 2015].

[8] RoboCup, 2016. [En línea]. Available: http://robocupssl.cpe.ku.ac.th/.

[9] Wikipedia, Febrero 2016. [En línea]. Available: http://wiki.robocup.org/wiki/Small_Size_League.

[10] U. S. Tomás, 2015. [En línea]. Available: http://electronica.usta.edu.co/index.php/noticias-electronica/548-participacion-del-equipo-de-futbol-robotico-en-robocup-2015.

[11] U. T. F. S. María, 2014. [En línea]. Available: http://ingenium.usm.cl/noticias/futbol-robotico/.

[12] FIRA, 2015. [En línea]. Available: http://www.fira.net/contents/sub01/sub01_1.asp.

[13] E. Universo, 2002. [En línea]. Available: http://www.eluniverso.com/2002/04/22/0001/15/502CDF9A0B434C64BFF0A2848CDC02FB.html.

[14] C. A. Villaroel Samaniego, C. A. Calderón Garzozi y R. P. Carrollo, «Diseño e implementación de un equipo de robots autónomos que toman desiciones en tiempo real: Futbol Robótico - Componente Inteligente,» Guayaquil, 2003.

[15] D. Paillacho Chiluisa y V. Salamea Tituana, «Diseño e implementación de un equipo de robots autónomos que toman desiciones en tiempo real: Futbol Robótico - Componente Electromecánico,» Guayaquil, 2003.

[16] E. Dávila Ferri, «Un ecuatoriano ganó por goleada en la RoboCup 2015,» *El Comercio,* 2015.

[17] A. Lara , C. U. Urrea y K. A. Barbosa, «Controller design of Industrial Cooperative Robots,» de *Eléctrica, Ingeniería Electrónica, Tecnologías de la Información y Comunicación (CHILECON), 2015 Conferencia sobre CHILENA*, Santiago, 2015.

[18] S. Kosuke, N. Akinori y L. Zhiwei, «Cooperative Stochastic Position Estimation of Multiple Mobile,» de *System Integration (SII), 2013 IEEE/SICE International Symposium on*, Kobe, 2013.

[19] N. Kubota, . Y. Toda y S. Suzuki, «Cooperative Formation of Multi-robot Based on Spring Model,» de *Robot, Vision and Signal Processing (RVSP), 2013 Second International Conference on*, Kitakyushu, 2013.

[20] G. Eoh, J. Jeon, . J. Choi y . B. Lee, «Multi-robot cooperative formation for overweight object transportation,» *System Integration (SII), 2011 IEEE/SICE International Symposium on,* pp. 726 - 731, Dec 2011.

[21] J. Acevedo, B. Arrue, I. Maza y A. Ollero, «Cooperative perimeter surveillance with a team of mobile robots under,» *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on,* pp. 5067 - 5072, Nov. 2013.

[22] M. R. Modesti y S. P. Mateus, «Sistemas de guiado para colaboración en navegación de,» [En línea]. Available:

http://www.investigacion.frc.utn.edu.ar/sensores/Aplicaciones/Publicaciones/032.pdf.

[23] A. Rosales Acosta, «Control Dinámico de Sistemas Robóticos Móviles, Un enfoque basasdo en el Algebra Lineal,» San Juan, 2009.

[24] M. Á. Fernández Lancha, D. Fernández Sanz y C. Valmaseda Plasencia, Planificación de Trayectorias para un Robot Móvil, Madrid: Universidad Computense, 2009.

[25] S. Cameron, «Obstacle avoidance and path planning,» *Industrial Robot an International Journal,* pp. 9-14, 1994.

[26] J. C. Latombe, «Robot motion planning,» *Kluwer Academic Publishers,* 1991.

[27] A. Yandún y N. Sotomayor, «Planeación y seguimiento de trayectorias para un robot móvil,» Quito, 2012.

[28] M. Fernández, . D. Fernández D y C. Valmaseda, «Planificación de trayectorias para un Robot Móvil,» Madrid, 2009.

[29] A. Barrientos, Fundamentos de Robótica, 2 ed., I. D. ESPAÑA, Ed., Barcelona: S.A. MCGRAW-HILL, 2007.

[30] P. Arévalo y S. Herrera, Robótica Industrial Prototipo y Sistema de Visión Artifical, E. A. Española, Ed., Editorial Académica Española, 2012.

[31] directindustry, «directindustry,» 2015. [En línea]. Available: http://www.directindustry.es/. [Último acceso: 2015].

[32] electrotec, «electrotec,» 2015. [En línea]. Available: http://www.electrotec.pe/. [Último acceso: 2015].

[33] Aldebaran, «Aldebaran,» 2015. [En línea]. Available: https://www.aldebaran.com/en/humanoid-robot/nao-robot. [Último acceso: 2015].

[34] B. dynamics, «Boston dynamics,» 2015. [En línea]. Available: http://www.bostondynamics.com/robot_bigdog.html. [Último acceso: 2015].

[35] irobot, «irobot,» 2015. [En línea]. Available: http://www.irobot.com/. [Último acceso: 2015].

[36] NayLamp, 2015. [En línea]. Available: http://www.naylampmechatronics.com/robotica/105-plataforma-robot-movil.html.

[37] K. &. Co, 2014. [En línea]. Available: http://www.kwapil.com/produkte/maxon/dcmotor/maxon-re-max.php.

[38] M. F. Rodriguez Borja y S. D. Sandobalín Guaman, «Diseño y Construcción de tres mini robots exploradores cooperativos,» Quito, 2013.

[39] Significados, «Significados,» 2013. [En línea]. Available: http://www.significados.com/inteligencia/. [Último acceso: 2015].

[40] D. conceptos, «De conceptos,» 2015. [En línea]. Available: http://deconceptos.com/ciencias-sociales/artificial. [Último acceso: 2015].

[41] Natureduca, «Natureduca,» 2015. [En línea]. Available: http://www.natureduca.com/anat_funcrelac_sentidos2.php. [Último acceso: 2015].

[42] I. D. García Santillán, Visión Artificial y Procesamiento Digital de Imágenes usando Matlab, Ibarra, 2008.

[43] R. Molina, Introducción al Procesamiento y Análisis de Imágenes Digitales, D. d. C. d. I. C. e. IA., Ed., Granada, 1998.

[44] R. M. Haralick y S. Linda G., Computer and Robot Vision, Addison-Wesley, 1992.

[45] K. FU, R. González y G. Lee, Robótica, Control, Detección, Visión e Inteligencia, Madrid: Mc Graw-Hill, 1990.

[46] M. Á. Fernández Lancha, D. Fernández Sanz y C. Valmaseda Plasencia, «Planificacion de trayectorias para un Ronot Móvil,» Madrid, 2010.

[47] Y. Ronald R. y D. P. Filev, Essentials of Fuzzy Modeling and Control, NY: Wiley Interscience, 1994.

[48] E. Zamora, 2015. [En línea]. Available: https://www.youtube.com/watch?v=xLFNUo0mTZE&list=PLIyIZGa1sAZoWAeT_tL7zCv3wi1ISrBa0.

[49] A. Sanjay Krishnankutty . [En línea]. Available: http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/mamdani1.htm.

[50] I. Pérez y B. León, Logica Difusa para principiantes, Caracas: Texto, C.A., 2007.

[51] E. 5Hertz, «Tienda 5 Hertz electrónica,» 2015. [En línea]. Available: http://5hertz.com/index.php?main_page=product_info&products_id=415. [Último acceso: 2015].

[52] ifuture, «Tienda iFuture,» 2015. [En línea]. Available: http://www.ifuturetech.org/product/dc-stepper-motor-drive/. [Último acceso: 2015].

[53] Sengoku, «Tienda Sengoku,» 2015. [En línea]. Available: http://sengoku-akb.jugem.jp/?eid=7. [Último acceso: 2015].

[54] GeekBot, «GeekBot,» 2015. [En línea]. Available: http://www.geekbotelectronics.com/producto/qrd1114/. [Último acceso: 2015].

ANEXOS

7.1. ANEXO 1: MANUAL DE USUARIO.

En este manual encontrará la forma de uso del software para el control de la operación del sistema cooperativo de robots para el transporte de un objeto, para lo cual debe seguir los siguientes pasos:

1. Ejecute Matlab, ya sea desde el ícono de acceso del escritorio o el botón de inicio.

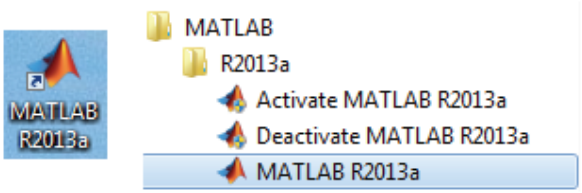


Figura 1. Acceso al Software.

2. Ubique en el path la carpeta que contiene los archivos proporcionados para la ejecución, de acuerdo a donde haya copiado el contenido. Con esta acción le aparecerá en el explorador todos los archivos relacionados para la ejecución del sistema.

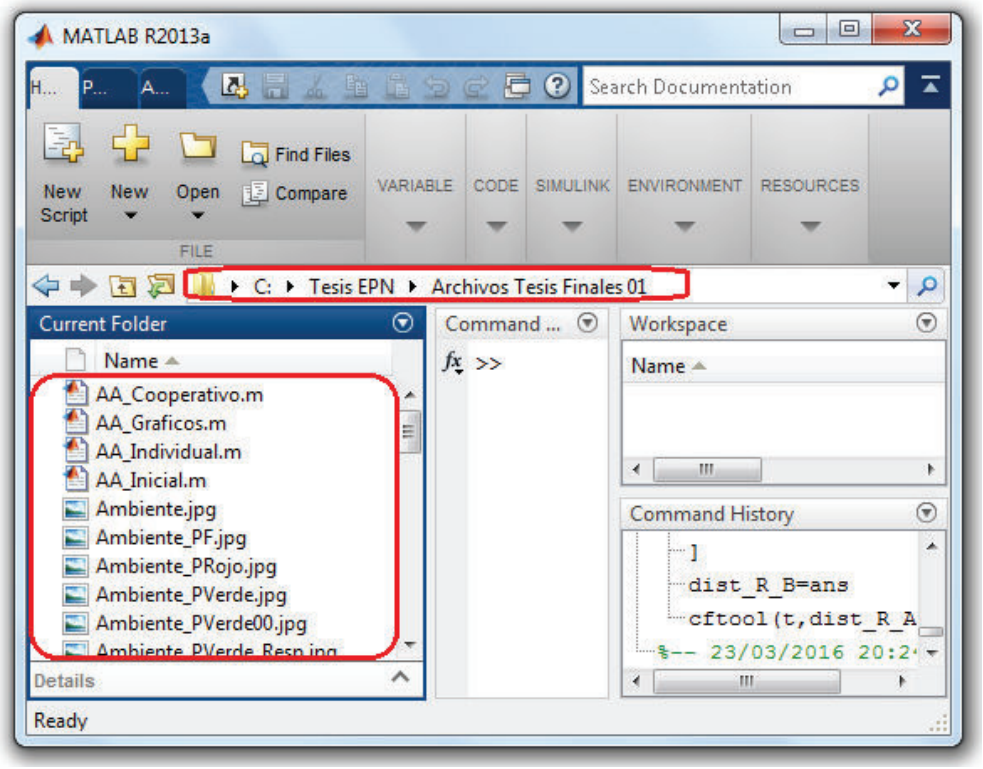


Figura 2. Entorno de trabajo

3. Teclee en la ventana de comandos y presione enter. Siga las instrucciones que se vayan desplegando conforme avanza en el desarrollo del software:

   >> AA_Inicial

   Conforme se desplieguen las preguntas vaya tecleando las letras de confirmación y presione la tecla enter.

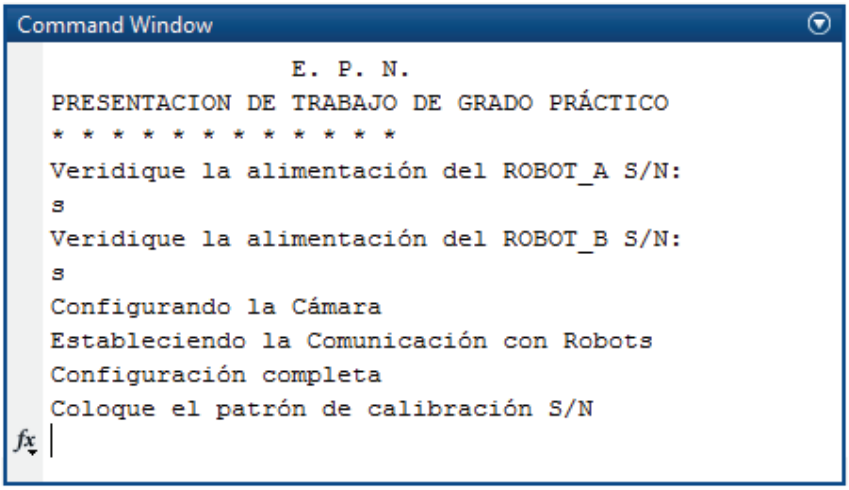


Figura 3. Ventana de comandos.

El sistema le solicitará que verifique la alimentación de los robots, para lo cual deberá encender los interruptores de control y de potencia, estos se muestran en la Figura 4, están en las mismas posiciones para los dos agentes. Mirándolos desde la parte posterior a la derecha el interruptor de alimentación de control y a la izquierda la alimentación del circuito de potencia, deben ser encendidos de acurdo a como han sido nombrados, primero control y después potencia.

Figura 4. Interruptores de encendido

Para verificar la alimentación los robots están provistos de luces indicadoras de color verde mostradas en la Figura 4.

Lo siguiente es encuadrar la cámara, para lo cual aparecerá una ventana la misma de le permitirá hacer los ajustes físicos necesarios para que la cámara se oriente hacia el espacio y su eje de lente esté ortogonal al espacio.

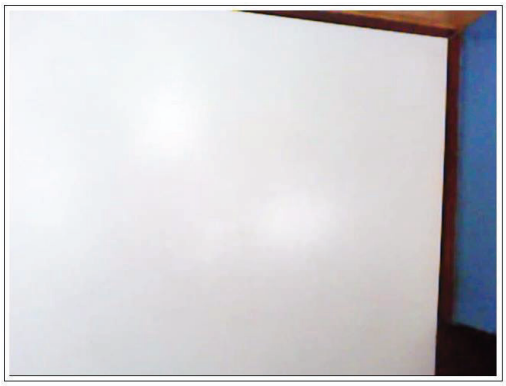


Figura 5. Cámara Descuadrada

Posterior a eso el sistema establecerá la comunicación con cada uno de los agentes, en este caso en la parte posterior de los mismos estarán titilando unas luces de color rojo en los módulos HC06, cuando la comunicación esté lista las luces dejarán de parpadear, esto se aprecia en la Figura 6.

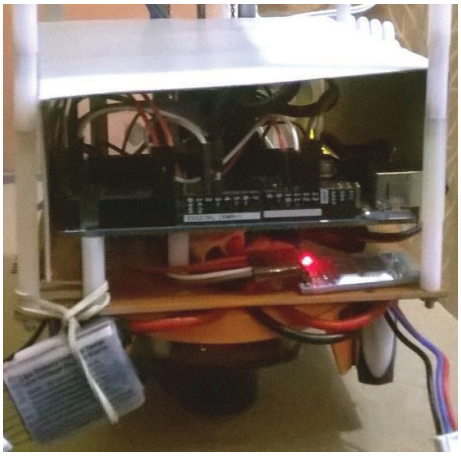


Figura 6. Módulo Bluetooth.

Lo siguiente es colocar el patrón de calibración para hacer las mediciones iniciales que servirán en el futuro para los cálculos del sistema.

Figura 7. Patrón de calibración.

Una vez colocado el patrón de calibración y realizados los cálculos por parte del sistema en la ventana de comandos seguirán apareciendo más ordenes, como lo muestra la Figura 8. En la que se pueden apreciar los valores los valores calculados para la distancia entre los robots, así como también la distancia de los lados de los cuadrados en el patrón de calibración.

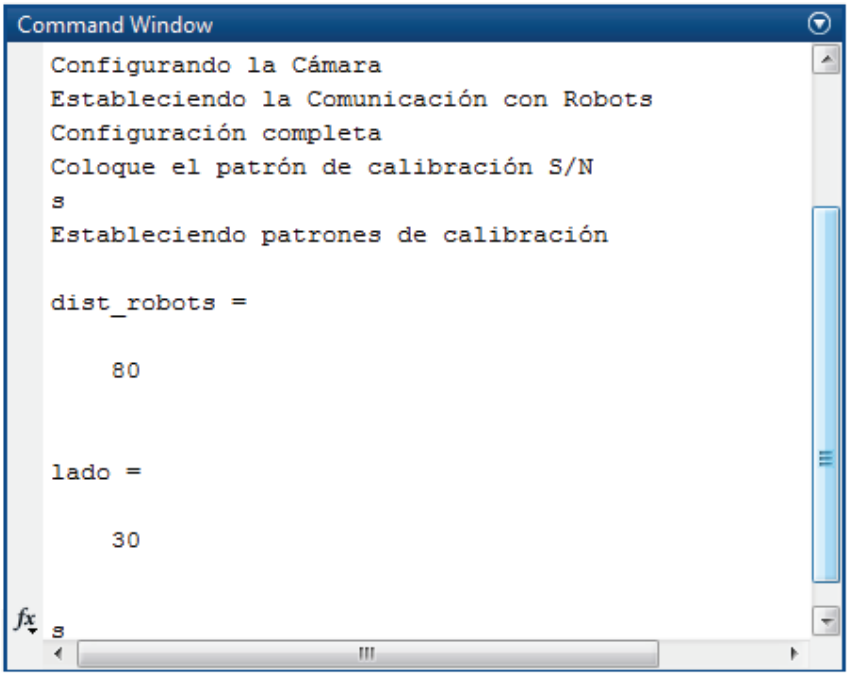


Figura 8. Ventana de comandos.

Lo siguiente es colocar el entorno para que el sistema tenga una apreciación inicial del mismos para que con esta información realice operaciones de ubicación de puntos y compresión de la imagen en los pasos siguientes.

El sistema solicita el armado del entorno, esto se ve en la Figura 9.

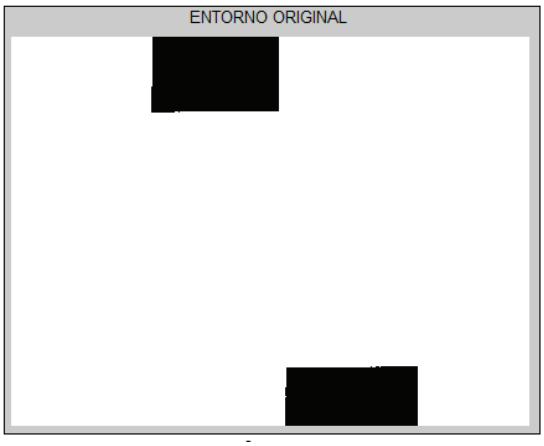


Figura 9. Entorno inicial

Los siguientes mensajes serán los que aparecen en la Figura 10, ante los mismos hay que hacer las mismas acciones descritas anteriormente, es decir confirmar con la letra y presionar la tecla enter.

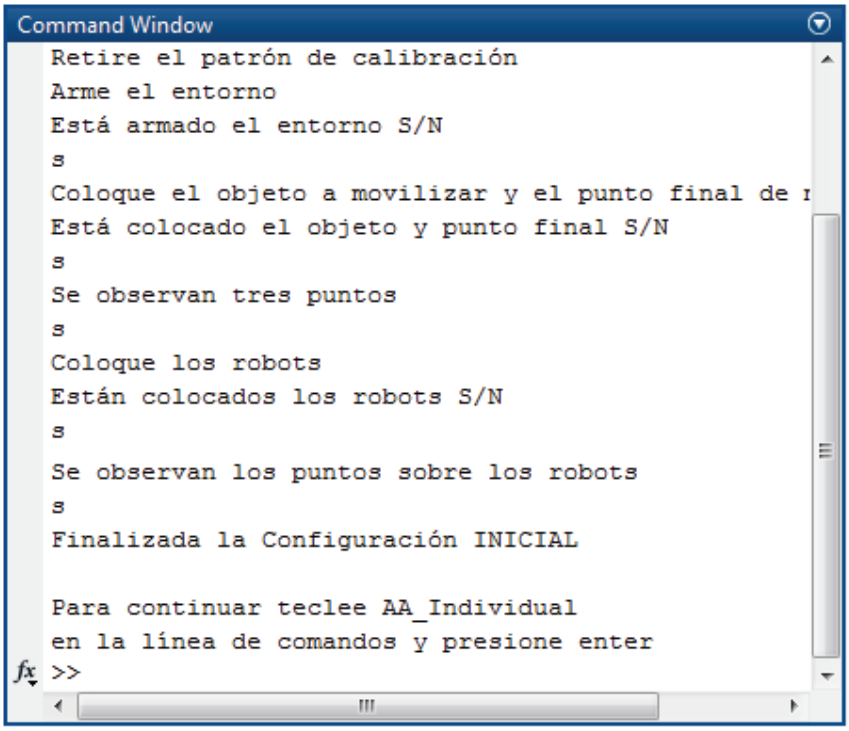


Figura 10. Ventana de Comandos.

Los siguiente que solicitará el sistema es la inclusión del objeto, así como también del punto final para ubicarlos dentro del entorno, con los datos obtenidos mediante filtrado y búsqueda de plantillas el sistema tendrá los puntos finales del movimiento individual de los robots. esto se parecía en la Figura 11.

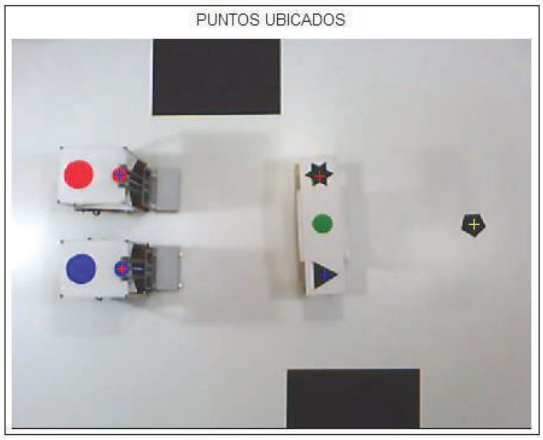


Figura 11. Puntos ubicados dentro del entorno.

4. El siguiente comando a ejecutar es:

   >> AA_Individual

   Con este comando se ejecutan las siguientes acciones. Primero establece la reducción del entorno, ubica los puntos y los márgenes de espacio de tolerancia para el movimiento de los robots, esto se aprecia en la Figura 12.

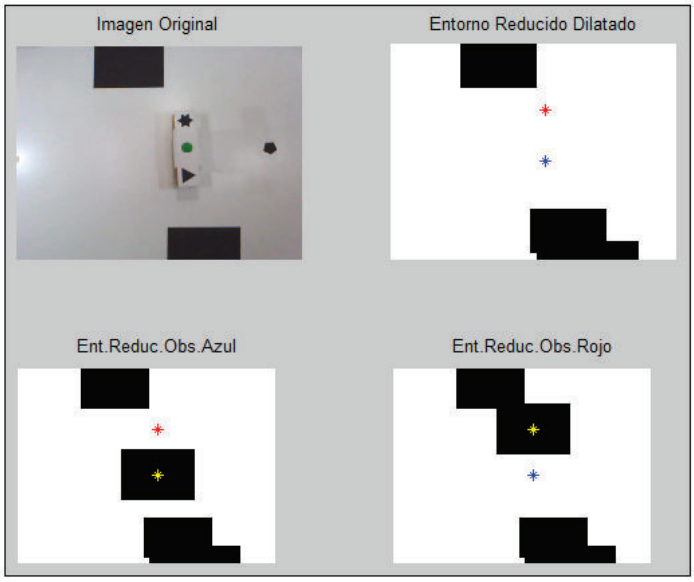
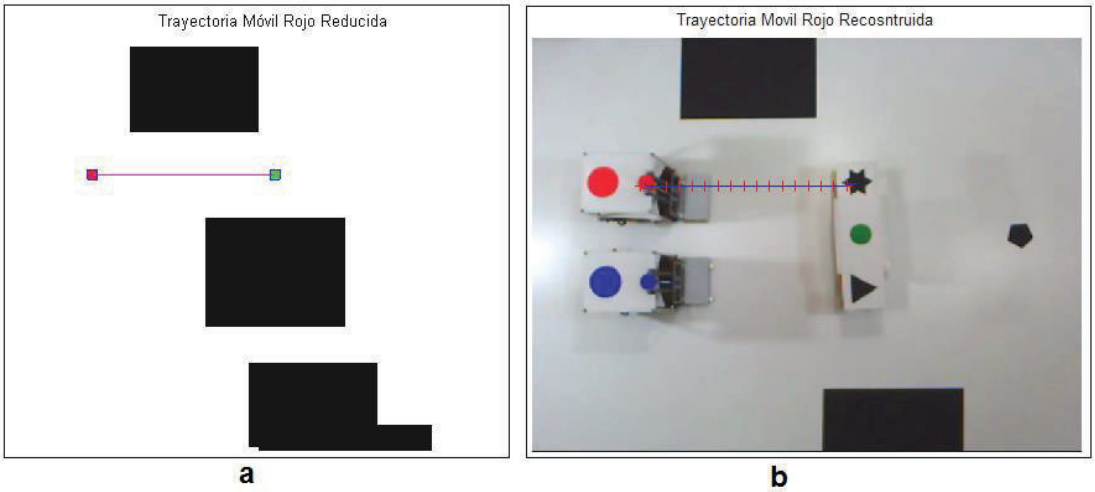


Figura 12. Reducción Dilatado y márgenes de movimiento de los robots.

Posterior a esto el sistema calcula la ruta en el entorno reducido para el Robot_A, reconstruye la ruta al entorno original y desplaza al robot a lo largo de la ruta calculada hasta llegar al objeto. Estas dos acciones las ejecuta también para el Robot_B.



a) Ruta en entorno reducido b) Ruta reconstruida en entorno

Figura 13: Rutas Robot_A

Las mismas imágenes mostradas en la Figura 13 para el Robot_A, aparecerán en el sistema, pero con el Robot_B.

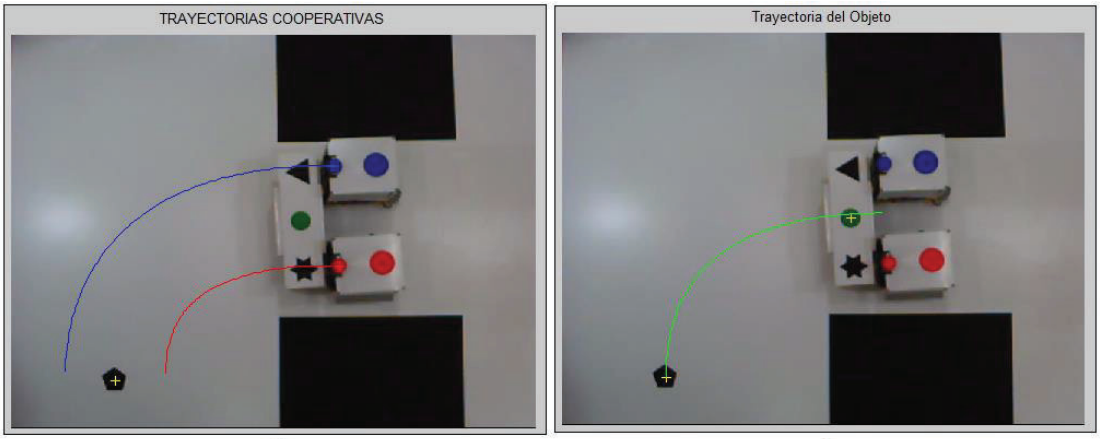
5. Una vez que los robots se encuentran enfrente del objeto se tiene que ejecutar el comando y presionar enter:

   >> AA_Cooperativo

   El sistema da la orden de elevar el objeto a los robots de uno en uno, de esta forma se tiene una primera acción cooperativa. Esto se ve en la Figura 14.

Figura 14. Objeto elevado por los agentes.

Con el objeto ya elevado, se calcula las rutas para cada robot y para el objeto. Esto se ven en la Figura 15.



a b

a) Rutas de los robots cooperativos b) Ruta del objeto

Figura 16. Rutas Cooperativas.

Con las nuevas Rutas se ejecuta el movimiento cooperativo de los robots de uno en uno a la vez hasta llegar al punto final o punto de destino para el objeto

6. Con lo hasta aquí expuesto de tendría terminado el movimiento cooperativo, únicamente restaría graficar los datos obtenidos, para lo cual deberá escribir: >> AA_Graficos. Y presionar enter. Con lo cual aparecerán las imágenes de la Figura 17, para su análisis.

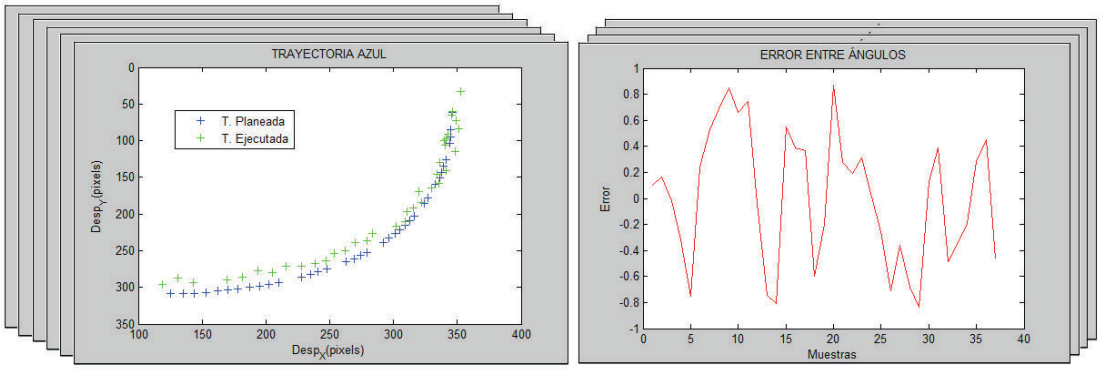
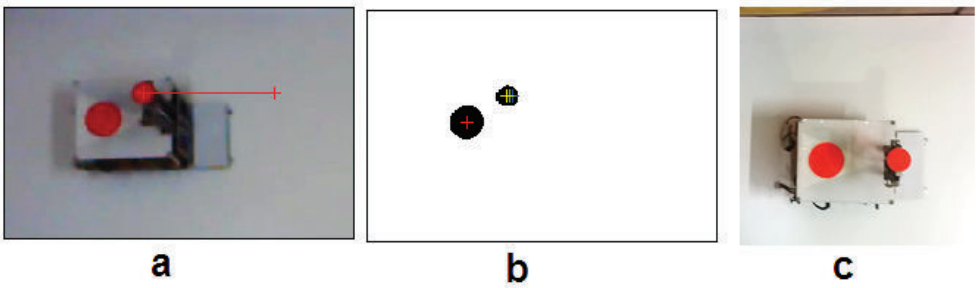


Figura 17. Gráficas del sistema.

7.2. ANEXO 2: ESTUDIO PARALAJE.

Estudio de paralaje de las Imágenes adquiridas

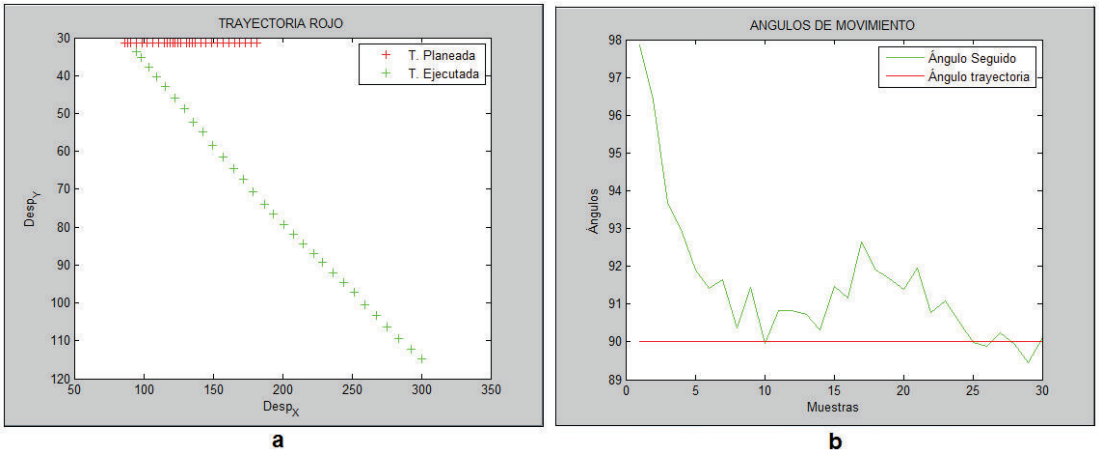
Se ha generado una ruta virtual de izquierda a derecha en la parte superior del entorno.



a) Ruta virtual b) Angulos de movimiento c) Orientacion en entorno

Figura 1: Imágenes de ruta horizontal superior

En la Figura 1, se puede observar la toma inicial realizada por la cámara cenital, en donde se ha trazado la ruta virtual, allí se puede observar que se da que la ruta a seguir será a 90°, pero la orientación del móvil está formando un ángulo diferente, esto se puede apreciar de mejor manera en la Figura 1b, pero en realidad la orientación en el entorno es el que se muestra en la Figura 1c, entonces se aprecia que el Robot_A está paralelo al borde del entorno. Razón por la que con la disposición que se tiene de las marcas, el sistema arrancará con un error.



a) Ruta del Robot_A b) Ángulos del Robot_A

Figura 2: Ruta horizontal superior

De la Figura 2a, se puede analizar que el error inicial del ángulo del robot con respecto a su orientación real dentro del entorno se vuelve acumulativo, haciendo que la diferencia entre la posición del robot con respecto a la posición que debería tener se hace cada vez más grande. Por otra parte, se puede apreciar en la Figura 2b, que el controlador está operando en buena forma, debido a que la diferencia entre el ángulo de la ruta con respecto al ángulo de robot, cada vez se hace menor, de tal forma que el error entre los dos se va minimizando hasta hacerse cero, que correspondería cuando el robot está siguiendo la ruta planificada.

A lo largo del desarrollo de esta prueba sucedió un caso interesante en el que las marcas llegaron a fusionarse haciendo que el sistema pierda totalmente su orientación y generando un error dentro de la ejecución, este caso de la fusión de las dos marcas de orientación se aprecia en la Figura 3.

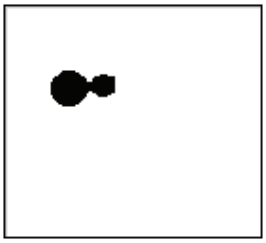
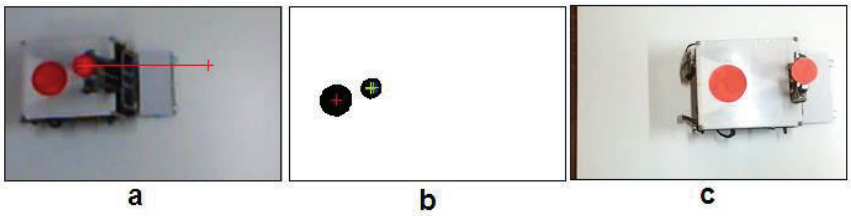


Figura 3: Imagen de las dos marcas de orientación fusionadas

Para corroborar esta información se realiza una nueva prueba para la cual se establece un movimiento horizontal de izquierda a derecha en la parte central del espacio de trabajo, obteniendo las imágenes mostradas en la Figura 4.

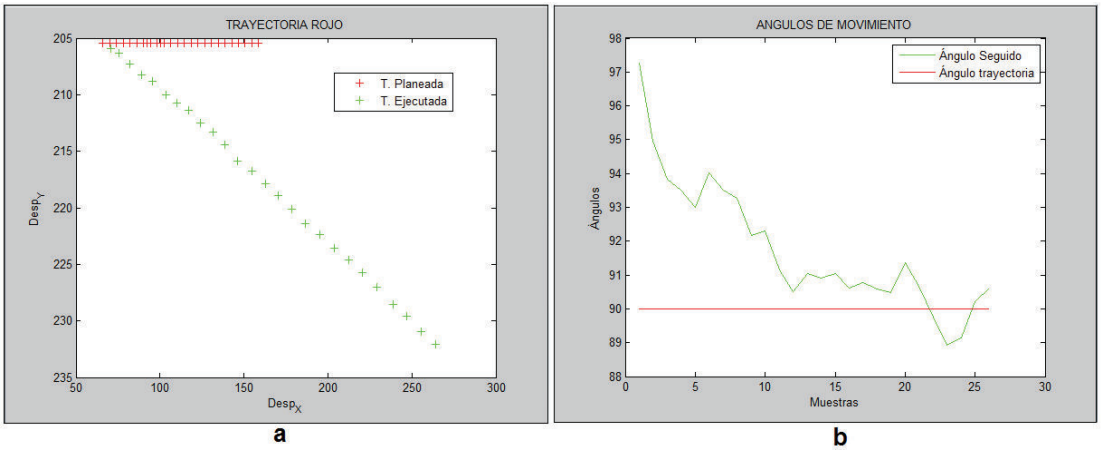


a) Ruta virtual b) Angulos de movimiento c) Orientacion en entorno

Figura 4: Imágenes ruta horizontal central

En la Figura 4 se puede apreciar que los mismos problemas de error inicial del ángulo del Robot_A se va repitiendo igual que en la prueba horizontal superior, a

pesar que la orientación del agente sea paralela al borde izquierdo, tomado con la parte posterior del agente.

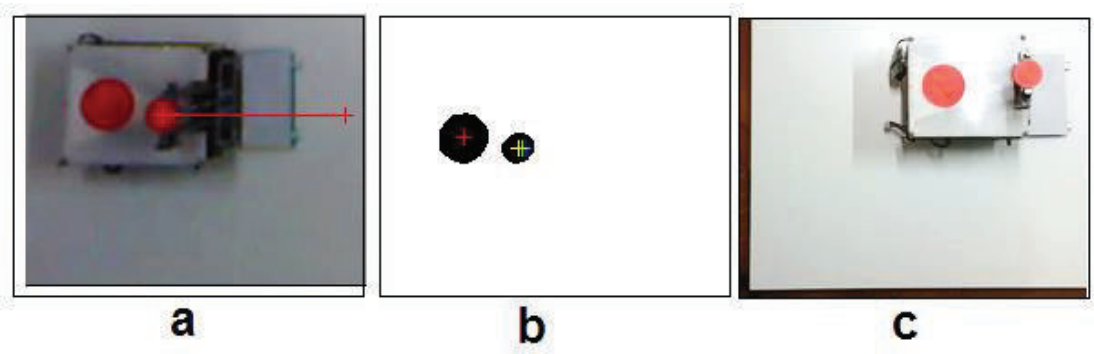


a) Ruta del Robot_A b) Ángulos del Robot_A

Figura 5: Ruta horizontal media

La respuesta obtenida en esta prueba es similar a la obtenida de la experiencia anterior, para terminar de cubrir todo el lado izquierdo del entorno será necesario reperir la prueba con una ruta horizontal inferior.

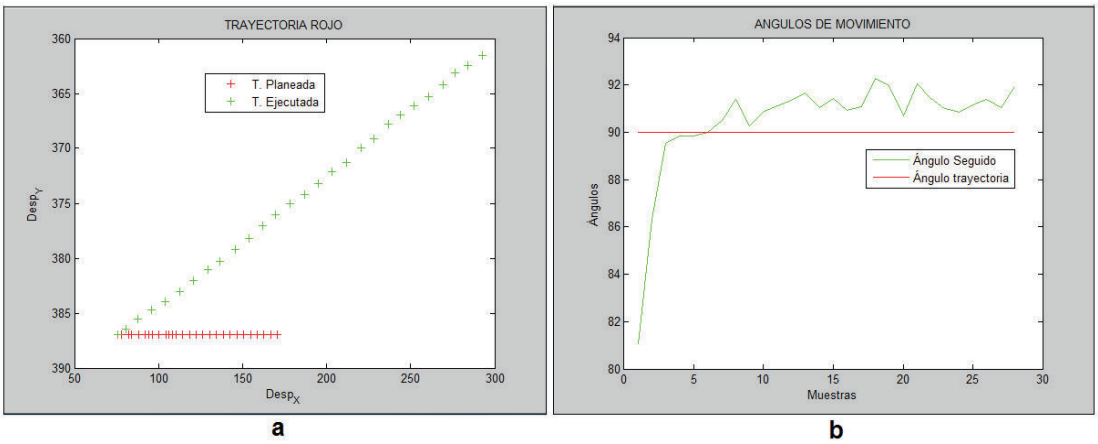
Se podría analizar y deducir que se puede obtener una respuesta similar, debido a las dos pruebas anteriores, pero hay que estar seguro de lo que se puede concluir de las experiencias, razón por la que se realiza esta nueva experiencia.



a) Ruta virtual b) Angulos de movimiento c) Orientacion en entorno

Figura 6: Imágenes ruta horizontal inferior

En las condiciones iniciales del sistema se tienen los mismos resultados que las pruebas anteriores, ahora con respecto a las rutas y a los ángulos se tiene como respuestas las mostradas en la Figura 7.

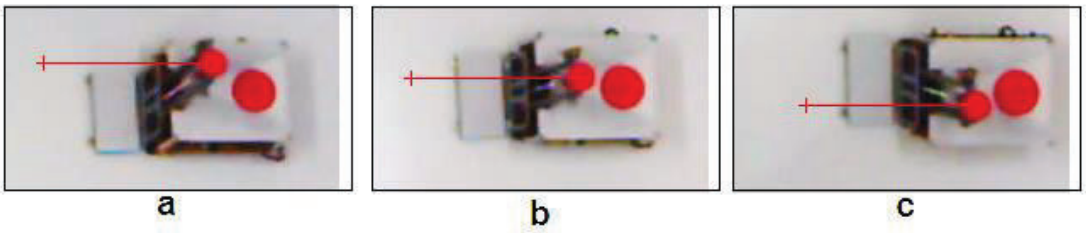


a) Ruta del Robot_A b) Ángulos del Robot_A

Figura 7: Ruta horizontal inferior

De las tres pruebas hasta ahora realizadas se han obtenido resultados similares, por consiguiente se determina que la ubicación de las marcas de orientación se encuentran en una mala posición, analizando el comportamiento de sistema.

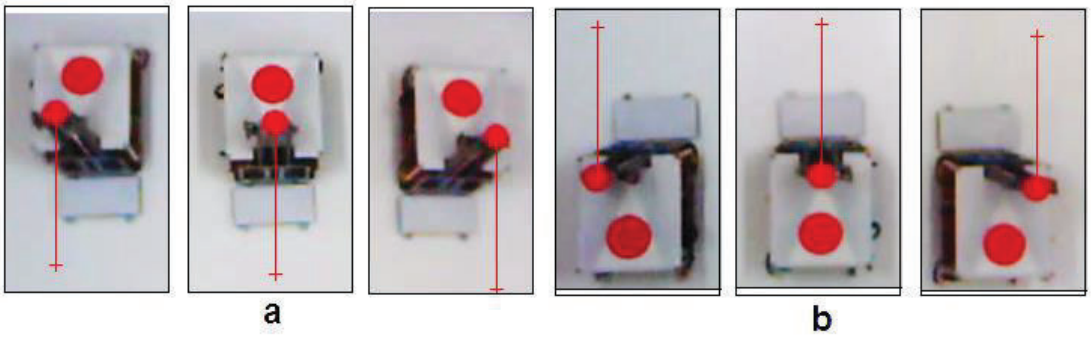
Para generalizar esta situación en todo el entono se toman imágenes de puntos de partida en todo el espacio de trabajo en el lado derecho del entorno, esto se aprecia en la Figura 8. A simple vista se puede apreciar que existe un error tanto en la toma superior, Figura 8a, en la central, Figura 8b o en la inferior 8c, ya que el ángulo esperado de orientación del Robot_A sería de -90°, pero son totalmente diferentes al valor esperado.



a) Toma extremo derecho superior b) Toma extremo derecho central c) Toma extremo derecha inferior.

Figura 8: Imágenes extremo derecho

Hasta aquí en forma horizontal para el desplazamiento, pero el movimiento de los dos robots es en todos los sentidos, se realizan tomas para un desplazamiento vertical en el entorno.



a) Desplazamiento vertical arriba hacia abajo tomas izquierdo, dentral y derecho b) Desplazamiento vertical abajo hacia arribao tomas izquierdo, dentral y derecho

Figura 9: Tomas iniciales movimiento vertical de los robots

De la Figura 4-9, tanto a como 4b, se aprecia que existe error en el ángulo inicial, excepto en las tomas centrales en donde sí coinciden los ángulos, pero hay que recordar que los dos robots se tienen que desplazar por todo el entorno y en todas las direcciones, por consiguiente, no hay como particularizar.

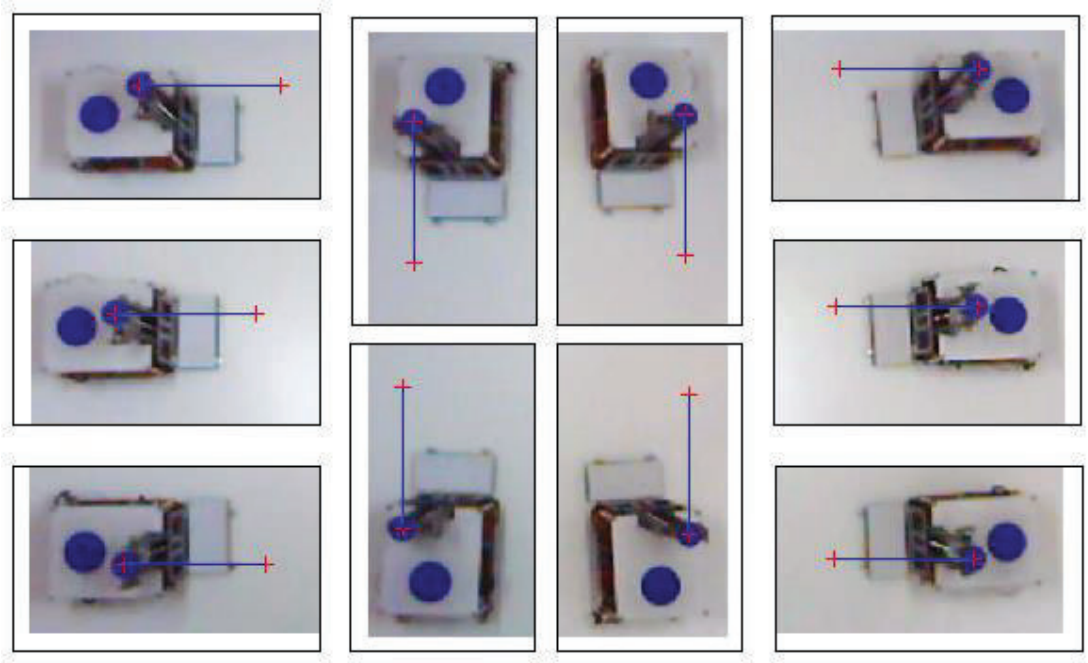


Figura 10: Tomas iniciales de Robot_B

Hay que descartar todo tipo de posibilidad de fallo por consiguiente debido a que se tienen dos robots cada uno con diferentes colores de marcas, se analiza la toma inicial del robot con marcas de orientación de color azul, obteniendo la Figura 10, en la misma se puede observar que el error de la toma inicial es la mismo que el robot con marcas rojas, por consiguiente se puede determinar que las posiciones en las que se encuentran las marcar de orientación ubicadas están mal para cada uno de los robots.

Ahora se tiene que analizar qué es lo que sucede con el sistema, revisando información sobre visión se ha encontrado un error similar que se dan en los sistemas de medición este es denominado “paralaje”, esto sucede en los sistemas de medición de pluma y consiste en que dependiendo de la posición del observado se obtienen medidas diferentes, razón por la que las tomas iniciales difieren de la realidad de la posición de los agentes dentro del entorno de trabajo.

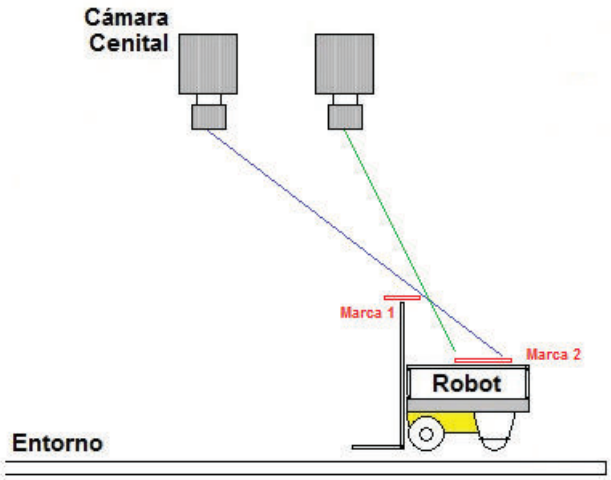
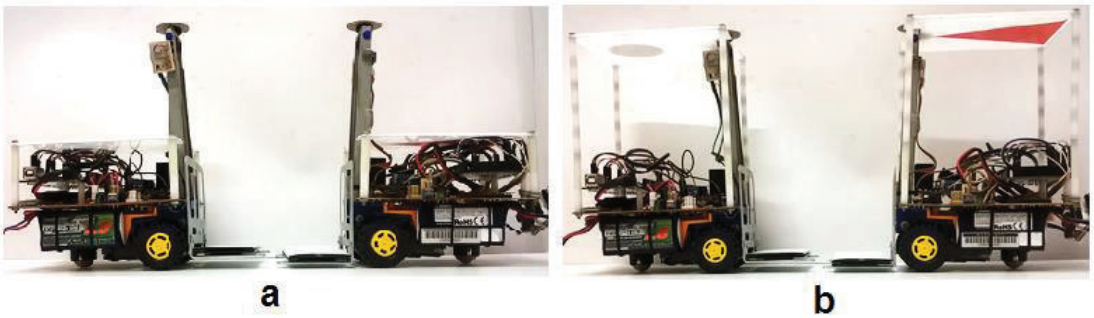


Figura 11: Falla de la toma inicial

Esta condición de la ubicación de las marcas permite comprender el caso particular de la fusión de las dos imágenes en las tomas como se muestra en la Figura 3.

Entonces la solución a este problema será el tener las dos marcas al mismo nivel para que la apreciación de la toma obtenida por la cámara cenital sea en el mismo plano de visión, Figura 12.

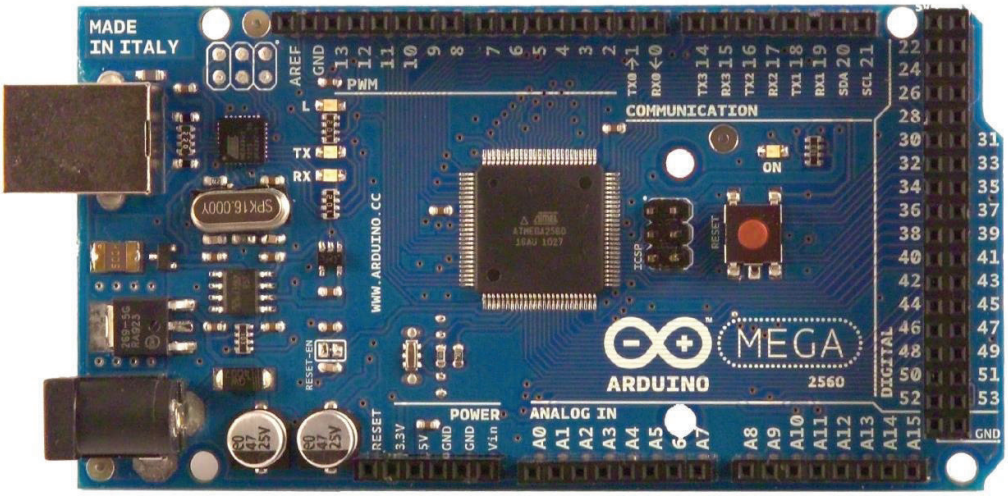


a) Robots con marcas originales b) Robots con marcas en el mismo plano

Figura 12: Robots con marcas de orientación modificadas

7.3. ANEXO 3: DATOS TÉCNICOS PLACA ARDUINO MEGA.

Arduino MEGA 2560



CE

Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications — Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials — Page 6

Terms & Conditions — Page 7

Enviromental Policies
half sqm of green via Impatto Zero® — Page 7

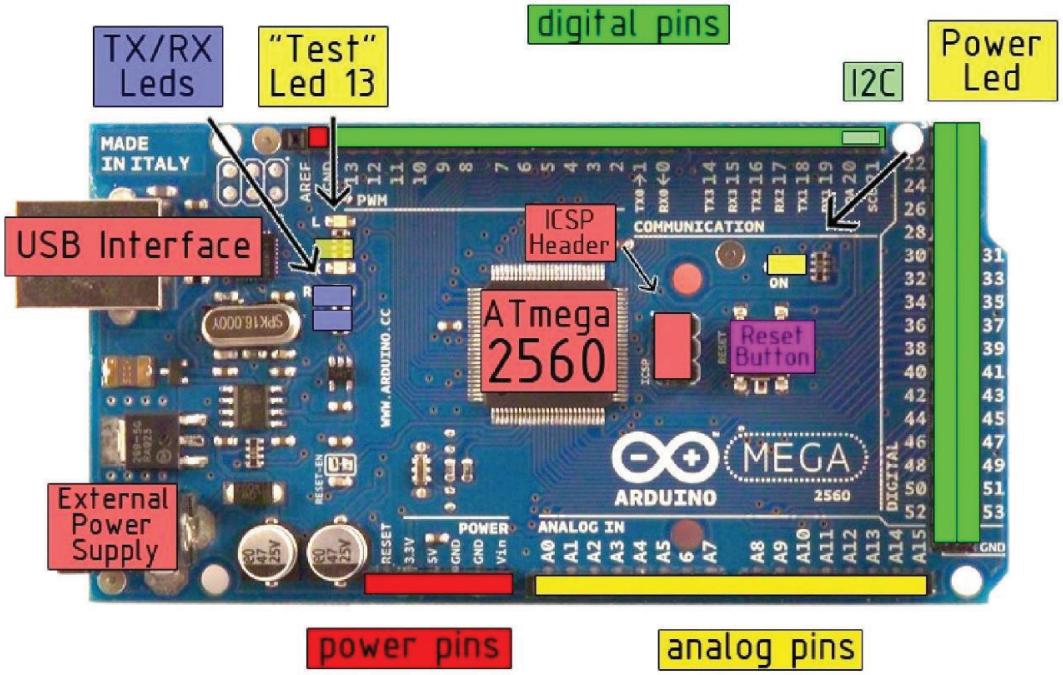
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

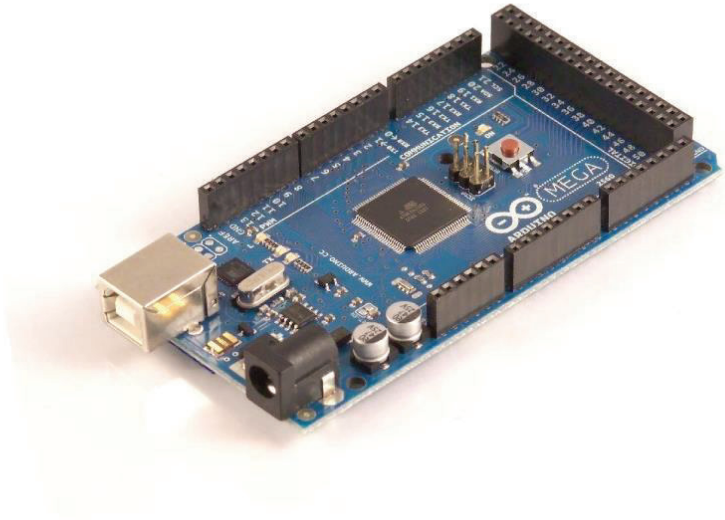
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.



Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I^2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

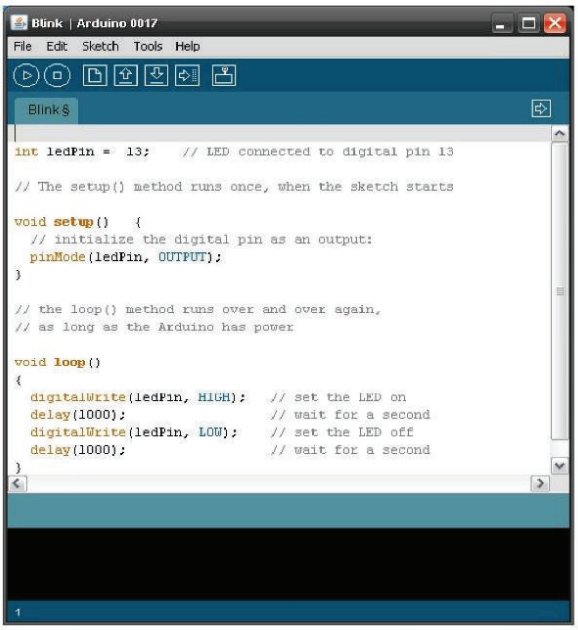
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

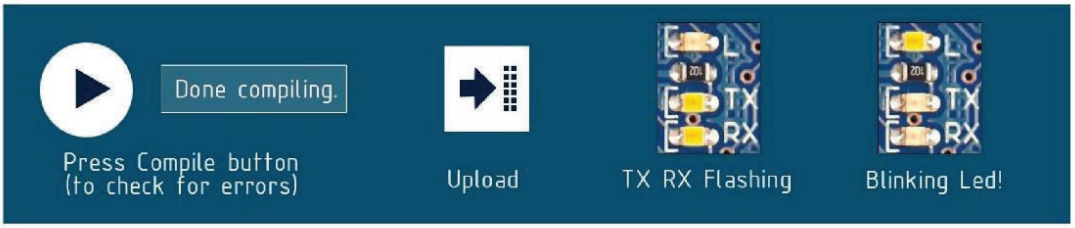
File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

Once you have your skecth you'll see something very close to the screenshot on the right.

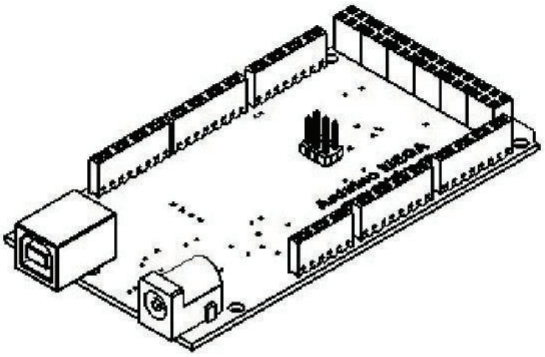
In **Tools>Board** select MEGA

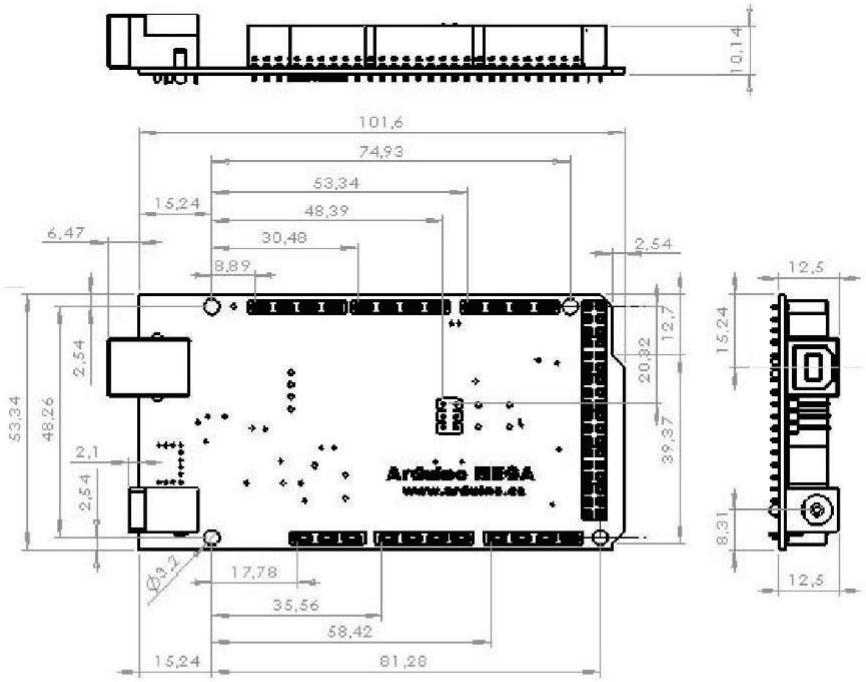
Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.





Dimensioned Drawing





Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

7.4. ANEXO 4: DATOS TÉCNICOS DRIVER DE HORQUILLA.

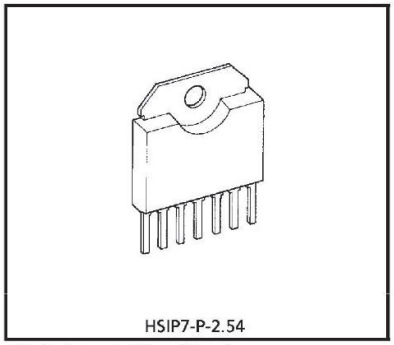
TOSHIBA TA7257P

TOSHIBA BIPOLAR LINEAR INTEGRATED CIRCUIT SILICON MONOLITHIC

TA7257P

BRIDGE DRIVER

The TA7257P is a Full Bridge Driver for blashed DC Motor Rotation control.
Forward Rotation, Reverse Rotation, Stop and Braking operations are available.
It's designed for Loading and Reel Motor driver for VTR and Tape Deck, and any other consumer and industrial applications.
TA7257P have Operation Supply Voltage terminal and Motor Driving Supply Voltage terminal independently therefore Servo control operation is applicable.

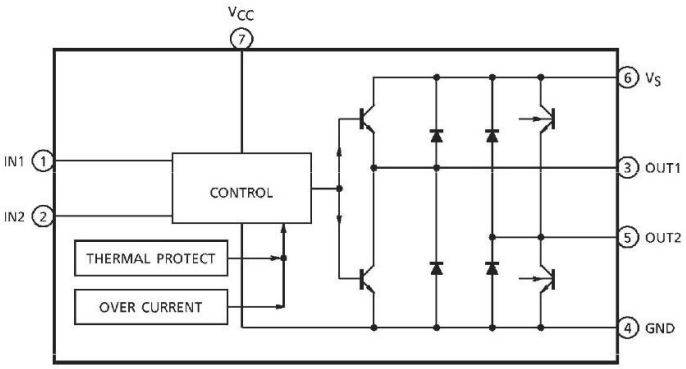


Weight : 1.88 g (Typ.)

FEATURES

- Output Current Up to 1.5 A (AVE.), and 4.5 A (PEAK).
- 4 Function Modes (CW, CCW, STOP and Brake) are Controlled by 2 Logic Signals Fed Into 2 Input Terminals.
- Build in Over Current Protector and Thermal Shut Down Circuit.
- Operating Voltage Range : $V_{CC\,(opr.)}$ = 6~18 V, $V_{S\,(opr.)}$ = 0~18 V

BLOCK DIAGRAM



980910EBA1

- TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

PIN FUNCTION

PIN No.	SYMBOL	FUNCTIONAL DESCRIPTION
1	IN1	Input terminal
2	IN2	Input terminal
3	OUT1	Output terminal
4	GND	GND terminal
5	OUT2	Output terminal
6	V_S	Supply voltage terminal for Motor drive
7	V_{CC}	Supply voltage terminal for Logic

FUNCTION

IN1	IN2	OUT1	OUT2	MODE
1	1	L	L	Brake
0	1	L	H	CW / CCW
1	0	H	L	CCW / CW
0	0			Stop

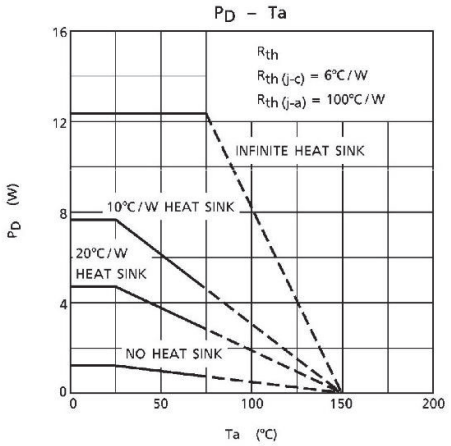
MAXIMUM RATINGS (Ta = 25°C)

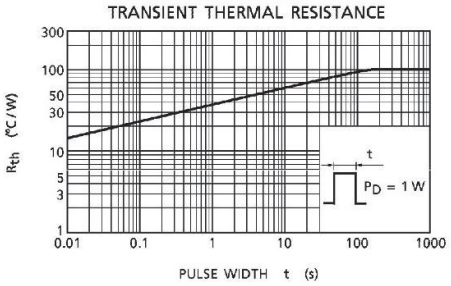
CHARACTERISTIC		SYMBOL	RATING	UNIT
Peak Supply Voltage	Peak	V_{CC} (MAX.)	25	V
	Operate	V_{CC} (opr.)	18	
Output Current	PEAK	I_O (PEAK)	4.5	A
	AVE.	I_O (AVE.)	1.5	
Power Dissipation		P_D	12.5 (Note)	W
Operating Temperature		T_{opr}	−30~75	°C
Storage Temperature		T_{stg}	−55~150	°C

(Note) Tc = 75°C

ELECTRICAL CHARACTERISTICS (Ta = 25°C)

CHARACTERISTIC		SYMBOL	TEST CIR-CUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Supply Current		I_{CC1}	—	V_{CC} = 18 V Output OFF stop mode	—	6.5	13	mA
		I_{CC2}		V_{CC} = 18 V Output OFF CW / CCW mode	—	10	20	
Saturation Voltage	Upper	$V_{S1}U$	—	V_{CC} = 18 V, I_O = 0.1 A	—	0.7	1.0	V
	Lower	$V_{S1}L$			—	0.6	0.9	
	Upper	$V_{S2}U$		V_{CC} = 18 V, I_O = 1.1 A	—	1.0	1.4	
	Lower	$V_{S2}L$			—	0.9	1.3	
Output Transistor Leakage Current	Upper	I_L U	—	V_S = 18 V	—	—	100	μA
	Lower	I_L L			—	—	100	
Input Voltage 1, 2		V_{IN} (H)	—	T_j = 25°C, pin ① and pin ②	3.0	—	—	V
		V_{IN} (L)			—	—	0.8	
Diode Forward Voltage		V_F U	—	I_F = 1.0 A	—	2.0	—	V
		V_F L			—	1.25	—	
Limiting Current		I_{SC}	—	—	—	3.5	—	A
Input Current		I_{IN}	—	—	—	1	10	μA





APPLICATION NOTE

(1) Input circuit

Input circuit is shown in Fig.1. It's a "Low active" type voltage comparator that's one input connect to Input terminal (pin ①, or ②) and the other to built-in temperature compensated voltage reference (V_{TH} = 1.4 V Typ.)
If a voltage above $V_{IN\,(H)}$ fed into the Input Terminal that means "Logic 1" and less than $V_{IN\,(L)}$ or connect to GND means "Logic 0".

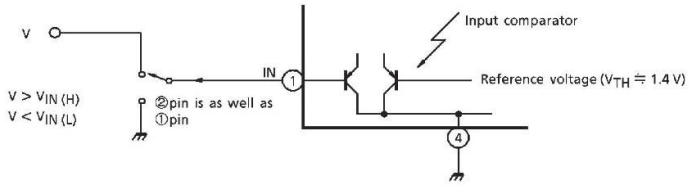


Fig.1

(2) Basic application circuit

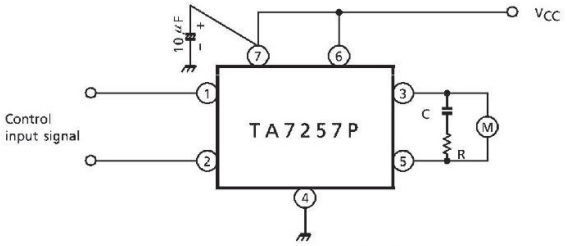


Fig.2

(Note 1) Fig.2 shows the basic application circuit.
Optimum values of the C, R depend on the inherent constant of a motor and parasitic C, R values around the circuit.
Normally, recommended to use 0.1 μF and 33 Ω.

(Note 2) Utmost care is necessary in the design of the output line, V_S, V_{CC} and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

(Note 3) Be careful when switching the input because rush current may occur.
When switching, stop mode should be entered or current limitation resister R should be inserted.

(Note 4) The IC functions cannot be guaranteed when turning power on of off.
Before using the IC for application, check that there are no problems.

(3) Additional diode

i) If the braking operation is so loose, connect a additional diode between each output to GND, (See Fig.3)

ii) If the back electromotive pulse generated in output coil is so strong.
Internally connected back electromotive suppression diode may be damaged by this pulse.
In such a case connect a additional diode between each output to V_{CC}. (See Fig.4)

iii) In case of mounted on radiators, do not use silicon rubber.

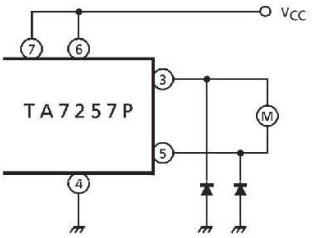


Fig.3

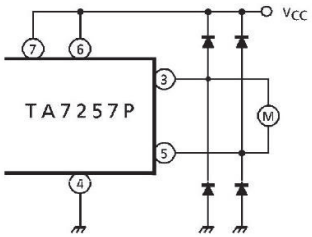
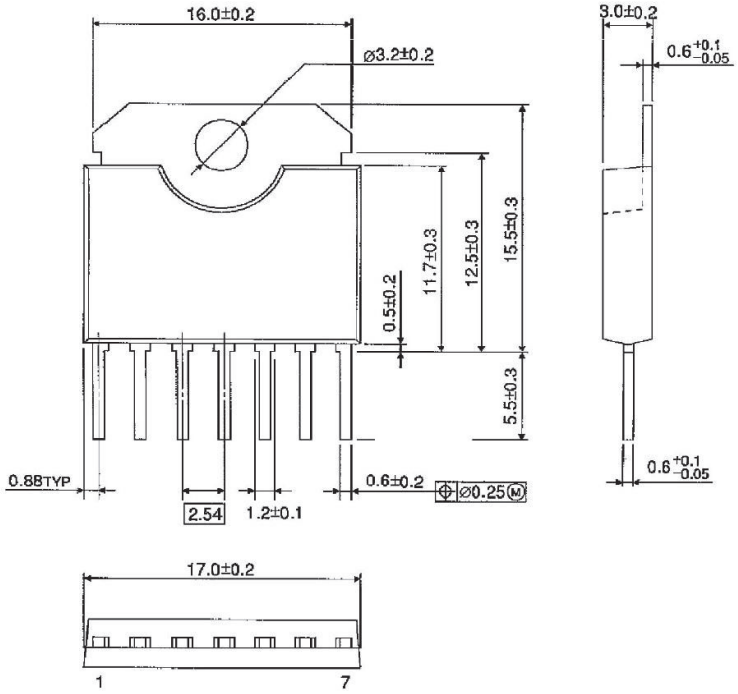


Fig.4

OUTLINE DRAWING

HSIP7-P-2.54

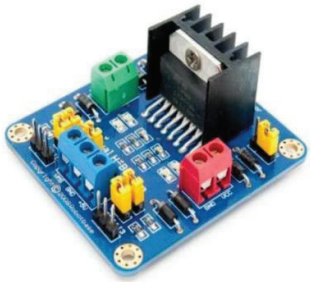
Unit : mm



Weight : 1.88 g (Typ.)

7.5. ANEXO 5: DATOS TÉCNICOS DRIVER DE MOTORES.

L298 Dual H-Bridge Motor Driver



Double H driver module uses ST L298N dual full-bridge driver. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors.

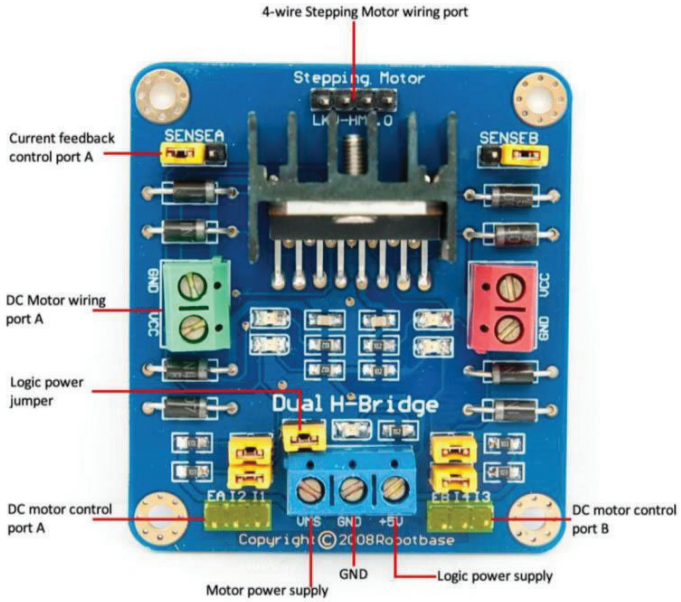
Features

- Light weight, small dimension
- Super driver capacity
- FWD protection
- Heavy load Heat sink
- Power selection switch
- 4 pull up resistor switch
- 2 DC motor/ 4 coil dual phrase stepper motor output
- Motor direction indication LED
- 4 standard mouting holes

Specifications

- Driver: L298
- Driver power supply: +5V~+46V
- Driver peak current: 2A
- Logic power output Vss: +5~+7V (internal supply +5V)
- Logic current: 0~36mA
- Controlling level: Low -0.3V~1.5V, high: 2.3V~Vss
- Enable signal level: Low -0.3V~1.5V, high: 2.3V~Vss
- Max drive power: 25W (Temperature 75 ℃)
- Working temperature: -25℃~+130℃
- Dimension: 60mm*54mm
- Driver weight: ~48g

Hardware Installation



Double H driver module can drive two DC motors at the same time.
Port A is completely symmetrical as port B on the board.

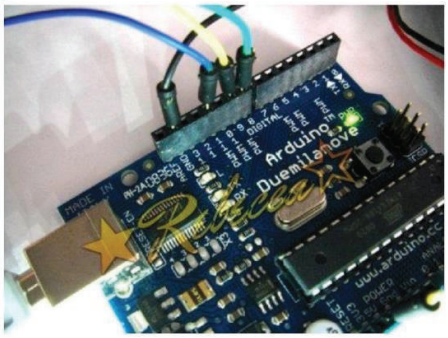
DC motor input port A has three pins, I1, I2 and EA. I1 and I2 are digital ports which are used to control the direction of motor, EA is connecting with PWM port of control board to control the speed of motor.
If I1=1and I2=0, the motor rotates clockwise.
If I1=0 and I2=1, it rotates anticlockwise.
If I1=I2， it stops rotating.

EA	I1	I2	Motor A status
》0	0	1	Clockwise rotation
》0	1	0	Anticlockwise rotation

Normally we do not need to connect wire to supply logic power.
Only if the motor power supply is +5V～+7V or +18V～+46V we have to take off the logic power jumper and connect 5v to supply logic power.
We can control 4-wire stepping Motor the same way as two DC motors with the signals from EA,I1,I2 and EB,I3,I4.

Arduino sample program

```
int pinI1=8;//define I1 port
int pinI2=9;//define I2 port
int speedpin=11;//define EA(PWM speed regulation)port
void setup()
{
  pinMode(pinI1,OUTPUT);//define this port as output
  pinMode(pinI2,OUTPUT);
  pinMode(speedpin,OUTPUT);
}
void loop()
{
  analogWrite(speedpin,100);//input a value to set the speed
  delay(2000);
  digitalWrite(pinI1,LOW);// DC motor rotates clockwise
  digitalWrite(pinI2,HIGH);
  analogWrite(speedpin,100);
  delay(2000);
  digitalWrite(pinI1,HIGH);// DC motor rotates anticlockwise
  digitalWrite(pinI2,LOW);
  analogWrite(speedpin,100);
  delay(2000);
  digitalWrite(pinI1,HIGH);// DC motor stop rotating
  digitalWrite(pinI2,HIGH);
  delay(2000);
}
```



Revision History

Rev.	Descriptions	Release date
1.1	Initial public release	30.06.2010

7.6. ANEXO 6: DATOS TÉCNICOS MÓDULO BLUETOOTH.

1. Product's picture

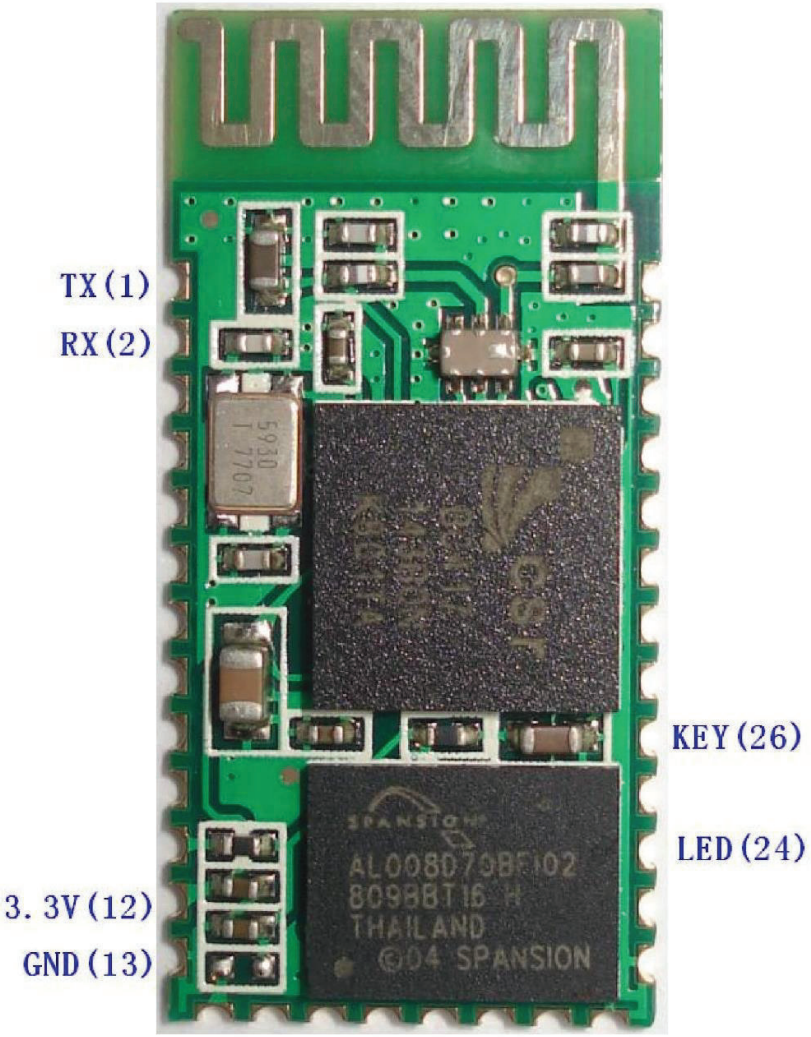


Figure 1 A Bluetooth module

2. Feature

- Wireless transceiver
  - Sensitivity (Bit error rate) can reach -80dBm.
  - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
  - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
  - Has a build-in 2.4GHz antenna; user needn't test antenna.
  - Has the external 8Mbit FLASH
  - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30～40mA. The current in communication is 8mA.
  - Standard HCI Port (UART or USB)
  - USB Protocol: Full Speed USB1.1, Compliant With 2.0
  - This module can be used in the SMD.
  - It's made through RoHS process.
  - The board PIN is half hole size.
  - Has a 2.4GHz digital wireless transceiver.
  - Bases at CSR BC04 Bluetooth technology.
  - Has the function of adaptive frequency hopping.
  - Small (27mm×13mm×2mm)
  - Peripherals circuit is simple.
  - It's at the Bluetooth class 2 power level.
  - Storage temperature range: -40 ℃ - 85℃，work temperature range: -25 ℃ - +75℃
  - Any wave inter Interference: 2.4MHz，the power of emitting: 3 dBm.
  - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.

- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost

- Application fields:
  - Bluetooth Car Handsfree Device
  - Bluetooth GPS
  - Bluetooth PCMCIA , USB Dongle
  - Bluetooth Data Transfer
- Software
  - CSR

3. PINs description

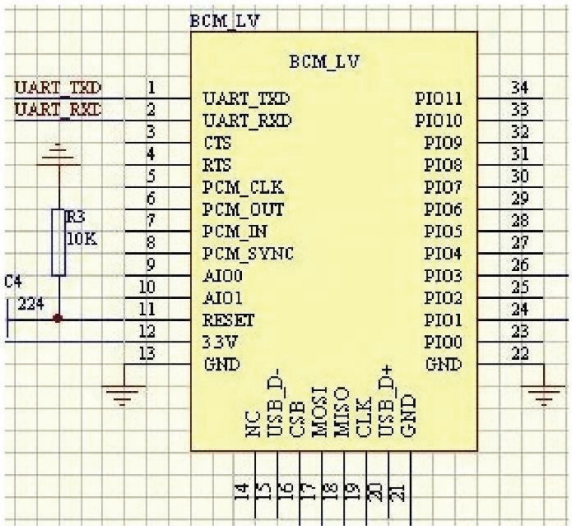


Figure 3 PIN configuration

The PINs at this block diagram is as same as the physical one.

PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
1V8	14	VDD	Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V	
VCC	12	3.3V		
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	

PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	CLK_REQ
PIO7	30	Bi-Directional	Programmable input/output line	CLK_OUT
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	
RESETB	11	CMOS Input with weak intemal pull-down		
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal	Chip select for serial peripheral interface, active low	

		pull-up		
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_-	15	Bi-Directional		
USB_+	20	Bi-Directional		
1.8V	14		1.8V external power supply input	Default : 1.8V internal powe r supply.
PCM_CLK	5	Bi-Directional		
PCM_OUT	6	CMOS output		
PCM_IN	7	CMOS Input		
PCM_SYNC	8	Bi-Directional		

4. The parameters and mode of product

LINVOR BLUE T

www.linvor.com

Bluetooth Module
Bluetooth

CSR,BC417143B

V 2.0

2006/09/6

蓝牙 **RF** 模块

1. 采用 CSR BC4 +8M FLASH 方案
2. 具有 PIO0-PIO11、AIO0、AIO1、USB、PCM、UART 及 SPI 接口，模块内置 8MFLASH，功能强大，用户可定制软件,适用于各种蓝牙设备，内置 RF 天线,便于调试。

蓝牙协议版本	Bluetooth Specifcation V2.0 With EDR
USB 协议 USB Protocol	Full Speed USB V1.1 Compliant With USB V2.0
频率	2.4Ghz ISM band
调制方式	GFSK(Gaussian Frequency Shift Keying)
发射功率	-4 ->4 dBm, Class 2
灵敏度	≤-80dBm at 0.1% BER
通讯速率	Asynchronous:2Mbps(Max)
供电电源	3.3V
工作温度	-20~+55 Centigrade
封装尺寸	27mmX13mmX2mm

Page 1of 2

If you want more information, please visit www.wavesen.com.