



SIMULACIÓN, INTEGRACIÓN E IMPLEMENTACIÓN DE UN ALGORITMO DE
ROBOTICA COLABORATIVA PARA DOS ROBOTS MOVILES DE
ARQUITECTURA HOMOGENEA

FREY GIOVANNI ZAMBRANO PINILLA

Trabajo de grado para optar al título de
INGENIERO EN MECATRONICA

Director

Ing. Ricardo Castillo Ph. D.

UNIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA EN MECATRÓNICA

BOGOTÁ D.C.

2018

Tabla de Contenido

TABLA DE CONTENIDO	2
ÍNDICE DE ILUSTRACIONES	5
ÍNDICE DE TABLAS	9
AGRADECIMIENTOS	10
RESUMEN	11
CAPÍTULO I	12
INTRODUCCIÓN	12
1.1 DEFINICIÓN DEL PROBLEMA	13
1.2 OBJETIVO GENERAL.....	13
1.3 OBJETIVOS ESPECÍFICOS	13
1.4 JUSTIFICACIÓN	14
1.5 ALCANCE O DELIMITACIÓN DE LA PROPUESTA	14
1.6 METODOLOGÍA.....	15
CAPÍTULO II	18
MARCO REFERENCIAL	18
2.1 ANTECEDENTES.....	18
2.2 MARCO TEÓRICO	22
2.2.1 <i>Robótica móvil</i>	22
2.2.2 <i>Robótica colaborativa</i>	24
2.2.2.1 Ambientes colaborativos	25
2.2.3 <i>Agentes</i>	25
2.2.3.1 Clasificación de los Agentes.....	25
2.2.3.2 Sistemas multi-agentes.....	26
2.2.3.3 Arquitectura de los RMAS.....	27
2.2.3.4 Características de un sistema multi-agente.....	28
2.2.3.5 Tipos de un sistema multi-agente	28
2.2.3.6 Agentes robóticos colaborativos	28
2.2.3.7 Diseño de grupos o enjambres de robots	29
2.2.4 <i>Sistemas Colaborativos</i>	30
2.2.5 <i>Inteligencia artificial</i>	32
CAPÍTULO III	35
DISEÑO MECATRÓNICO	35
3.1 METODOLOGÍA PARA EL DISEÑO MECATRÓNICO	35
3.1.1 <i>Concepto de Operación</i>	36
3.1.2 <i>Requisitos del sistema mecánico y estructural</i>	36
3.2 DISEÑO DETALLADO DE LA PARTE MECÁNICA Y ESTRUCTURAL	37
3.2.1 <i>Software de Diseño Mecánico (SolidWorks®)</i>	37
3.2.2 <i>Selección de componentes estructurales de diseño del robot homogéneo</i>	38
3.2.3 <i>Diseño del prototipo No 1 Robot Diferencial</i>	41
3.2.4 <i>Diseño del prototipo No 2 Robot omnidireccional</i>	41
3.2.5 <i>Análisis de la estructura por elementos finitos</i>	48

3.3 DISEÑO DETALLADO PARTE ELECTRÓNICA DEL ROBOT MÓVIL 004.....	49
3.3.1 Tarjeta de desarrollo.....	50
3.3.2 Sensores de contacto.....	52
3.3.3 Módulo de comunicación.....	53
3.3.4 Actuadores.....	53
3.4 DISEÑO DEL ALGORITMO COLABORATIVO.....	56
3.4.1 Percepción visual (visión de maquina).....	56
3.4.2 Orientación de los robots y objeto.....	58
3.4.3 Corrección del ángulo.....	59
3.4.4 Posición de acoplamiento.....	61
3.4.5 Subprograma de acoplamiento.....	63
3.4.6 Subprograma de la trayectoria de los robots al objetivo.....	65
3.4.7 Subprograma de Orientación de la plataforma.....	67
3.4.8 Subprograma de la trayectoria de la plataforma al punto de entrega.....	69
CAPITULO IV.....	71
SIMULACIÓN.....	71
4.1 SOFTWARE DE SIMULACIÓN DE ROBÓTICA WEBOTS®.....	71
4.2 TUTORIALES DESARROLLADOS WEBOTS®.....	72
4.3 DESARROLLO DE LA SIMULACIÓN DE TRABAJO COLABORATIVO.....	80
4.3.1 Simplificación del modelo en SolidWorks.....	80
4.3.2 Nuevo sistema de coordenadas del CAD.....	81
4.3.3 Importación del CAD a software de simulación.....	82
4.3.4 Ensamble de los robots en la simulación.....	86
4.3.5 Capas de desarrollo del software.....	89
4.3.5.1 Fase 1.....	90
4.3.5.1.1 Capa reactiva de los robots.....	91
4.3.5.1.2 Capa deliberativa del supervisor.....	91
4.3.5.2 Fase 2.....	91
4.3.5.2.1 Capa reactiva de los robots.....	92
4.3.5.2.2 Capa deliberativa del supervisor.....	93
4.3.5.3 Fase 3.....	93
4.3.5.3.1 Capa reactiva de los robots.....	93
4.3.5.3.2 Capa deliberativa del supervisor.....	94
CAPITULO V.....	95
IMPLEMENTACIÓN.....	95
5.1 CONSTRUCCIÓN DEL MODELO ESTRUCTURAL, MECÁNICO Y ELECTRÓNICO.....	95
5.2 DESARROLLO DEL SISTEMA MULTI-AGENTE.....	98
5.2.1 Programa del Agente Supervisor.....	99
5.2.1.1 Forma de captura de la posición de los robots en la visión de maquina.....	99
5.2.1.2 Capa deliberativa del Agente Supervisor.....	99
5.2.2 Programa de los Agentes Robot Azul y Robot Rojo.....	100
5.2.2.1 Capa reactiva de los robots.....	100
5.3 COSTOS DE LA IMPLEMENTACIÓN DE LOS ROBOTS.....	102
CAPITULO VI.....	103
PRUEBAS Y RESULTADOS.....	103
6.1 PROGRAMA DE IDENTIFICACIÓN DE COLORES.....	103

6.2 PROGRAMA DEL AGENTE SUPERVISOR.....	104
6.3 RESULTADOS DE LAS TRAYECTORIAS DE LOS ROBOTS VS TIEMPO	110
6.3.1 <i>Trayectorias directas</i>	110
6.3.2 <i>Cruce de trayectorias</i>	112
CONCLUSIONES	115
BIBLIOGRAFÍA	118
ANEXOS.....	121
ANEXO 1: PLANO DE ROBOTS Y PIEZAS.	121
ANEXO 2: PROGRAMA EN C DE LOS ROBOTS AZUL Y ROJO	122

Índice de Ilustraciones

Ilustración 1. Sistema multi-agente para el mapeo de entornos interiores de trabajo [2].	19
Ilustración 2 Sistema multi-agente para mapeo de entornos interiores de trabajo [3]	19
Ilustración 3 Robot utilizado en desastres naturales [12].	20
Ilustración 4 Equipo heterogéneo [31]	20
Ilustración 5 Robot móvil en trabajo colaborativo [14].	21
Ilustración 6 Robot Yumi ABB [10].	22
Ilustración 7 Contextualización de los agentes	23
Ilustración 8 Sistemas multi-agentes, comunicación e interacción [7].	27
Ilustración 9 Contextualización de los agentes	30
Ilustración 10 Robots móviles homogéneos	31
Ilustración 11 Dibujo del Método V [30]	35
Ilustración 12 Objeto	36
Ilustración 13 Detalle de la rueda sueca y su disposición en un robot omnidireccional tomada de fuente [2].	39
Ilustración 14 Robot diferencial [23].	40
Ilustración 15 Robot móvil con ruedas suecas y su maniobrabilidad (a). Robot Uranus (Universidad de Michigan) [22].	40
Ilustración 16 Robot diferencial 001	41
Ilustración 17 Diferentes estructuras de Robots Omnidireccionales (a) [26], (b) [27], (c) [24] y (d) [25].	41
Ilustración 18 Estructura interna de la rueda	42
Ilustración 19 Rodillo	42
Ilustración 20 Pasador	43
Ilustración 21 Eje conector	43
Ilustración 22 Rueda omnidireccional	43
Ilustración 23 Rueda conectada al motorreductor	43
Ilustración 24 Robot Híbrido [28]	44
Ilustración 25 Placa inferior	44
Ilustración 26 Placa superior	45
Ilustración 27 Vista isométrica del Robot Móvil 002	45
Ilustración 28 Vista inferior del Robot Móvil 002	46
Ilustración 29 Placa superior (b) y placa inferior (b)	46
Ilustración 30 Soporte de los motorreductores	47
Ilustración 31 Vista isométrica del Robot Móvil 004.	47
Ilustración 32 Estructura para la realización de estudio de elementos finitos	48
Ilustración 33 Estructura sometida al estudio de elementos finitos	49
Ilustración 34 Arduino Uno [29]	51
Ilustración 35 Diagrama de conexión del sensor [29]	52

Ilustración 36 Motorreductor	54
Ilustración 37 Diagrama de flujo para componente de vision	57
Ilustración 38 Orientación del robot	58
Ilustración 39 Orientación del objeto.....	59
Ilustración 40 Orientación de robot	60
Ilustración 41 Diagrama de flujo de subprograma corrección de ángulo	61
Ilustración 42 Robot y objeto en posición de inicio	61
Ilustración 43 Diagrama de flujo de los puntos de acoplamiento.....	63
Ilustración 44 Robots en posición de acoplamiento	64
Ilustración 45 Robots acoplados.....	64
Ilustración 46 Diagrama de flujo del subprograma de acoplamiento	65
Ilustración 47 Distancia de los robots a sus puntos de acoplamiento	66
Ilustración 48 Diagrama de flujo de la trayectoria del robot	67
Ilustración 49 Robos acoplados con su orientación	68
Ilustración 50 Diagrama de flujo de la función corregir ángulo	69
Ilustración 51 Trayectoria de la plataforma hasta el punto de entrega	69
Ilustración 52 Diagrama de flujo de la función trayectoria al punto de entrega	70
Ilustración 53 Interfaz del programa.....	71
Ilustración 54 Creación del mundo nuevo.....	72
Ilustración 55 Programa del e-punk.....	72
Ilustración 56 Diagrama de sensores.....	73
Ilustración 57 Creación de una esfera en el mundo.....	73
Ilustración 58 Opciones de nodo del programa.....	75
Ilustración 59 Árbol del programa principal.....	76
Ilustración 60 Mundo de dos robots heterogéneos.....	76
Ilustración 61 Mundo completo	77
Ilustración 62 Robots heterogéneos	78
Ilustración 63 Ventana para guardar modelo de SolidWorks®.....	79
Ilustración 64 Nodo creados a partir de la importación del modelo a Webots®	79
Ilustración 65 Modelo robot móvil 002 y 004	80
Ilustración 66 Modelo Robot móvil 002 y 004 simplificado	80
Ilustración 67 Ventana de guardar en formato VRML 97	81
Ilustración 68 Prototipos con nuevo sistema de coordenadas	82
Ilustración 69 Importación del robot a Webots	82
Ilustración 70 Identificación de las piezas por color	83
Ilustración 71 Robot identificado y construido	83
Ilustración 72 Opción de nodos para el robot.....	84
Ilustración 73 Nodo Physics.....	84
Ilustración 74 Robots importados a Webots.....	85
Ilustración 75 Escenario de la simulación.....	86

Ilustración 76	Árbol principal de la simulación	87
Ilustración 77	Árbol de nodos de los robots y el supervisor	87
Ilustración 78	Grafica de la brújula de los robots.....	88
Ilustración 79	Grafica de los motorreductores	89
Ilustración 80	Grafica de los servomotores de las pinzas	89
Ilustración 81	Diagrama de flujo de las fases de simulación.....	90
Ilustración 82	Robots en fase 1	90
Ilustración 83	Llegada de los robots al punto de acoplamiento	91
Ilustración 84	Momento del acoplamiento	92
Ilustración 85	Robots en la fase de acoplamiento	92
Ilustración 86	Robots acoplados y cargados con el objeto	93
Ilustración 87	Llegada del robot acoplado a punto llegada	94
Ilustración 88	Placas superiores en acrílico.....	95
Ilustración 89	Placa inferior de los robots.....	95
Ilustración 90	Ensamble motorreductor eje y llanta omnidireccional	96
Ilustración 91	Placa inferior con motorreductor y rueda omnidireccional (a) y placa inferior del Robot Azul con los motorreductores (b)	96
Ilustración 92	Robot Azul con la parte electrónica	96
Ilustración 93	Robot Azul piezas del acoplamiento hembra y Robot Rojo piezas de acoplamiento macho	97
Ilustración 94	Placa inferior del Robot Rojo con los motorreductores	97
Ilustración 95	Robot Rojo parte superior con el sistema electrónico	98
Ilustración 96	Sistema multi-agente.....	98
Ilustración 97	Ventana del programa de umbralizacion de colores	103
Ilustración 98	Diagrama de flujo del programa de detección de colores	104
Ilustración 99	Ventada del programa del Agente software	104
Ilustración 100	Diagrama de flujo del Agente Supervisor	105
Ilustración 101	Programa en ejecución	105
Ilustración 102	Diagrama de flujo fase 1.....	106
Ilustración 103	Programa en ejecución del desplazamiento de los robots	106
Ilustración 104	Programa en Fase 2	107
Ilustración 105	Fase 2 acoplamiento.....	107
Ilustración 106	Diagrama de flujo de la Fase 2.....	108
Ilustración 107	Fase 3	108
Ilustración 108	Diagrama de flujo Fase 3	109
Ilustración 109	Grafica de la trayectoria de los robots en la interfaz de usuario	110
Ilustración 110	Grafica del desplazamiento en X Y del robot Azul.....	110
Ilustración 111	Grafica del desplazamiento en X Y del robot Rojo	111
Ilustración 112	Grafica de la orientación de los robots	111
Ilustración 113	Grafica de posiciones de los robots al objeto	112

Ilustración 114 Grafica de desplazamiento del Robot Móvil Rojo distancia en X vs tiempo	112
Ilustración 115 Grafica de desplazamiento del Robot Móvil Rojo en Y vs tiempo	113
Ilustración 116 Grafica del Robot Móvil Azul distancia en X vs tiempo	113
Ilustración 117 Grafica del Robot Móvil Azul distancia en Y vs tiempo	114

Índice de Tablas

Tabla 1 Problemas de la robótica móvil	23
Tabla 2 Ventajas y desventajas de los sistemas multi-agente robóticos	26
Tabla 3 Cuadro que identifica los diferentes escenarios de la Inteligencia artificial y la robótica.	32
Tabla 4 Cuadro del cálculo de la masa y de la fuerza que ejerce el robot móvil sobre los motores, para determinar el torque necesario de los motores para poder desplazarse a una velocidad angular de 5.2 rad/seg.	48
Tabla 5. Cuadro comparativo de las tarjetas de desarrollo consultadas para el proyecto	51
Tabla 6 Cuadro comparativo de módulos de comunicación para los robots	53
Tabla 7 Cuadro comparativo de módulos los controladores de los motores.	55
Tabla 8 Representa la comparación de las características evaluadas en el software de simulación Webots® de los dos prototipos.	85
Tabla 9 Nodos que componen a los robots y al supervisor.	88
Tabla 10 Gastos del proyecto	102

Agradecimientos

Gracias a los docentes por el tiempo invertido en cada uno de los estudiantes de Mecatrónica, a mi Esposa mujer paciente, humilde, amorosa y ante todo por su fortaleza espiritual, apoyándome de cuerpo y alma, a mi Madre y mi Padre, hogar que dio todo por sus hijos y ante todo la gloria a Jesús y María sin ellos nada de esto sería posible.

Resumen

En este trabajo de investigación se diseñó, implementó y desarrolló un sistema multi-agente que utiliza un algoritmo como estrategia, para el trabajo de dos robots móviles homogéneos, en el desarrollo de una tarea de forma colaborativa.

Para el proyecto se diseñó un robot omnidireccional en su parte estructural, mecánica, electrónica y de software. El robot se simuló en el programa Webots® determinando su comportamiento e implementando el algoritmo colaborativo para ejecución de la tarea determinada.

El sistema multi-agente está compuesto de 3 agentes: dos robots omnidireccionales y un Supervisor (software de Matlab®), que comandó el sistema, utilizando técnicas de Inteligencia Artificial clásica, implementando capas híbridas que están compuestas de capas reactivas y deliberativas.

El objeto por transportar tiene una dimensión superior a la dimensión de un robot, razón por la cual, se diseñó e implementó un sistema mecánico y electrónico que permite a los dos robots acoplarse y formar una plataforma. El agente Supervisor dirigió y controló el movimiento y desplazamiento de forma coordinada y colaborativa.

La implementación, integración y verificación de la tarea colaborativa a desarrollar fue dividida en tres fases: 1) desplazamiento de los robots desde diferentes puntos iniciales hasta dos puntos de acoplamiento. Durante esta fase los robots son dirigidos por el agente Supervisor (software), con el empleo de la visión de máquina, detectando la posición, orientación de los robots y del objetivo, empleando el algoritmo se determinó las trayectorias a los puntos de acoplamiento, dirigido y controlado el desplazamiento de los robots y su orientación en un ángulo fijo de $\pi/2$ finalizando la fase al llegar a los puntos de acoplamiento. Los robots al estar ubicados en los puntos de acoplamiento dan inicio a la fase 2) allí toman el mismo ángulo del objetivo e inician el acoplamiento, para formar una sola plataforma y poder transportar el objeto. Al finalizar el acople se le carga el objeto finalizando la fase 2. La 3) los robots se mueven como uno sola pieza manteniendo el ángulo ideal de $\pi/2$, transportando el objeto hasta un punto de entrega finalizando el desarrollo de la tarea colaborativa.

Palabras claves: multi-agente, algoritmo, heterogéneo, colaborativa, protocolo, robot.

CAPÍTULO I

INTRODUCCIÓN

La implementación de sistemas multi-agentes en el desarrollo de tareas de forma colaborativa, ha logrado que en la actualidad las diferentes líneas de investigación busquen como integrar en el desarrollo de una tarea varios tipos de robots, como en este proyecto dos robots homogéneos, empleando técnicas de agente reactivo, deliberativo e híbrido de Inteligencia Artificial clásica.

La metodología empleada en el desarrollo mecatrónico del proyecto fue el Modelo V, integrando las partes: mecánicas, electrónicas y el software en tres fases generales: definición del proyecto, implementación y pruebas e integración.

En el proyecto fue importante el diseño de un robot móvil base, teniendo en cuenta que serían dos robots homogéneos, para el desarrollo de la tarea y del trabajo colaborativo, basado en los requerimientos establecidos: mecánicos y electrónicos, logrando un acoplamiento estructural que permitió su movimiento como una sola pieza y ser orientado y controlado para cumplir con la tarea de forma colaborativa.

El desarrollo de robots homogéneos que puedan interactuar y trabajar de forma colaborativa, representa para la robótica la integración de robots que cumplan tareas de forma individual y también grupal, lo cual genera que deban ser compatibles y reemplazables por los mismos robots, cada uno puede ser reemplazado por otro y lo más importante su unión y acoplamiento para el desarrollo de objetivos en común.

La implementación de los robots en la simulación represento el mejor camino para la selección del robot a ser construido, determinado su comportamiento individual y colectivo, reduciendo el tiempo en pruebas reales y encontrando el mejor prototipo para el desarrollo de la tarea de forma colaborativa.

Se da inició con el diseño de dos robots homogéneos modelándolos en un software de diseño asistido como CAD, para luego ser construidos e implementados en simulación. Y seguidamente llevar a cabo el prototipado físico de estos mismos para desarrollar el programa principal, siempre bajo la orientación del agente Supervisor (software).

1.1 Definición del problema

En la actualidad los desarrollos en robótica colaborativa que emplean multi-agentes, ha permitido el análisis del comportamiento de los sistemas, evaluando la reducción de costos en el empleo de robots de menor complejidad, fraccionando así las tareas para su ejecución y cómo la pérdida de un agente permite que sea relevado por otro; teniendo en cuenta su homogeneidad lo cual hace que el sistema permanezca en ejecución, con un mayor grado de estabilidad.

El proyecto esta implementado en la línea de investigación de robótica móvil colaborativa, que consta de tres fases: detección de objetos, transporte de materiales e integración de la fase uno y dos y navegación, el presente trabajo está enmarcado en la segunda fase de la línea de investigación.

Con el proyecto se plantea desarrollar un algoritmo que permita el trabajo colaborativo de dos robots móviles de arquitectura homogénea en la ejecución de una tarea, empleando sus características físicas, teniendo en cuenta su entorno y los posibles escenarios que se les pueda presentar. La comunicación y la sincronización son tareas colaborativas que deben desarrollarse, inicialmente en simulación y posteriormente en el prototipo físico, llegando al interrogante del proyecto ¿Cómo hacer que dos Robots móviles realicen de forma autónoma, tareas de forma Colaborativa?

1.2 Objetivo general

Desarrollar, simular e implementar una estrategia de trabajo en robots móviles de arquitectura homogénea para el desarrollo de una tarea colaborativa, implementándola en un programa de simulación de robots (Webots) y luego de forma física utilizando robots reales.

1.3 Objetivos específicos

- ✓ Diseñar los robots móviles en su parte electrónica y mecánica, modelándolos en un software de diseño asistido.
- ✓ Definir y simular la tarea colaborativa en el programa Webots y así generar el escenario, tarea y robots.
- ✓ Implementar las partes electrónicas y mecánicas necesarias para ensamblar los robots y desarrollar la tarea colaborativa.

- ✓ Realizar las pruebas finales con los robots en el desarrollo de la tarea colaborativa y evaluar los resultados obtenidos en la simulación y la parte real.

1.4 Justificación

Los escenarios de aplicación de la robótica han evolucionado desde entornos muy simples y controlados, a entornos dinámicos en exteriores; pasando de los robots manipuladores pesados y anclados a robots móviles y pequeños, que tengan la capacidad de adaptarse a diferentes espacios de trabajo. Así mismo, la complejidad de las tareas crea la necesidad de ser divididas en varias, lo cual dio origen a la robótica colaborativa como solución teniendo en cuenta que las tareas pueden ser diferentes, generando sistemas que consideren la colaboración de robots heterogéneos, que puedan ser empleados en múltiples aplicaciones, como la robótica de servicio en entornos urbanos o monitorización de desastres.

Hoy día las necesidades del hombre de la industria han cambiado, buscando una mayor incursión de la tecnología en las pequeñas y mediana empresas e incluso en el hogar, lo que ha generado que la tecnología busque desarrollar nuevos robots acordes a las necesidades, que permitan un acceso mayor, que su estructura y comportamiento, concedan su interacción con el hombre, velando por la seguridad de este. Por lo cual la robótica colaborativa es el camino, para acercar al hombre a una interacción más cercana con el robot, siendo una herramienta indispensable para las necesidades y retos que impone el mismo hombre.

Es por esto, que el empleo de dos robots móviles de arquitectura homogénea, en el desarrollo de una tarea colaborativa permite analizar el tiempo y el cumplimiento de la tarea desde dos escenarios posibles, como lo son: la realización de la tarea por un solo robot o los dos robots, permitiendo evaluar el algoritmo colaborativo empleado.

1.5 Alcance o delimitación de la propuesta

Se realizará la simulación en un software, de los dos robots móviles de arquitectura homogénea, implementando un sistema multi-agente que controle los dos robots en una tarea colaborativa, realizando pruebas y evaluando el sistema desarrollado en el programa Webots. Se llevará a cabo el prototipo de los robots y se desarrollarán la tarea antes simulada, evaluando y verificando la simulación y la ejecución física de la tarea.

Este trabajo se efectuará en las instalaciones de la Universidad Militar Nueva Granada en los laboratorios de robótica, automatización, electrónica y taller de diseño mecánico. El tiempo requerido para cumplir con las labores estipuladas es de máximo seis meses.

1.6 Metodología

Las etapas en las cuales se va a desarrollar para la ejecución del proyecto son:

1. Diseño de los robots parte mecánica y electrónica:
 - ✓ Elección de prototipos para la selección del robot principal.
 - ✓ CAD de los prototipos.
 - ✓ Selección, simulación y pruebas de los elementos electrónicos para los diferentes agentes.

2. Conocimiento y manejo del software:
 - ✓ Evaluación y conocimiento del software a ser implementado para el desarrollo del proyecto.
 - ✓ Desarrollo de tutoriales para conocer las capacidades y la forma de uso del software de simulación Webots®.
 - ✓ La integración de los prototipos a Webots® evaluando su comportamiento y así la selección del robot que más se adaptara al desarrollo de la tarea.

3. Diseño de un Robot prototipo:
 - ✓ Diseño de la estructura, elementos mecánicos, electrónicos y el algoritmo de programación.
 - ✓ Importación, integración de los elementos de la estructura en el software de simulación Webots®.

4. Sistema multi-agente:

- ✓ Selección de la tarea a desarrollar de forma colaborativa por medios de un sistema multi-agente y técnicas de control de Inteligencia Artificial.
- ✓ Diseño del algoritmo que determina el comportamiento de los agentes.
- ✓ Evaluación del algoritmo e implementación en el software.
- ✓ Desarrollo de tarea por parte del sistema multi-agente creado, desarrollo de pruebas para verificar el programa y los agentes.

5. Implementar y evaluar el funcionamiento

- ✓ Implementación de los agentes robots físicamente y el agente Supervisor como sistema multi-agentes, el desarrollo del algoritmo diseñado en la simulación, evaluar su comportamiento e implementar un protocolo de comunicación y ejecución de órdenes.
- ✓ Realización de pruebas al sistema multi-agente que emplea los programas de comando de Inteligencia Artificial clásica para determinar su comportamiento y el cumplimiento de los objetivos trazados.

6. Redacción del artículo

El artículo plasmara las bases teóricas, las posibles soluciones para el desarrollo del problema, y así generar alternativas de solución para implementar en las simulaciones y de manera física.

Capítulo II

Marco Referencial

2.1 Antecedentes

Los robots están presentes entre los seres humanos hace bastante tiempo, en una evolución constante en relación con el desarrollo tecnológico, convirtiéndose en los aliados del hombre para el desarrollo de tareas que el hombre por el nivel de riesgo no es aconsejable que las realice para lograr una optimización en la producción y la ejecución de tareas repetitivas que el robot podrá realizar más rápidamente.

Ambientes colaborativos (CSCL). El empleo de la colaboración empleado para facilitar el trabajo y la transmisión de información fue inicialmente propuesto por la National Science Foundation en 1989 y después promovido por el National Research Council en 1993. los ambientes colaborativos son aquellos en los cuales, un grupo de entidades se encuentran trabajando conjuntamente en un mismo ambiente de trabajo, teniendo como principio la colaboración, en la cual todos hacen parte de los que cumplen una tarea específica [13].

El termino CSCW (Computer-Supported Collaborative Work), fue creado para exponer el rol de la tecnología en el ambiente de trabajo (Grudin 1994). La palabra cooperativo se refiere a la búsqueda de una meta común mediante la división de tareas y colaborativo que es al logro de un objetivo compartiendo tareas [13].

En el año 1988, crearon un programa llamado KALI; que integra un sistema de control de multiprocesador y multi-manipulador, obteniendo un equilibrio entre un gran número de requisitos, determinando que la sencillez tiene prioridad, sobre la eficiencia, empleando bloques de construcción que permiten ser empleados de diferente manera [1].

En el año 2010 plantearon el diseño de un sistema multi-agente robótico, constituido por un robot móvil y un agente de software, que cooperan entre sí, en la construcción de modelos de entornos interiores de trabajo para la navegación colaborativa en escenarios estructurados [2] logrando mapear un lugar, por medio del robot que reconocía cada uno de los obstáculos, enviando la información al programa que los interpretaba y generaba el entorno en el cual se encontraba el robot, Ilustración 1.

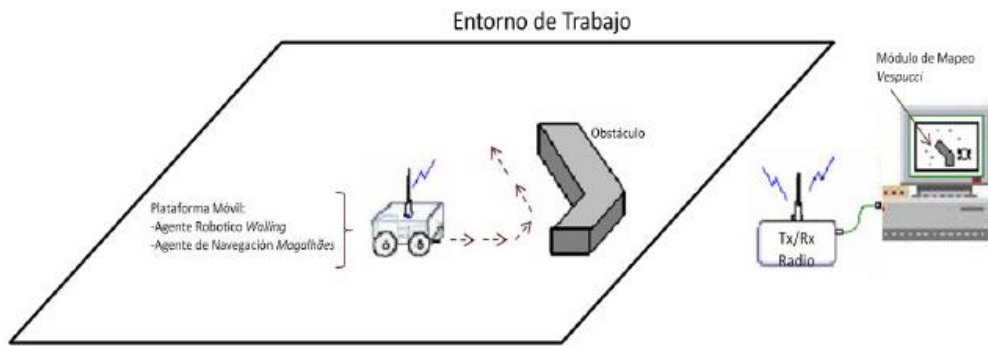


Ilustración 1. Sistema multi-agente para el mapeo de entornos interiores de trabajo [2].

En el 2011 se diseñó un sistema de control colaborativo de robots autónomos basado en la planificación automática, el cual plantea el diseño de una arquitectura de control híbrida, distribuida y modular para robots heterogéneos basada en planificación automática, mediante la actuación combinada de robots, buscando un comportamiento inteligente en la Ilustración 2 se puede ver la posible ruta que debe tomar el robot móvil parte del trabajo que debe realizar el control colaborativo.

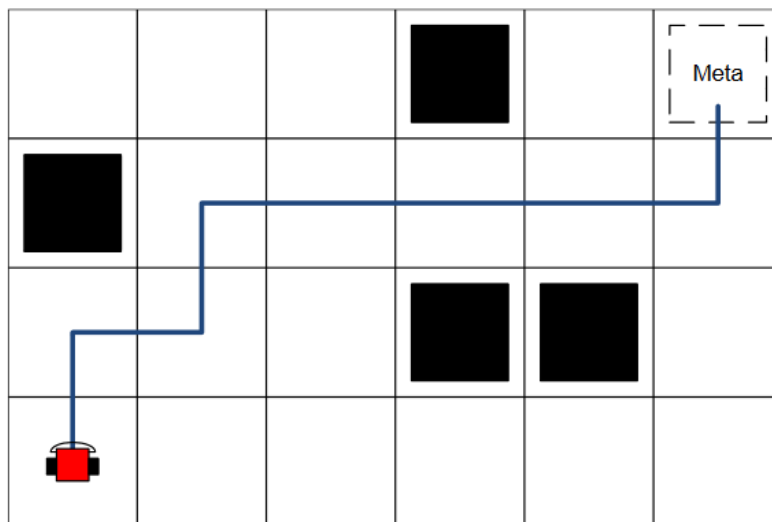


Ilustración 2 Sistema multi-agente para mapeo de entornos interiores de trabajo [3]

En el 2011 investigador de la Universidad Santiago de Chile dan vida a una comunidad de robots heterogéneos, los cuales tienen la misma estructura, para ser empleados en distintas áreas con énfasis en zonas de desastres, al tener en cuenta la posición geográfica de Chile, lo cual hace que sea propensa a terremotos, maremotos, tsunamis, etc., siendo una posible

solución para afrontar dichos acontecimientos, minimizando el riesgo del personal de rescatista e incrementando la probabilidad de vida de los afectados, tomando como modelo la colonia de hormigas, buscando imitar dicho comportamiento por medio de un programa [18].

La investigación y el desarrollo de la comunidad de robots, es también empleándolos también en la capacitación de estudiantes de colegios [18] Ilustración 3.



Ilustración 3 Robot utilizado en desastres naturales [12].

En el año 2012 se diseñó un equipo robótico heterogéneo [31], compuesto por un avión robótico, un vehículo de superficie y un vehículo submarino autónomo, los cuales están integrados para el monitoreo de los arrecifes de coral, cada uno cumple una misión en relación con la tarea principal, Ilustración 4 equipo heterogéneo en el desarrollo de la tarea principal.

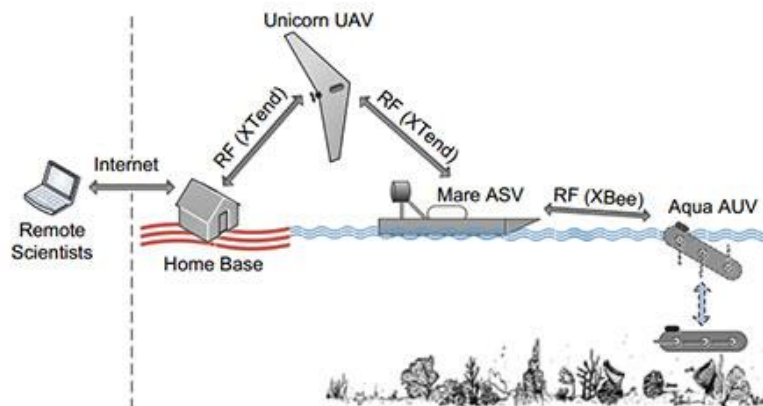


Ilustración 4 Equipo heterogéneo [31]

Robótica móvil colaborativa

Emplea robots móviles provistos de ruedas, extremidades o algún otro mecanismo que le permita moverse de forma autónoma, empleando los datos obtenidos por medio de los sensores para ejecutar control del movimiento [21].

Los datos obtenidos, son empleados, para que le programa los emplee de forma colaborativa para la navegación de un punto hasta un punto, empleando la información obtenida por cada uno de los robots móviles, creando de forma colaborativa la navegación más óptima para poder llegar al lugar determinado, así mismo, empleando dicha información para realizar cálculos de ruta, determinar los diferentes obstáculos presentes en su camino y poder realizar la evasión, transmitiendo a sus compañeros la ubicación propia y de los obstáculos que ha reconocido.

Una tarea muy colaborativa es la comunicación entre robots, determinando una jerarquía para que exista una sincronización en la transmisión de datos, todos dependen de la información de los otros para que todos puedan emplear la mejor ruta [21] Ilustración 5.

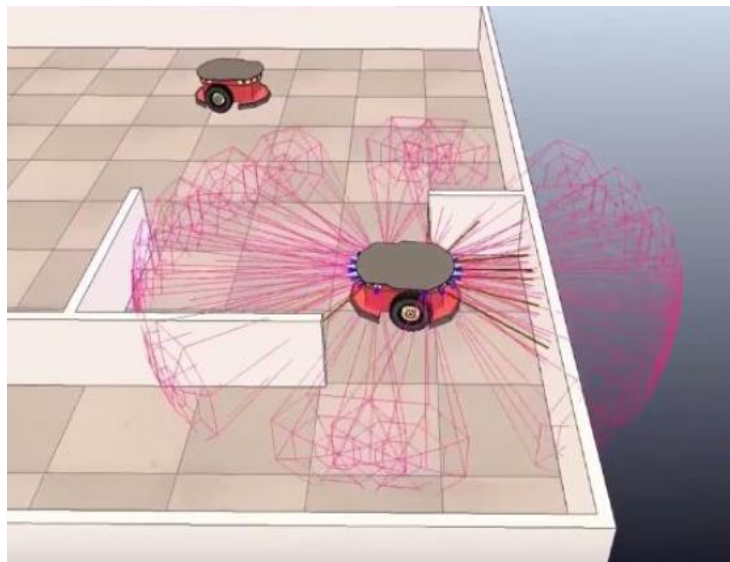


Ilustración 5 Robot móvil en trabajo colaborativo [14].

El empleo de robots móviles en su área de acción como lo son la navegación gracias a sus características, que le permiten un desplazamiento rápido, siendo empleados en tareas que requieran un reconocimiento de terreno o el movimiento de un punto a otro punto.

La firma especializada en tecnologías eléctricas y de automatización ABB presente el primer y único robot industrial de doble brazo colaborativo, los poderosos brazos de YuMi hacen del robot de última generación, el cual trabaja juntamente con humanos gracias a su diseño intrínsecamente seguro, según la empresa que lo creó [17]. Este robot cuenta con dos manos antropomórficas, de catorce (14) ejes que son adaptables y configurables, empleando sensores y cámaras que le permiten realizar tareas de montaje y ensamblado con piezas y objetos diminutos, este robot hace de la colaboración entre humanos y robots sea una realidad Ilustración 6.



Ilustración 6 Robot Yumi ABB [10].

2.2 Marco Teórico

2.2.1 Robótica móvil

La robótica móvil, tiene la función de explorar el medio colaborativo, por lo cual en el momento de hacer parte de un grupo robots, es muy versátil en el campo de la robótica, pero también presenta unos problemas asociados a la misma que están representados en la Tabla 1, son interrogantes que el programador y diseñador deben resolver para poder dar cumplimiento a la solución del problema planteado, o la misión que deben cumplir.

Tabla 1 Problemas de la robótica móvil

Problemas Asociados a la Robótica Móvil		
ítem	Asuntos	Interrogantes por Resolver
1	Percepción	¿Qué hay en el mundo?, ¿Cómo percibo mi entorno?
2	Localización	¿Dónde estoy?, ¿Cómo me ubico en mi entorno?
3	Navegación	¿Cómo me desplazo de un sitio a otro sitio?
4	Inteligencia	¿Qué tengo que hacer?
5	Autonomía	¿Cómo lo tengo que hacer?
6	Cooperación	¿Cómo interactuó con el ser humano? ¿Cómo interactuó con otros robots?

➤ Robótica móvil en entornos urbanos

Los robots móviles se clasifican con relación al medio en el que se desplazan: terrestres, marinos y aéreos, los terrestres generalmente se desplazan mediante ruedas o patas, son empleados en aplicaciones de rastreo y traslado de objetos, evasión de obstáculos, es ahí donde la interacción con un ambiente dinámico, como lo es un entorno urbano, hace que el robot cuente con unas características determinadas y se necesite un sistema de robots o sensores que se comporten de forma colaborativa.



Ilustración 7 Contextualización de los agentes

La Ilustración 7 muestra al robot y los principales sensores que emplea el Robot Romeo en la realización de tareas en entornos urbanos, dispone de codificadores en las ruedas que determinan la velocidad, giroscopio y una unidad media inercial para la estimación de los ángulos y velocidades angulares, dispone de cuatro (4) sensores de distancia laser y una cámara para la navegación y localización.

Robótica de servicio

- El guiado.
- El transporte de persona y objetos.
- Las zonas peatonales hacen que el robot tenga la capacidad de navegar al mismo tiempo que las personas y posiblemente con otros robots.
- Algoritmo de identificación de personas, obstáculos.
- Múltiples nodos de asistencia robótica para la realización de una tarea.
- Los sistemas multi-agente no solo aceleran el cumplimiento de la tarea, aumentando la robustez y eliminan los fallos aislados.
- Buscan la movilidad en un área de una manera eficiente y organizada.

Red de cámaras fijas o móviles son empleadas para cubrir el entorno y permite seguir objetos de interés, permite obtener información de un entorno que, al combinarlo con la información obtenida por el robot, por medio de sus sensores mejoran la percepción del entorno.

Red inalámbrica de sensores la electrónica de baja potencia y sistema de comunicación inalámbrica: forman redes as-doc que hace posible rutar toda la información hacia un concentrador que recibe la información, son llamadas redes inalámbricas de sensores (Wireless Sensor Networks, WSDs). [2] Se emplea la comunicación inalámbrica para construir y compartir mapas colectivos del medio, explorando el entorno al máximo con el tiempo mínimo, se debe contar con una estrategia de comunicación que determina si es necesaria y con qué frecuencia se deben conectar los nodos para obtener resultados eficientes.

2.2.2 Robótica colaborativa

La robótica colaborativa es la encargada de estudiar, diseñar e implementar la integración de los robots como parte de un equipó, que es utilizado para el cumplimiento de una tarea. Se espera tener un comportamiento de grupo basándose en los comportamientos individuales. [5] La robótica colaborativa emplea múltiples nodos robóticos para el logro eficiente de tareas complejas y arduas, que requieren de una coordinación y una ejecución de tareas que sean cumplidas con eficiencia, buscando reducir el tiempo y el menor consumo de energía [7].

2.2.2.1 Ambientes colaborativos

Es un grupo de entidades, agentes que trabajan conjuntamente, en un mismo ambiente de trabajo, propiciando la colaboración. En un escenario colaborativo es necesario que se den tres elementos claves: Igual participación, responsabilidad individual e independencia positiva. [16]

CSCW (Computer - Supported Collaborative Work) rol de la tecnología en un ambiente de trabajo.

- Cooperativo: búsqueda de una meta común mediante la división de tareas.
- Colaborativo: logro del objetivo compartiendo tareas.

2.2.3 Agentes

Retomando el concepto de agente se remonta desde los años 70s al modelo de “Actor concurrente”, derivando en el concepto de Agente autónomo, interactivo y ejecutor-concurrente, presenta un estado interno encapsulado y podía responder a los mensajes de otros objetos similares. Los agentes son entidades que actúan en representación de sus usuarios humanos o dueños para realizar tareas complicadas, empleando un protocolo de comunicación y realizan sus acciones concurrentemente. Poseen propiedades como; autonomía, habilidad social, reactividad, proactividad, movilidad, continuidad temporal, adaptabilidad y aprendizaje. Puede habitar en ambientes dinámicos y complejos censando y actuando de manera autónoma, en busca de su adaptación, realizando una serie de tareas o metas para las cuales fueron diseñados.

2.2.3.1 Clasificación de los Agentes

Los agentes se pueden clasificar en: bióticos y de vida artificial. El primer grupo hace parte de los seres humanos, animales, bacterias entre otros. El segundo grupo hace parte de los agentes de software y los agentes robóticos; estos se pueden clasificar en agentes de tarea específicas y agentes de entrenamiento.

2.2.3.2 Sistemas multi-agentes.

Un sistema multi-agente es una agrupación de agentes que actúan entre sí para el desarrollo de una tarea determinada. Debe existir una estructura que permita la comunicación y/o la interacción entre los agentes.

la planeación y la programación determinan las tareas que pueden ser ejecutadas generando una coordinación general. Pueden trabajar de forma cooperativa para alcanzar objetivos globales, combinación de capacidad de decisión y actuación, para hacer frente de forma dinámica. [7]

Podemos agrupar las ventajas y desventajas en la Tabla 2.

Tabla 2 *Ventajas y desventajas de los sistemas multi-agente robóticos*

VENTAJAS Y DESVENTAJAS DE LOS SISTEMAS MULTI-AGENTE ROBÓTICOS	
Ventajas	1. Mejora del Rendimiento: Se requiere de menos tiempo para llevar a cabo una tarea. 2 acción distribuida y percepción distribuida: Mejora la tolerancia a fallos, presenta redundancia.
Desventajas	1. Interferencias entre sistemas sensoriales cuando se emplean varios robots 2. Incertidumbre respecto a otros agentes robóticos: por ejemplo, las intenciones operativas que puedan tener los demás robots. 3. Complejidad del sistema de comunicación: este permite la transferencia de datos y ordenes entre diferentes agentes.

Los agentes adquieren una nueva dimensión en el momento que se agrupan, generando el sistema multi-agente. Características de un sistema multi-agente:

Organización social: la forma en la cual los agentes se organizan, de acuerdo con una función o rol que le sea asignado. Cooperación: es la negociación entre los agentes, el cual comparten información, resultados con otros agentes. Coordinación: es la coordinación de sus acciones, con los agentes que están presentes en su entorno, con el objetivo de cumplir la tarea que se le ha sido asignada.

Control: es el mecanismo básico de apoyo para el control global y control local. Comunicación: los agentes pueden comunicarse con el fin de alcanzar sus metas o las de la sociedad o sistema el cual habitan ver Ilustración 8.

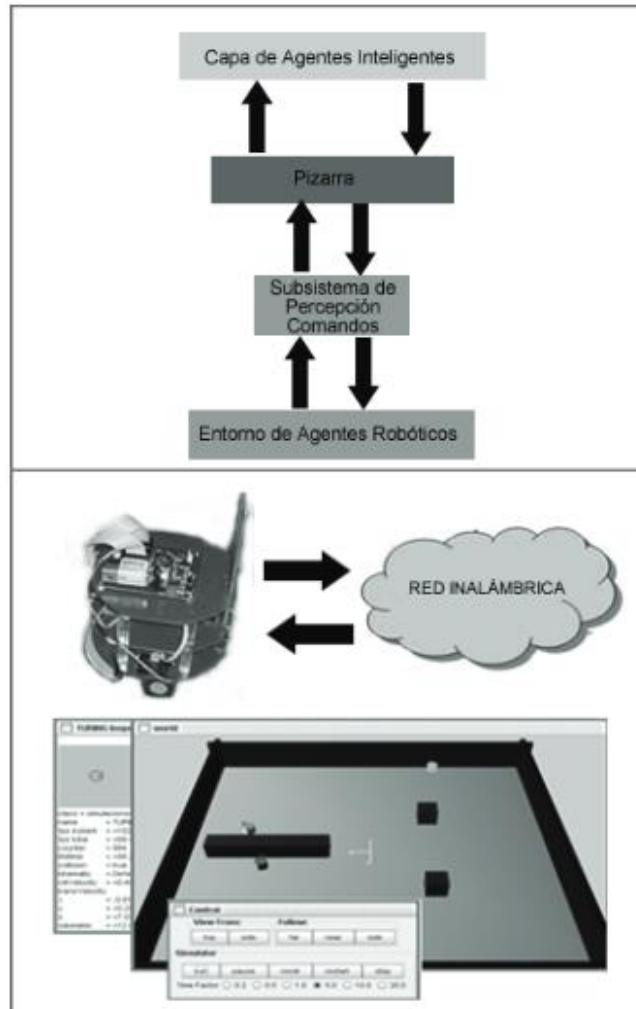


Ilustración 8 Sistemas multi-agentes, comunicación e interacción [7].

2.2.3.3 Arquitectura de los RMAS

- ✓ Arquitectura de árbol fijo: jerarquía tradicional y medidas de control.
- ✓ Red de agentes conectados: comunicación entre agentes, emplea esquemas de comunicación.
- ✓ Colección de grupos de agentes conectados: los agentes que componen un grupo se comunican vía un agente escogido, que a su vez se conectan en un grafo completamente conectado.
- ✓ Niveles de representación basados en la Abstracción: es la imagen de sistemas distribuidos jerárquicos.

2.2.3.4 Características de un sistema multi-agente

Los agentes adquieren una nueva dimensión en el momento que se agrupan, generando el sistema multi-agente. Características de un sistema multi-agente:

- Organización social: la forma en la cual los agentes se organizan, de acuerdo con una función o rol que le sea asignado.
- Cooperación: es la negociación entre los agentes, el cual comparten información, resultados con otros agentes
- Coordinación: es la coordinación de sus acciones, con los agentes que están presentes en su entorno, con el objetivo de cumplir la tarea que se le ha sido asignada.
- Control: es el mecanismo básico de apoyo para el control global y control local.
- Comunicación: los agentes pueden comunicarse con el fin de alcanzar sus metas o las de la sociedad o sistema el cual habitan.
- Ventajas y desventajas de los RMAS, exigen del diseñador una mayor coordinación, que nos es muy significativa con un agente.

2.2.3.5 Tipos de un sistema multi-agente

- Auto-organizado: cada agente dentro del sistema realiza su tarea y el comportamiento global depende de los comportamientos individuales de los agentes. No cuenta con un elemento que organice y controle.
- Cooperativos: la comunicación es constante o por solicitud, compartiendo información, organizando la tarea que debe realizar cada agente, para cumplir con una meta global.
- Adaptativos: está determinado por su experiencia, así determina su comportamiento.

2.2.3.6 Agentes robóticos colaborativos

- Agentes robóticos dotados de percepción (sistema sensorial).
- Cierta grado de inteligencia realice tareas útiles en su entorno de trabajo (detalle en la comprensión espacial de su entorno).

Un solo agente robótico dispone de toda los sensores y potencia computacional, lo cual determina que su diseño sea complejo, para lograr la integración de los sensores y su procesador. [16]

Desventajas:

- Costos de diseño y construcción.
- Tamaño físico del agente.
- Mayo consumo de energía.
- Menor confiabilidad.

En la actualidad en la industria se requiere emplear dispositivos mecatrónicos diversos con un solo fin o tarea compleja tarea colaborativa, esto hace que disminuya el tiempo en la ejecución de las tareas, aumente la productividad y disminuyan los costos, se logra mediante la división de procesos complejos en varios procesos simples, los cuales va a ser ejecutadas en la realización de tareas paralelas, por robots sencillos y de un menor costo.

El desarrollo e implementación de una estrategia para la coordinación de un sistema de robots multi-agente heterogéneos, con fin de que el sistema multi-robots pueda desarrollar diferentes de tareas, de forma colectiva, empleando IA y visión de maquina

Diseñar e Implementar un software que permita simular el sistema multi-robot, de forma que sea posible visualizar su comportamiento al realizar actividades de exploración y localización, recolección y transporte de elementos de diferentes características y que al mismo tiempo permita la evasión de obstáculos dentro de un ambiente.

2.2.3.7 Diseño de grupos o enjambres de robots

Las desventajas presentadas por los agentes robóticos colaborativos, hace que la tecnología busque, la solución, por lo cual el estudio de la naturaleza, en la cual se presentan las colonias (hormigas, abejas entre otras) solución de la naturaleza para la supervivencia de una especie, llevándolo a la tecnología, la creación de un grupo de robots

- Simplificación y distribución de tareas: permite la división o fraccionamiento de las tareas para que su cumplimiento sea más rápido y fácil.
- Sensórica e inteligencia distribuida: la implementación de los sensores es de forma distribuida, permitiendo abarcar mayor espacio en un menor tiempo.
- Posibilidad del diseño de mini-robots de bajo consumo de energía.

Mayor capacidad de trabajo, pero el diseñador debe enfrentarse a problemas más complejos:

- La coordinación entre agentes.
- Toma decisiones, la inteligencia y sensórica distribuida, debe estar soportados en por un sistema de comunicación robusta y confiable.

2.2.4 Sistemas Colaborativos

La industria necesita estar al ritmo del desarrollo a nivel mundial para poder seguir fuerte y vigente, orientado al aumento de la productividad y la reducción de los costos, requiriendo sistemas de manufactura colaborativa, distribuida e inteligente que ofrezca flexibilidad y reconfiguración.

La gestión de manufactura colaborativa (CMM), surge como solución al CIM, en el cual los sistemas automatizados de control (SAPs), integran el software y el hardware, en una arquitectura de control, fuertemente jerárquica y centralizada y en una estructura de funcionamiento (Planeamiento, agendamiento y ejecución) secuencial y rígida.

En la industria estos sistemas cuentan con la limitación de su rigidez y jerarquía lo cual hace que, en el momento de parar por una interrupción del sistema, tampoco permite una adaptación con eficiencia y efectividad a cambios.

➤ Sistemas basados en agentes

Los sistemas multi-agentes (MAS) se derivan del concepto de la inteligencia artificial distribuida (DAI). El MAS es un conjunto de agentes que representan los recursos de un sistema. Cuentan con una heterarquía, alto nivel de autonomía y coordinación. La colaboración surge de un alto flujo de información entre agentes y el entorno, la comunicación puede ser de tipo indirecta empleando el ambiente, directo intercambio de información entre agentes específicos, sintaxis y semántica determinada, en la Ilustración 9 podemos ver la organización de los agentes y su relación con la inteligencia artificial. [2]

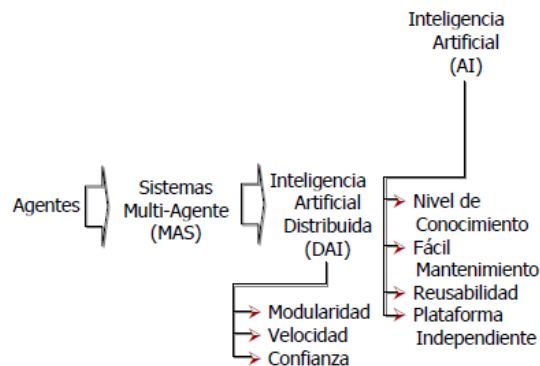


Ilustración 9 Contextualización de los agentes

Arquitectura de sistemas de fabricación holónicos: holarquia sistemas holons organizados dentro de una jerarquía y con posibilidad de cooperación entre si cambiando los conocimientos y habilidades de los holons individuales, lo cual genera robustez en la distribución, el control no está centralizado en un solo elemento, ante una perturbación, perdida no desencadena en inoperancia, parada o reinicio del sistema y como ventaja la adición o retirada o modificación de los módulos de hardware y software mientras el sistema se encuentra funcionando.

➤ Sistemas homogéneos

Es el sistema conformado por robots de una misma estructura y características. Este sistema permite que haya un relevo dentro del mismo, porque pueden ser reemplazados por sus mismos agentes, así mismo pueden trabajar de forma colaborativa.

Los robots modulares hacen parte de estos sistemas homogéneos, los cuales son reconfigurables que comparten información para llevar a cabo tareas específicas, sus características físicas permiten que puedan ser organizados de diferentes maneras para un tipo de necesidad. En la ilustración 10 se ven tres robots homogéneos, en los cuales se busca por medio de la programación su integración y trabajo colaborativo.

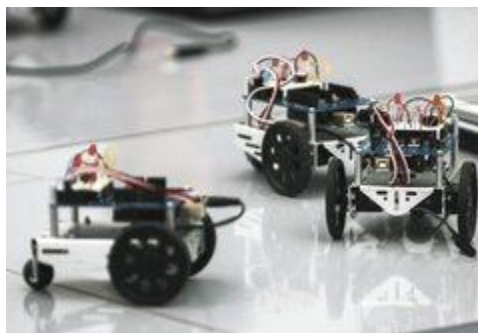


Ilustración 10 Robots móviles homogéneos

➤ Sistemas heterogéneos

Es el sistema conformado por robots que son de diferentes tipos y características. Son robots diferentes tanto físicamente, como en sus comportamientos, el cual puede cumplir una tarea específica, determinados por su parte física (transportar objetos robots móviles, selección de objetos robots manipuladores).

las estrategias para la coordinación, para el trabajo colaborativo de robots pueden ser realizadas por los métodos heurísticos, algoritmo genético, técnicas micro-oportunistas, métodos basados en los mercados, o algoritmos bio-inspirados. Las arquitecturas control y mando pueden ser arquitectura de control jerárquicas (centralizadas), arquitectura heterarquicas (totalmente descentralizadas) perturbación fácilmente superadas por la reconfiguración del sistema. [7]

2.2.5 Inteligencia artificial

La inteligencia artificial ha incursionado en el control de los robots, creando estrategias para controlar los diferentes procesos de la robótica en la actualidad, razón por la cual es de vital importancia conocer las técnicas que ha desarrollado, en la Tabla 3 se identifican los diferentes escenarios en los cuales la integración de la inteligencia artificial y la robótica es tan importante en la actualidad.

Tabla 3 Cuadro que identifica los diferentes escenarios de la Inteligencia artificial y la robótica.

Totalmente observable o parcialmente observable	Sensores toman los aspectos relevantes para la toma de decisiones	Las relevancias dependen de las medidas de rendimiento
Parcialmente observables	Cuando se presenta ruido o sensores que no son muy poco exactos	
Determinista Vs Estocástico	El siguiente estado está determinado, el estado actual y la decisión del agente se dice que el entorno es determinista	Es puede pensar que un entorno es determinista o escolástico, desde el punto de vista del agente
	Si el medio es determinista excepto por la acción de otros agentes, decimos que el medio es estratégico .	
Episódico Vs Secuencial	Son episodios atómicos en los cuales no se basa ni necesita un episodio anterior para tomar una acción o una decisión, en cambio los secuenciales se basa en el pasado para la toma decisiones.	
Estático Vs Dinámico	Los medios dinámicos constantemente están preguntándole al agente que hacer y si mantiene sin hacer nada se determina que el agente tomo la decisión de no hacer nada	Si el entorno no cambia, pero el rendimiento del agente cambia se dice que es semidinámico.

Discreto Vs Continuo	Se aplica al estado del medio y a la forma que se aplica el tiempo, percepciones y las acciones del agente	
Agente individual y multi-agente	Para determinar si es agente o multi-agente tenemos que determinar si el comportamiento de B esta mejor descrito por la maximización de una medida de rendimiento cuyo valor depende de A	

El problema que va a tratar la IA es diseñar el programa del agente que implemente la función del agente que proyecta las percepciones en acciones.

Agente= arquitectura + programa

El programa debe ser acorde a la arquitectura que debe tener el agente, sería ilógico que se programara caminar y la arquitectura no tuviera piernas, son aspectos que se deben tener en cuenta durante el desarrollo de la investigación.

Programación de agentes, empleando la inteligencia artificial:

Tabla de acciones → Cada secuencia posible de percepciones

1 acción → 1 percepción

Si determinamos que necesitamos una tabla de acciones que representa el comportamiento ante las percepciones podemos ver la fragilidad en la cantidad de tablas necesarias para lograr un comportamiento racional adecuado, se debe generar un programa que retome acciones en relación con una cantidad pequeña de datos.

➤ Agentes reactivos simples

Seleccionan las acciones con relación a las percepciones actuales, ignorando las percepciones históricas. Emplea la regla de condición-acción regla si entonces, empleando condicionales que determinan que tipo de acción realizar en el momento que reciba un estímulo ya identificado y programado.

Poseen una inteligencia limitada, basado en su simplicidad, funcionara solo si puede tomar la decisión correcta sobre la base de la percepción actual, lo cual es posible si el medio es totalmente observable.

Si el agente reactivo simple con capacidad de elegir acciones de manera aleatoria puede mejorar el resultado que un agente simple determinista.

➤ Agentes reactivos basados en modelos

La forma de manejar la visibilidad parcial es almacenar información, el estado actual percibido es generado con base en la percepción adquirida y un estado anterior almacenado, genera una conducta que el agente determina para toma de decisión.

➤ Agentes basados en objetivos

A parte de saber el estado actual es muy importante tener una meta o una dirección, lo que hace que el agente pueda tomar decisiones con respecto al objetivo que busca una meta que describa las situaciones que sean deseables con respecto al estado actual y a la meta u objetivo trazado.

➤ Agentes basados en utilidad

La meta por si sola determina un 1 o un 0, resultado que se busca, no permite obtener el mejor camino teniendo en cuenta que pueden ser varios los caminos para llegar a la meta, la pregunta es ¿cuál de todos los caminos es el mejor para llegar a la meta? lo que puede analizarse con la eficiencia, que permita la comparación y evaluación de los estados, obtenida al utilizar los resultados de los diferentes estados, determinando cual es el de mayor utilidad para el agente.

Capítulo III

Diseño mecatrónico

3.1 Metodología para el diseño mecatrónico

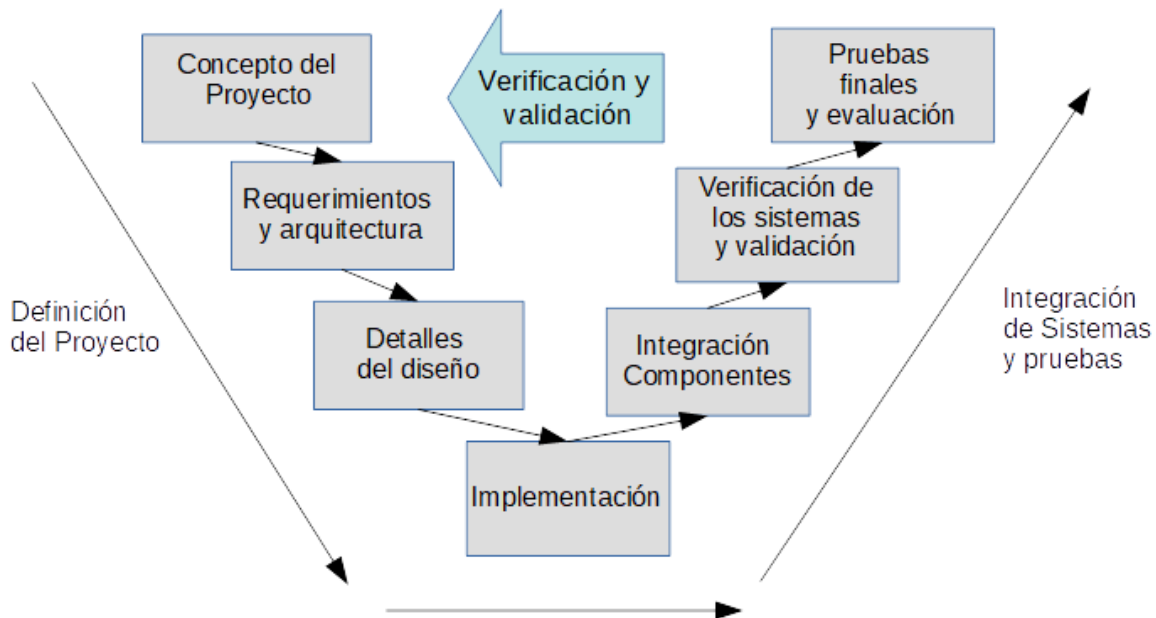


Ilustración 11 Dibujo del Método V [30]

La metodología que se utiliza en el desarrollo del proyecto es el Método V (Ilustración 11). la V está determinada por la verificación y validación, es un método de gestión de proyectos que permite seguir pasos lógicos para el desarrollo del trabajo de grado, es la representación gráfica del ciclo de vida del desarrollo del sistema, se describen las actividades y resultados que deben producirse durante el desarrollo del producto:

Lado izquierdo: Definición del proyecto

- Conceptos operacionales: que debe hacer el sistema a grandes rasgos.
- Requisitos del sistema y arquitectura.
- Diseño detallado.
- Descomposición de las necesidades.
- Creación de la especificación del sistema.

Lado central implementación.

Lado derecho: Integración del sistema.

- Integración de las distintas partes, pruebas y verificación de estas.
- Verificación y validación del sistema en conjunto.
- Mantenimiento del sistema.
- La integración de las piezas y su verificación.

3.1.1 Concepto de Operación

El sistema estará compuesto por tres agentes una agente Supervisor (software) y dos agentes robóticos homogéneos (Robot Móvil Azul y Rojo) y un elemento llamado objeto. El Agente Supervisor identificara los robots y el objeto, con el empleo de la visión de máquina y desarrollara un algoritmo que permitirá identificar sus posiciones y emitirá órdenes para su desplazamiento por medio de una trayectoria llegando a unos puntos de acoplamiento paralelos al objeto, donde se acoplaran y llevaran el objeto a un punto de entrega.

3.1.2 Requisitos del sistema mecánico y estructural.

Requerimientos para el diseño del prototipo:

- El objeto por transportar tiene unas dimensiones de 28 cm x 11 cm x 3 cm (Ilustración 12), el robot debe transportar un elemento de una dimensión inferior a la del robot, pero el objeto a emplear es superior al largo del robot, lo que hace que necesite de otro robot igual para poder transportar los dos el elemento.



Ilustración 12 Objeto

- Debe tener la capacidad de acoplarse al otro robot.
- Debe permitir el movimiento como el otro robot, por lo cual se debe determinar el tipo de robot móvil que cumpla con esta necesidad.
- Debe contar con un sistema de acoplamiento que permita su unión, con el otro robot homogéneo y que sea estable.

- Al tener que unirse al otro robot y acoplarse debe tener la capacidad de moverse de forma coordinada y en diferentes direcciones.

Los materiales y elementos de construcción deben ser comerciales y de fácil adquisición.

3.2 Diseño detallado de la parte mecánica y estructural

El desarrollo del prototipo está directamente relacionada a las necesidades para el cumplimiento de la tarea colaborativa que se desarrollara, el espacio, el área de trabajo, el tipo de robot a diseñar, el acoplamiento de los dos robots, que tendrán la capacidad de transportar un objeto de una mayor dimensión que el robot.

3.2.1 Software de Diseño Mecánico (SolidWorks®)

Después de haber determinado las características del robot móvil heterogéneo, los diseños iniciales, bosquejos relacionados directamente, con los objetivos del proyecto, se determina un robot móvil omnidireccional, en forma de plataforma que permita ser acoplado a otro robot igual de tal forma que se conviertan en una sola plataforma, con el fin de poder transportar un objeto de tamaño mayor al del robot omnidireccional.

Para el diseño de los robots omnidireccionales constan de cuatro partes fundamentales que son: diseño mecánico, diseño electrónico, entorno virtual y diseño de software, por lo cual se necesitará un software de diseño mecánico el cual será SolidWorks®, para el desarrollo de la simulación de los prototipos creados, la implementación del mundo de trabajo, elementos y desarrollo de la tarea colaborativa se empleara el software Webots®, programa que nos permite la creación de robots y su implementación para el desarrollo de Robótica, licencias suministradas por la universidad, empleando la versión 7.4.0.

El software utilizado para la realización de las piezas del Robot Móvil, en el cual se crearán cada una de las piezas mecánicas, partes físicas, ensamblajes y modelos completos del Robot Móvil ideal, es SolidWorks® 2016, este software es empleado ya que fue enseñado y empleado durante el desarrollo de los semestres en la universidad, conocimiento ideal para el desarrollo de este proyecto para la elaboración de todos los planos y prototipos, necesarios para ser implementado en Webots®, verificando su comportamiento y características en el mundo virtual creado para simular la mayor cantidad de tareas que nos permitan determinar la solución más óptima en el desarrollo de esta investigación.

SolidWorks® es un software para el diseño asistido por computador CAD, para realizar el modelamiento de elementos mecánicos, físicos, partes en 3D, para el sistema operacional Windows®, desarrollado por SolidWorks Corp., una filial de Dassault Systemes, S.A. (Suresnes, Francia).

El programa está orientado al desarrollo y modelamiento de piezas, ensambles que pueden ser creadas en 3D, así mismo la obtención de los planos técnicos y los modelos para ser exportados a otros programas para ser utilizado en simulaciones, como van a ser empleado en esta investigación exportando el ensamble del Robot Móvil al programa Webots®, teniendo en cuenta que son dos Robots Móviles, empleando un solo diseño de Robot.

El diseño del robot es inicialmente un robot móvil, de cual se construirán dos robots, que realizarán una tarea colaborativa, que consiste en recoger un elemento para ser transportado a un lugar determinado.

3.2.2 Selección de componentes estructurales de diseño del robot homogéneo

Las diferentes morfologías disponibles de los robots móviles, sus particularidades estructurales que van desde el tipo de rueda, el sistema de tracción, dirección y la forma física del robot, permiten una gran variedad de posibilidades de diseño por lo cual se determinara el escenario y así determinar los componentes a elegir.

Los robots móviles distribuyen generalmente su sistema de tracción y dirección en los ejes de las ruedas que van relacionados directamente con las exigencias de velocidad, maniobrabilidad y características del terreno, es ahí donde la precisión y rapidez para alcanzar el destino u objetivo está determinado por tener un sistema de tracción confiable y un sistema de dirección que permita la maniobrabilidad teniendo en cuenta que el robot debe acoplarse al segundo robot y poder desplazarse y tomar una dirección en el trabajo colaborativo.

➤ Tipo de entorno:

La manera en que los robots van a realizar la tarea colaborativa [22], es interior ya que será desarrollado en los laboratorios de mecatrónica, la característica más predominante es el nivel del suelo, sin tener rugosidades o divisiones que hicieran que el sistema de tracción y dirección sea robusto para que le permitiera el movimiento, la luminosidad de los laboratorios es alta y así mismo no presenta variaciones significativas durante el día.

➤ Tipos de locomoción:

Esta característica está directamente relacionada con el entorno en el cual desarrollara la tarea, al ser terrestre el robot puede tener patas, ruedas y cadenas [22]. Para el desarrollo de la presente investigación y teniendo en cuenta el entorno, puede utilizar ruedas omnidireccionales.

➤ Tipo de ruedas:

En los robots terrestres con ruedas su movilidad está determinada por dos factores que son: tipo de ruedas que poseen y su disposición sobre una estructura mecánica, la rueda que puede cumplir con las necesidades del proyecto es la rueda sueca o rueda omnidireccional, teniendo en cuenta que solo una componente de la velocidad del punto de contacto con el terreno se supone igual a cero durante el movimiento, lo que permite desplazarse de manera perpendicular al plano de la rueda Ilustración 13.

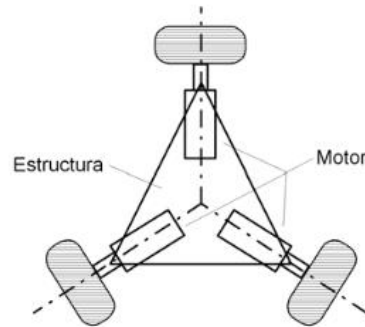
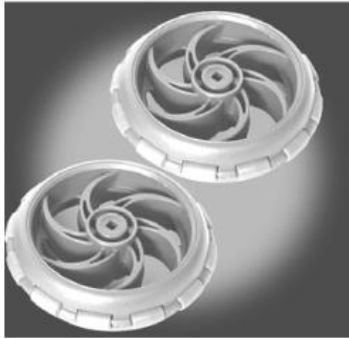


Ilustración 13 Detalle de la rueda sueca y su disposición en un robot omnidireccional tomada de fuente [2].

➤ Disposición de las ruedas:

La combinación de los diferentes tipos de ruedas permite una gran variedad de robots móviles, que les permite diferentes grados de movilidad, se eligió el robot omnidireccional para el desarrollo de la investigación por las siguientes características:

- a. Máxima maniobrabilidad en el plano que permite el desplazamiento en todas direcciones, sin necesidad de reorientarse, lo otros tipos de robot tiene una maniobrabilidad restringida, esta cualidad es muy importante teniendo en cuenta que, debe moverse de forma comunitaria los dos robots. Por lo cual, su maniobrabilidad no estaría limitada durante el movimiento. Ver en la ilustración 13 como se ve la estructura de un robot omnidireccional [23].
- b. En la Ilustración 14 se observa las limitaciones de maniobrabilidad de un robot móvil diferencial, teniendo en cuenta que el acoplamiento de los dos robots se verá afectado en la maniobrabilidad, el objeto que se va a transportar es mayor que los, lo cual sería otro impedimento, porque el movimiento podría ocasionar la caída del objeto y el no cumplimiento de la tarea colaborativa.

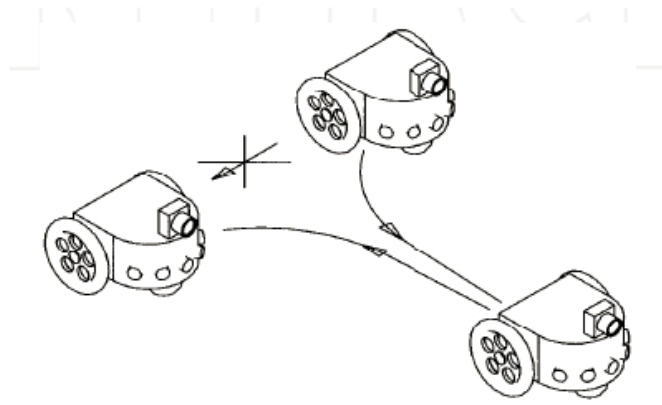
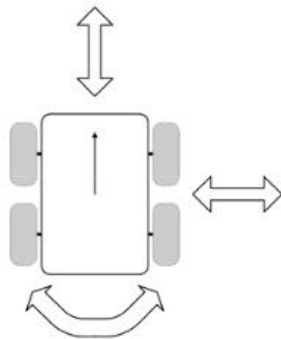
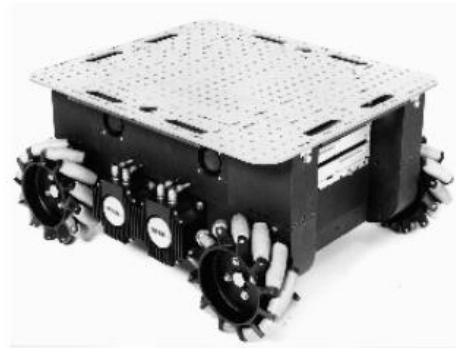


Ilustración 14 Robot diferencial [23].

- c. La maniobrabilidad de los robots omnidireccionales permite el movimiento en cualquier dirección, teniendo en cuenta el tipo de ruedas que emplea y que puede permitir su movimiento perpendicular a la dirección de movimiento de esta, teniendo una configuración de movimiento de 8 direcciones diferentes, así mismo el poder rotar sobre su propio eje. Ilustración 15 (a), el robot omnidireccional Uranus [22] es un ejemplo de configuración y diseño, siendo una plataforma de transporte de materiales, que permite que sea utilizado en cantidad para ese uso, Ilustración 15 (b).



(a)



(b)

Ilustración 15 Robot móvil con ruedas suecas y su maniobrabilidad (a). Robot Uranus (Universidad de Michigan) [22].

➤ Tracción y dirección:

La tracción y dirección esta sobre todos los ejes, esta configuración está orientada a ambientes en donde la velocidad de translación es menos importante, que la adherencia al mismo, el robot omnidireccional escogido presenta una menor complejidad para la solución de errores de odometría.

3.2.3 Diseño del prototipo No 1 Robot Diferencial

El prototipo llamado Robot diferencial 001, no cumple con las necesidades del proyecto, presenta menor complejidad, en su utilización por ser una estructura de robot diferencial comercial, pero su maniobrabilidad se ve comprometida por ser un robot diferencial en comparación con el robot omnidireccional. Ilustración 16.



Ilustración 16 Robot diferencial 001

3.2.4 Diseño del prototipo No 2 Robot omnidireccional

Alguno de los prototipos de robots omnidireccionales encontrados en internet Ilustración 17, el robot (d) es un robot omnidireccional de la empresa Kuka, de tipo plataforma que tiene la función de elevación y de transporte, que puede funcionar por módulos, que pueden acoplarse unos con otros, utilizado para transportar carga pesada.



(a)



(b)



(c)



(d)

Ilustración 17 Diferentes estructuras de Robots Omnidireccionales (a) [26], (b) [27], (c) [24] y (d) [25]

Los robots omnidireccionales (a) y (c) tiene una configuración de cuatro ruedas, el robot (b) de tres ruedas omnidireccionales. Del análisis de los prototipos vistos en la internet, surgen las siguientes posibilidades:

- ¿Qué tipo de rueda se puede emplear que permita tanto el cumplimiento de las tareas como el precio y la facilidad para ser conseguida?
- ¿Qué configuración de las ruedas de los modelos de 3 ruedas o 4 ruedas, cuál de las dos configuraciones nos permitirá cumplir los requerimientos iniciales de los prototipos?

Teniendo en cuenta estos cuestionamientos, al verificar en el mercado su valor y cantidad se pudo determinar que la rueda omnidireccional, no está muy presente en el mercado, encontrando un solo dos lugares en los cuales puede ser adquirida la del modelo (a), con un valor de \$30.000 pesos, siendo la de mayor cantidad y disponibilidad.

a. Modelado CAD de la rueda omnidireccional

La rueda omnidireccional tiene 18 partes diferentes, cada una de ellas será modelada.

1. Estructura: es la parte central de la rueda en la cual va el eje que va al motor Ilustración 18, los pasadores y los rodillos.

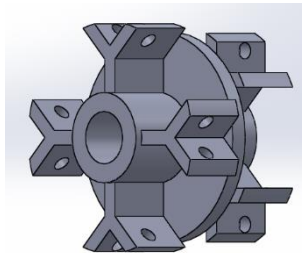


Ilustración 18 Estructura interna de la rueda

2. Rodillo: es el encargado de rotar en el pasador de la rueda, al no estar fijo permite que no genere resistencia cuando se mueve el robot en dirección perpendicular al movimiento de la rueda Ilustración 19, la cantidad de rodillos de la rueda 8.

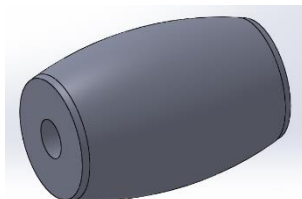


Ilustración 19 Rodillo

3. Pasadores: son en los cuales el rodillo gira permitiendo el desplazamiento Ilustración 20, teniendo una cantidad de 8.

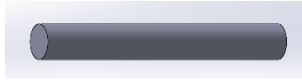


Ilustración 20 Pasador

4. Eje conector: es el eje que entra en la rueda y la une con el motorreductor que moverá la rueda. Ilustración 21.

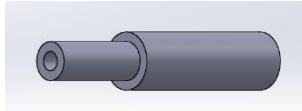


Ilustración 21 Eje conector

La rueda omnidireccional ensamblada, diseñada en SolidWorks®, base de los dos prototipos a desarrollar Ilustración 22.

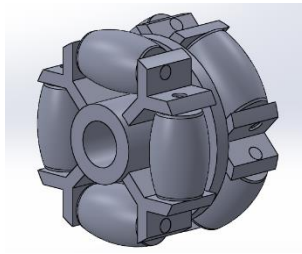


Ilustración 22 Rueda omnidireccional

La rueda omnidireccional ensamblada a el eje conector y al motorreductor en la Ilustración 23.

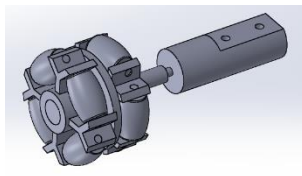


Ilustración 23 Rueda conectada al motorreductor

Ya teniendo las ruedas diseñadas y los motores, se inicia con el diseño de los dos prototipos, la primera configuración será de tres ruedas, el cual se llamará el Robot Móvil 002 y la segunda configuración de cuatro ruedas se llamará Robot Móvil 004, los robots se diseñarán tipo plataforma teniendo en cuenta que:

- Debe llevar objetos de carga que sean menor tamaño al robot, en este caso un robot llevaría el objeto y en caso de que sea de mayor tamaño, emplearíamos el modo acoplamiento para poder llevar el objeto.



Ilustración 24 Robot Híbrido [28]

El Robot omnidireccional plataforma deberá ser completamente plano en la parte superior y contar con orificios en la placa superior para el desarrollo de trabajos futuros como la adaptación de un robot manipulador Ilustración 24.

a. Diseño del Robot Móvil 002

El robot móvil estará conformado por la unión de dos placas, una inferior y otra superior, en la placa inferior estarán anclados los motores y tendrá todos los elementos de la parte electrónica, estarán unidas por unos tornillos a la placa superior, esta soportara los objetos que van a ser transportados, así mismo tendrá orificios para poder anclar elementos como un manipulador, que será determinado en el transcurso de la investigación y desarrollo.

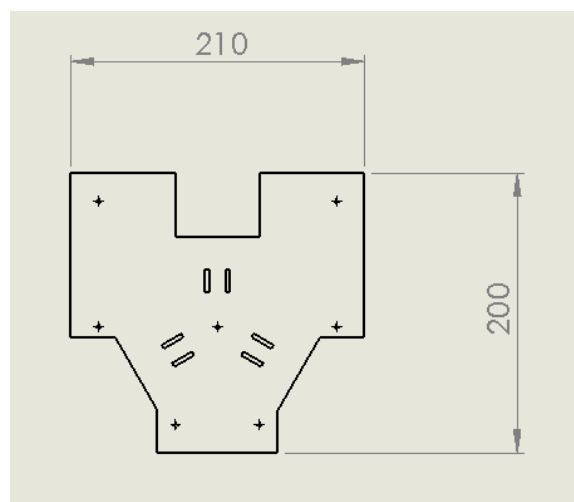


Ilustración 25 Placa inferior

Diseño del prototipo de la placa inferior con unas medidas de 21 cm x 20 cm x 0.3 cm. Ilustración 25, cuenta con los espacios para colocar los soportes de cada motor, la placa superior Ilustración 26 cuenta con las mismas dimensiones y estarán unidas por los tornillos que le darán una distancia entre placas de 4 cm, lugar en el cual estarán los elementos del sistema electrónico.

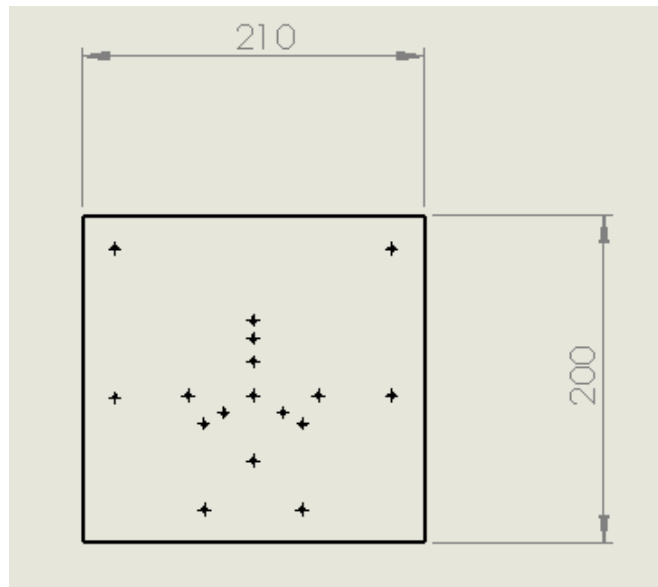


Ilustración 26 Placa superior

Robot Móvil 002, de forma rectangular que permita el acoplamiento con el otro robot homogéneo Ilustración 27 e Ilustración 28, vemos la vista isométrica del Robot.

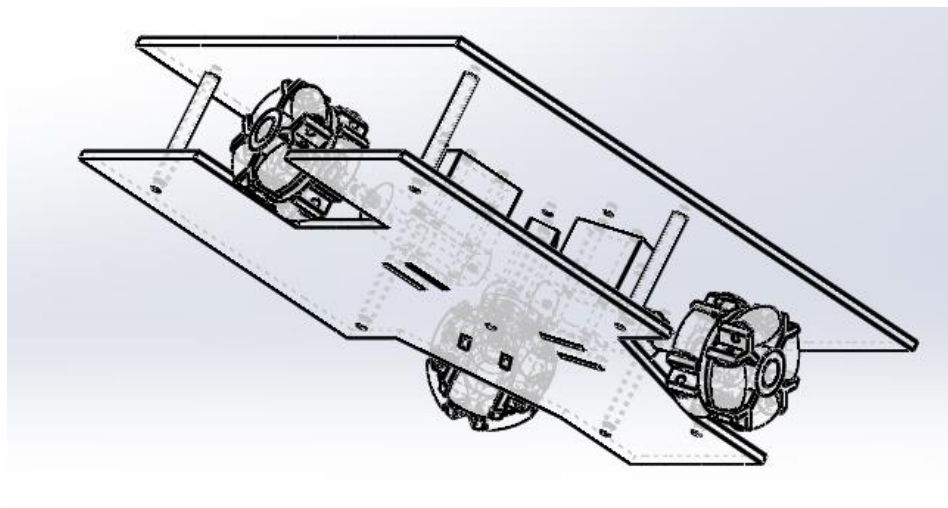


Ilustración 27 Vista isométrica del Robot Móvil 002

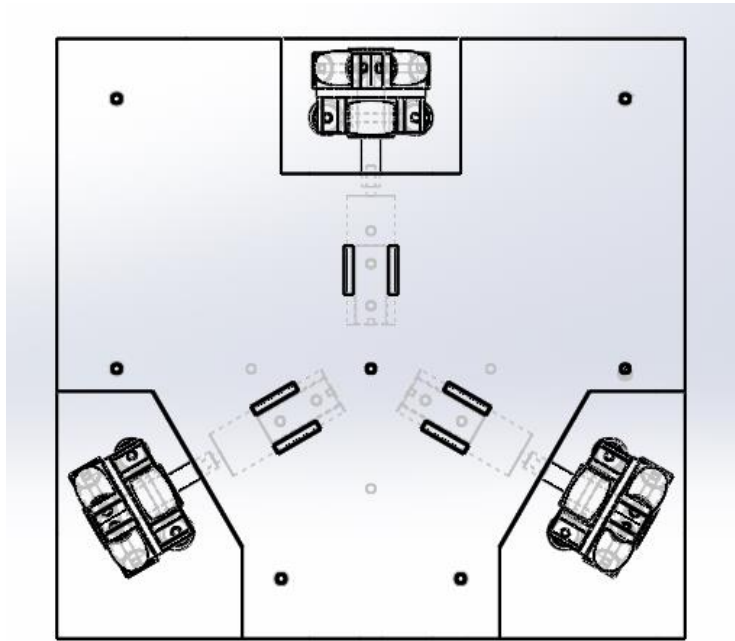
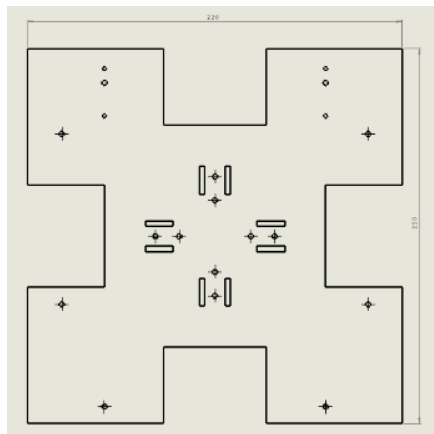


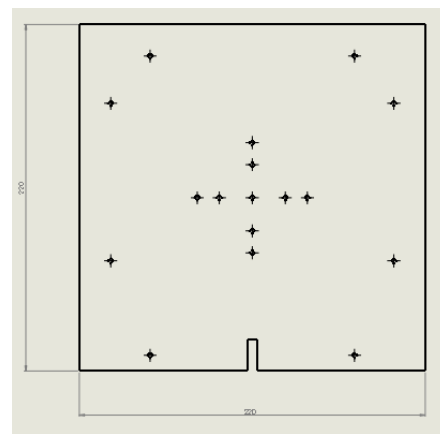
Ilustración 28 Vista inferior del Robot Móvil 002

b. Diseño del Robot Móvil 004

Este prototipo tendrá una configuración de cuatro ruedas, se implementará en Webots® para determinar su comportamiento y respuesta a las necesidades de la investigación. La placa inferior se modificó con relación a la del modelo anterior, por la cantidad de motorreductores y la dimensión de hizo cuadrada con unas dimensiones de 22 cm x 22 cm (Ilustración 29).



(a)



(b)

Ilustración 29 Placa superior (a) y placa inferior (b)

La placa superior estará unida de la misma forma que el prototipo anterior, empleando los tornillos de 4 cm y con los orificios necesarios para anclar un manipulador (ilustración 31), tiene a las entradas para los soportes de los motores, los soportes ajustaran los motorreductores (Ilustración 30), contará con una cantidad de 8 soportes, 2 para cada motor.

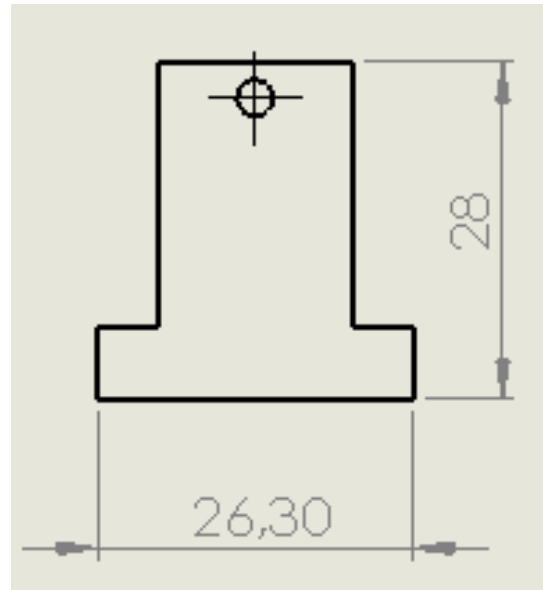


Ilustración 30 Soporte de los motorreductores

El Robot Móvil 004 contará con el sistema de acoplamiento, para poder acoplarse al otro robot homogéneo (Ilustración 31), en medio de sus placas estará los sistemas electrónicos y el sistema de carga.

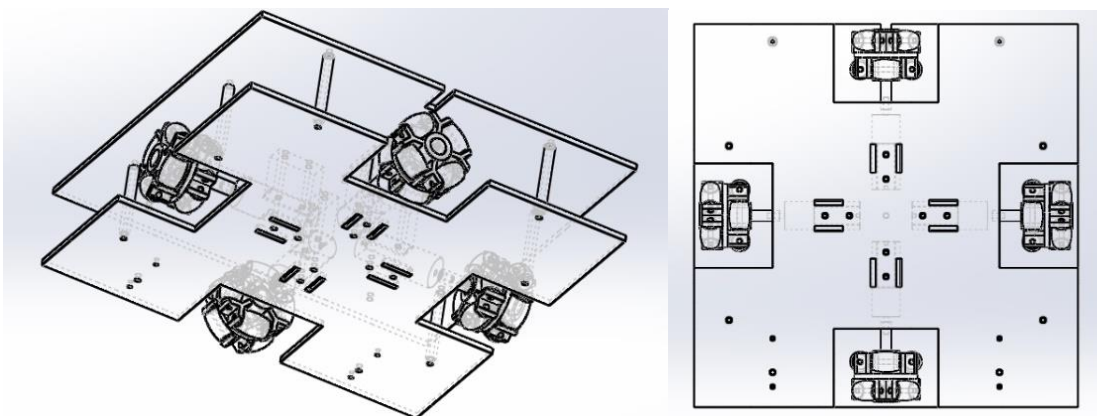


Ilustración 31 Vista isométrica del Robot Móvil 004.

Tabla 4 Cuadro del cálculo de la masa y de la fuerza que ejerce el robot móvil sobre los motores, para determinar el torque necesario de los motores para poder desplazarse a una velocidad angular de 5.2 rad/seg.

No	Elemento	Masa (gr)	Masa (Kg)	Cantidad	Densidad (g/cm ³)	Volumen (Cm ³)	Fuerza (N)
1	Parte en Acrílico 18 y 19	345,576	0,345576	2	1,19	145,2	3,3866448
2	Motorreductores 3	104	0,104				1,0192
3	Uniones	25	0,025				0,245
4	Ruedas	172	0,172	197			1,6856
5	Tarjetas y Accesorio	120	0,12				1,176
	Masa Robot Móvil Total	766,576	0,766576				7,5124448

La fuerza de los elementos del robot, con su respectivo calculo está representada en la Tabla 4. y el cálculo de la fuerza total ejercida sobre los motores Fr está determinada en la ecuación (1). La fuerza total que soportaran los cuatro motores en estado estático es de 7.51 N, valor de la fuerza que soportara cada uno de los motores es de 1,87 N.

$$\sum f_y = -3,38N - 1,01N - 0,245N - 1,68N - 1,176N + Fr = 0 \quad (1)$$

$$\sum f_y = Fr = 3,38N + 1,01N + 0,245N + 1,68N + 1,176N$$

$$\sum f_y = Fr = 7,51N$$

3.2.5 Analisis de la estructura por elementos finitos

La estructura será sometida al estudio de su comportamiento por el método de elementos finitos, empleando una carga máxima de 3 N, que sería la fuerza máxima que ejercería el objeto a ser transportado en el robot móvil (Ilustración 32).

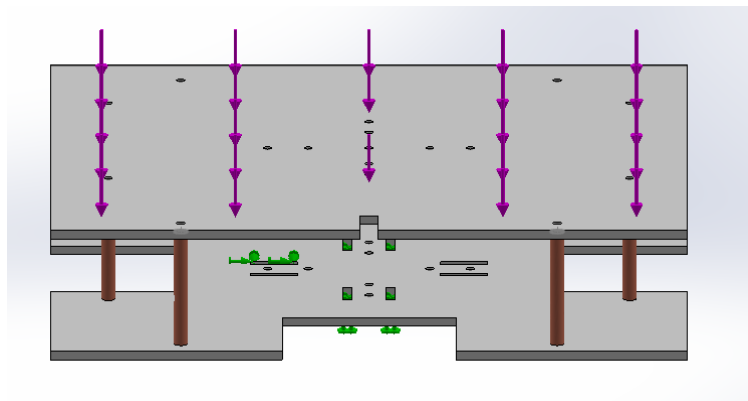


Ilustración 32 Estructura para la realización de estudio de elementos finitos

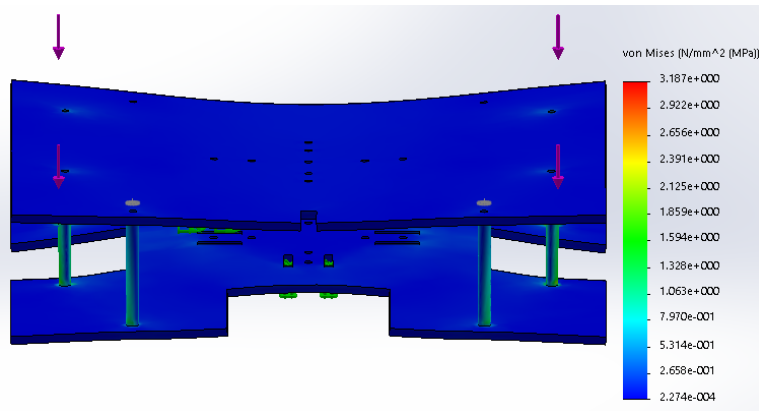


Ilustración 33 Estructura sometida al estudio de elementos finitos

La estructura fue sometida a una carga máxima de 3 N, comprobando su comportamiento en el análisis estático, llegando a un desplazamiento máximo de 0.000143687 m, cumpliendo con las necesidades del sistema en cuanto a resistencia de la estructura a la carga ejercida sobre ella, lo cual determina la carga que puede transportar (Ilustración 33).

La tensión mínima 0,000227441 N/mm² y máxima de 3,18748 N/mm², representadas en la escala de la ilustración 33, representado la tensión ejercida de la carga en la estructura.

3.3 Diseño detallado parte electrónica del Robot Móvil 004.

Requerimientos del sistema electrónico:

- Tarjeta de control para 4 motores, un módulo de comunicación, 4 sensores de contacto y tarjeta de potencia.
- 2 sensores de contacto para la detección del otro robot.
- 2 sensores de contacto para el sistema de pinza para el acoplamiento.
- 4 motorreductores para cada robot que puedan mover 7.6 N.
- Módulo de comunicación para cada robot, que tenga un rango de distancia mínima de 5 m y permitiendo la comunicación con el agente software interfaz de usuario.
- Sistema de baterías que permita la autonomía de cada uno de los robots, con un tiempo mínimo de 20 min que alimente la tarjeta de desarrollo, con sus sensores y los 4 motores.

Los componentes electrónicos están divididos en cuatro partes inicialmente, el controlador, los sensores, el módulo de comunicación y los actuadores. En la parte del controlador del sistema tiene como elemento principal la tarjeta de desarrollo, en la parte de los sensores

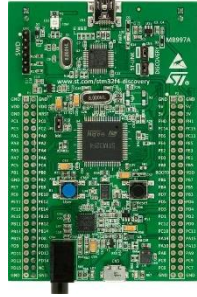

son los encargados de recibir la percepción del mundo en el cual el robot este determinando ya sea reactiva o deliberativamente su comportamiento, el módulo de comunicación enlazará los agentes del sistema y los actuadores son los encargados de ejecutar las órdenes del controlador.

3.3.1 Tarjeta de desarrollo

Es la encargada en este caso de recibir los datos de los sensores, desarrollar un programa ya compilado y ejecutar las ordenes mediante los actuadores, la elección de la tarjeta de desarrollo por las necesidades del proyecto y por el tiempo determinado, buscando las siguientes características:

- Que pueda utilizarse de forma rápida, que permita el prototipado rápido, lo que también nos indica una curva de aprendizaje rápida, para no demorar en conocer su funcionamiento.
- Los complementos necesarios sean compatibles con ella, que no necesitemos realizar acondicionamientos electrónicos complicados para su utilización.
- Su consumo sea bajo teniendo en cuenta que tiene que estar acoplada a 4 motores y varios sensores, teniendo en cuenta que lo ideal del sistema es que sea inalámbrico para que los robots puedan moverse libremente.
- El lenguaje empleado por la tarjeta sea de fácil aprendizaje, así mismo cuente con librerías necesarias para los diferentes módulos que se van a emplear.
- Que permita la conexión a bluetooth de forma fácil, teniendo en cuenta que va a estar en constante contacto con el computador.
- El precio sea asequible teniendo en cuenta que se necesitan un total de 2 tarjetas para los robots, por lo cual el precio influye directamente en su escogencia.

Tabla 5. Cuadro comparativo de las tarjetas de desarrollo consultadas para el proyecto

		
Procesador	ARM 32-bit Cortex™-M3 CPU Core	ATmega328 de la marca Atmel®
Frecuencia	72 MHz (1.25 DMIPS/MHz)	16MHz de velocidad
Memoria Flash	64k	32k
Entradas y salidas	26 entradas y salidas digitales, la mayoría tolerantes a 5V	14 pines digitales de I/O (6 salidas PWM).
Valor	\$ 119.000	\$ 30.000

Con relación a las necesidades mencionadas, la tarjeta seleccionada fue la Arduino® Uno (Ilustración 34 Arduino Uno [29]), de acuerdo con la Tabla 5. Cuadro comparativo de las tarjetas de desarrollo consultadas para el proyecto de comparación, que cumple con las características, es una tarjeta que utiliza el microcontrolador ATmega328 de la marca Atmel® , con sus diferentes acondicionamientos, reguladores de tensión, puerto serial, un puerto USB, el cual permite la conexión al PC y poder así programarlas, cuenta con librerías necesarias, cuenta con 14 pines, de los cuales tiene 6 entradas análogas, 14 pines digitales I/O, de los cuales 6 son salidas PWM, voltajes de alimentación de 7- 12 V, un reloj de 16 MHz de velocidad.



Ilustración 34 Arduino Uno [29]

3.3.2 Sensores de contacto

Es un sensor que produce una señal digital a partir del contacto que puede ser un LOW o un HIGH Ilustración 35 Diagrama de conexión del sensor [29], que será recibido por la tarjeta de desarrollo en los puertos análogos, teniendo en cuenta que solo quedaron disponibles ese puerto, los demás están ocupados por la comunicación serial y el control de los motorreductores.

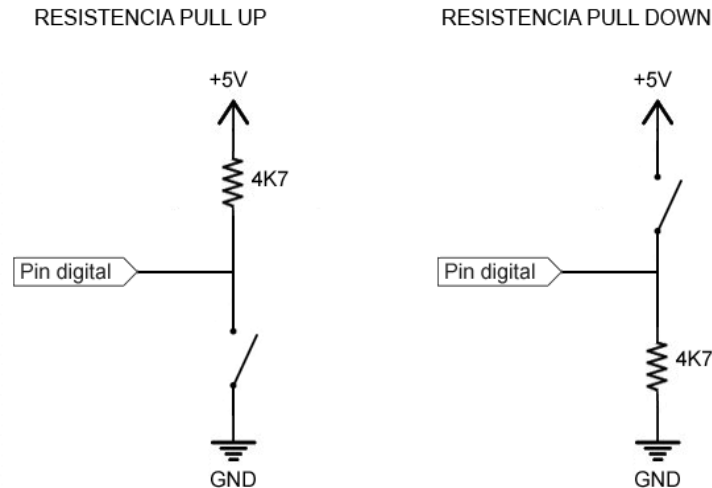


Ilustración 35 Diagrama de conexión del sensor [29]

El sensor cambiará de estado, empleando las resistencias de pull down, enviando una señal de 5V a la entrada análoga del Arduino, este sensor será utilizado en el sistema de acoplamiento para:

- Corregir la dirección del robot que realizar el acoplamiento, detectando el desplazamiento hacia la derecha o hacia izquierda, tomando la decisión el controlador de corregir la dirección por medio de los actuadores, teniendo en cuenta que para el robot de acoplamiento necesita está detrás completamente del robo que va a ser acoplado.
- Detectar el contacto entre las dos placas, tendrá dos sensores que enviaran la información al controlador y así accionara el sistema de acoplamiento que está compuesto de dos servomotores, cada uno de ellos provisto de una pinza que se anclara al robot y mantendrá esa posición durante el periodo de movimiento colaborativo.

Los sensores de contacto son de fácil obtención y con un valor económico, para su acondicionamiento, solo necesita una resistencia de 4.7k ohm.

3.3.3 Módulo de comunicación

Los módulos disponibles en el mercado que permiten la comunicación de los agentes del sistema en un radio de distancia superior a 5 m están en la Tabla 6 Cuadro comparativo de módulos de comunicación para los robots. Las características de comparación son: el valor, el sistema de comunicación que emplea y el alcance, el módulo seleccionado es el HC-05.

Tabla 6 Cuadro comparativo de módulos de comunicación para los robots

		
Nombre	NodeMcu Lua WIFI	HC-05
Protocolo de comunicación	WIFI	Bluetooth v2.0
Frecuencia	80 MHz	2.4 GHz
Memoria Flash	64k	32k
Entradas y salidas	6 pines GPIO	2 pines de transmisión y recepción
Alcance	25 m	5m a 10 m
Valor	\$ 24.000	\$ 18.000

La red de comunicación del proyecto está compuesta de dos robots y el computador que tiene el programa supervisor, para los robots se utilizara módulos Bluetooth (ilustración 32), el computador tiene uno y los módulos de comunicación que vende para las tarjetas de desarrollo son los HC-05, la comunicación serial que emplea Arduino inicialmente utiliza el código ASCII.

3.3.4 Actuadores

➤ Motorreductor

Son los actuadores del robot que son controlados por la tarjeta de desarrollo, que responde a las acciones recibidas del programa supervisor, durante el desarrollo del programa

principal en sus capas deliberativa y reactiva. La fuerza que soportaran los motores será aproximadamente de 7.6 N, por lo cual se escogieron motorreductores de torque de 4 kg-cm, con un valor por motor de \$25.000 pesos (Ilustración 36 Motorreductor).



Ilustración 36 Motorreductor

Características del motorreductor

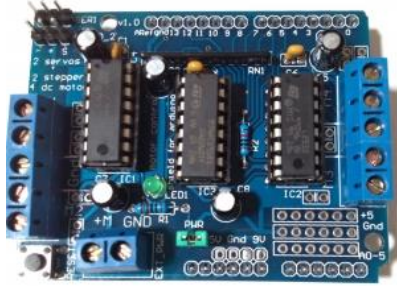
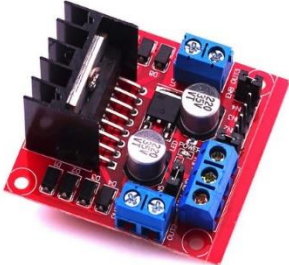
- Torque 4 kg-cm.
- Voltaje máximo 12 V.
- 100 rpm.
- Diámetro del eje 3mm.
- Longitud 47.40 mm.
- Diámetro máximo 15.45 mm.
- Corriente máxima 1.6 A.
- Corriente mínima 0.2 A con un voltaje de 5 V.

➤ Tarjeta controladora de motores L293.

Las tarjetas controladoras de motores consultadas que están presentes en el mercado cumplen con los requerimientos del proyecto Tabla 3, de las cuales la que presenta mejores características es la L293, la cual permite la conexión de los 4 motorreductores y 2 servomotores necesarios para cada robot.

Es una tarjeta que se viene diseñada para Arduino, lo cual hace que sea fácilmente conectada, cuenta con una librería para Arduino que permite controlar los motores de forma fácil, eligiendo el motor a utilizar y tres tipos acciones que puede ejecutar, el motor puede estar quieto o girar a la derecha e izquierda, así mismo se puede controlar la velocidad de cada motor por medio de PWM de 0 a 255, cuenta con dos salidas para servomotores que son controlados con la librería.

Tabla 7 Cuadro comparativo de módulos los controladores de los motores.

		
Nombre	Tarjeta L293	L298N
Voltaje máximo ext.	24 V	24 V
Corriente máxima	1.2 A	2 A
Conectores Servomotores	2	0
Conectores motorreductores	4	2
Valor	\$ 18.000	\$ 15.000

Características técnicas

- 2 conectores para servomotores.
- 4 conectores para motorreductores bidireccionales DC con una velocidad de 8 bits de resolución.
- Con una entrada de voltaje externo de máximo 24 V.
- Soporta hasta 0.6 A por canal (1.2 A pico).
- Pull down para mantener los motores deshabilitados en el encendido.
- Botón de reset compatible con Arduino.

Es una tarjeta controladora de motorreductores adecuada a las necesidades de los actuadores que van a ser empleados en cada robot, dos servomotores y cuatro motorreductores.

➤ Selección de las baterías

En cuanto la elección de las baterías es de vital importancia si se quiere que los robots funcionando de forma autónoma, pueda moverse de forma libre sin las limitaciones del cableado, así mismo el tiempo de ejecución de la tarea colaborativa, tendrá un alto impacto en el paso, se escogieron batería de litio comerciales AA, con un voltaje por batería de 1.2 V con una corriente de 2.3 A.

Cálculos de las baterías

$$I = 0.5 A \text{ por motor}$$

Para los 4 motores por robot se necesitaría

$$I = 0.5 A * 4$$

$$I = 2 A$$

Solamente para alimentar la tarjeta de control de los motores necesitaríamos que las baterías suministraran una corriente de 2 A, las 4 baterías AA recargables nos darían una corriente mínima de 2.3 A, lo cual supliría la necesidad de corriente de los motores, teniendo en cuenta que solo las baterías suministrarían voltaje a los motores, necesitaríamos otras cuatro baterías para suministrar los 50 mA de la tarjeta Arduino.

3.4 Diseño del algoritmo colaborativo

3.4.1 Percepción visual (*visión de maquina*)

La percepción visual es la encargada de extraer de la imagen todos los elementos de interés (el objetivo y los dos robots) del stream de video obtenido de la cámara. Se implementó la librería de Matlab® y su función umbraliza para simplificar el procesamiento de la imagen, debido a la facilidad de su manejo, documentación disponible y por ser utilizada en anteriores proyectos. Teniendo en cuenta que los robots son de diseño homogéneo, se ubicaron en su tapa superior hojas de colores para su identificación, para identificar su orientación se utilizaron dos colores más para hacer los marcadores que serían identificados y al determinar el ángulo entre el centroide del robot y el centroide del marcador determinar su orientación. El objeto tiene forma rectangular identificado por un color y con un marcador de otro color para poder identificar su ubicación y orientación.

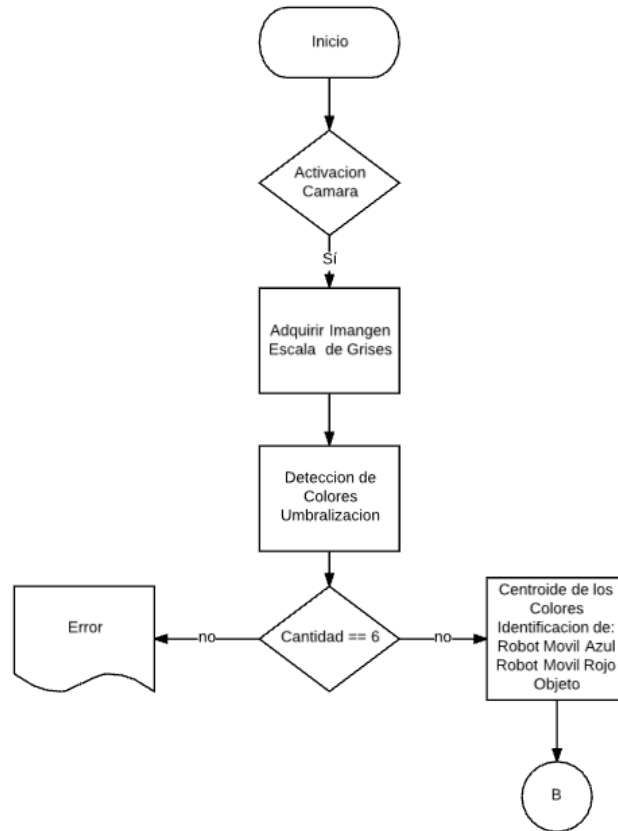


Ilustración 37 Diagrama de flujo para componente de vision

En la Ilustración 37 Diagrama de flujo para componente de vision

está representado el diagrama de flujo del componente. La cámara captura 30 *frames* por segundo (fps) a una resolución de 640 por 480 pixeles. La altura de la cámara con relación al entorno de trabajo es de 1,70 m, determinando unas dimensiones de 1,14 m por 0,86 m donde cada pixel tiene un tamaño equivalente a un cuadrado de 0.17 mm, teniendo una resolución alta para el trabajo a desarrollar. El algoritmo de toma de imagen se desarrolla con un intervalo de 0.6 seg, tiempo que permite que el programa no presente bloqueos durante su ejecución. Al tomar la imagen el programa la convierte a escala de grises e inicia la umbralizacion con respecto a los valores antes determinados para cada color, son seis valores diferentes que fueron determinados bajo unos parámetros de luminosidad determinado.

Al tener identificado cada uno de los colores, el algoritmo determina los centroides identificando su ubicación que inicialmente está determinada por el pixel y que posteriormente es convertida a metros (1). Esta información es la necesaria para iniciar con el desarrollo del programa.

$$\begin{aligned}
 function [pX, pY] &= Interpolar(posX, posY) \\
 pX &= (posX * 114) / 640;
 \end{aligned}
 \tag{1}$$

$$pY = (posY * 86)/480;$$

3.4.2 Orientación de los robots y objeto

➤ *Orientación de los Robots*

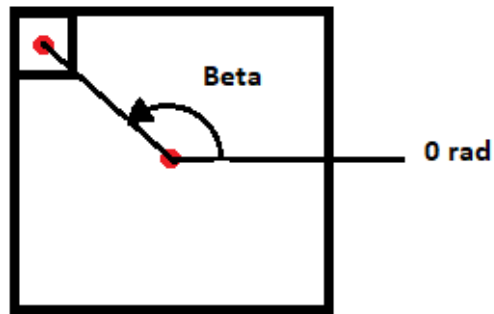


Ilustración 38 Orientación del robot

La orientación del robot es obtenida por medio de la imagen tomada por la cámara, en la cual se identificaron el color del robot y del marcador, posteriormente se identificó sus centroides determinando los puntos para hallar el ángulo de orientación ecuaciones (2), los puntos identificados son:

- *PR* es el centroide del robot.
- *PM* es el centroide del marcador del robot.

$$\begin{aligned}
 &PR(x1, y2), PM(x2, y2) && (2) \\
 &\Delta y = (y2 - y1) \\
 &\Delta x = (x2 - x1) \\
 &\gamma = \tan^{-1} \frac{\Delta y}{\Delta x} \\
 &\beta = \alpha + \gamma
 \end{aligned}$$

El ángulo β (Ilustración 38), debe ser ajustado al desfase que presenta la ubicación del marcador, el ángulo γ está determinado por la ubicación de los centroides y α es el ángulo desfasado ecuación (3).

$$\alpha = \pi/4 \quad (3)$$

➤ *Orientación del objeto*

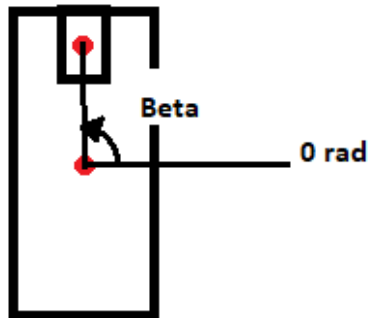


Ilustración 39 Orientación del objeto

La orientación del objetivo es igual a la forma en la cual se obtuvo la de los robots, los puntos identificados son:

- *PO* es el centroide del robot.
- *PM* es el centroide del marcador del robot.

Las ecuaciones para determinar los puntos son (4), suministrando los datos necesarios para el desarrollo del algoritmo.

$$\begin{aligned} PO(x1, y2), PM(x2, y2) & \quad (4) \\ \Delta y &= (y2 - y1) \\ \Delta x &= (x2 - x1) \\ \beta &= \tan^{-1} \frac{\Delta y}{\Delta x} \end{aligned}$$

El ángulo β , es determinados como indica la ecuación anterior, teniendo en cuenta que el marcador está ubicado en la mitad de la parte superior del objeto (Ilustración 39).

3.4.3 Corrección del ángulo

Esta función está determinada por los siguientes requerimientos:

- El robot es omnidireccional, siendo un robot que no tiene restricciones de movimiento.
- El robot debe mantener su orientación el ángulo de $\pi/2$, durante su desplazamiento.
- El Robot determina su orientación por medio de la visión de maquina suministrada por el agente Supervisor (software).

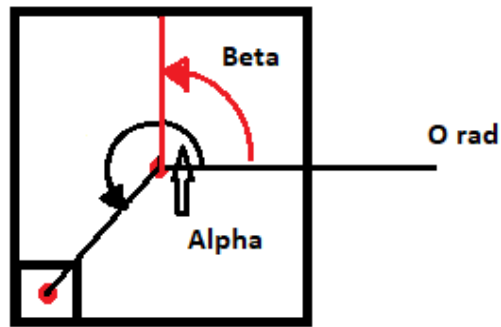


Ilustración 40 Orientación de robot

En la Ilustración 40, se verifica la orientación del robot, de cual se determinan 2 ángulos diferentes:

1. Ángulo Alpha: formado por la semirrecta del centroide del marcador y el centroide del robot y la semirrecta del centroide del robot al origen del plano, el ángulo real del robot, determinado por el procesamiento digital de imagen.
2. Ángulo Beta: es el ángulo ideal que debe tener el robot para su desplazamiento y orientación con respecto al plano, ángulo ideal.

El programa Supervisor monitorea los ángulos de los dos robots durante su desplazamiento, empleando la función MDelta ecuación (5), teniendo en cuenta que se mueve omnidireccionalmente, puede mantener un mismo ángulo, esta función compara el ángulo real con el ángulo ideal determinando un margen de 0.07 rad, para iniciar la corrección de la orientación.

$$\begin{aligned}
 & \text{Funcion MDelta}(angIdeal, angReal) && (5) \\
 & \Delta = angIdeal - angReal \\
 & \text{if } \Delta > 0.07 \parallel \Delta < -0.07 \\
 & \text{Funcion CorregirAngulo}
 \end{aligned}$$

El diagrama de flujo de la Ilustración 41 determina como el programa se activa ante la función MDelta (5), verificando el ángulo del robot por medio del agente Supervisor (software) y tomando la acción que determine para su corrección.

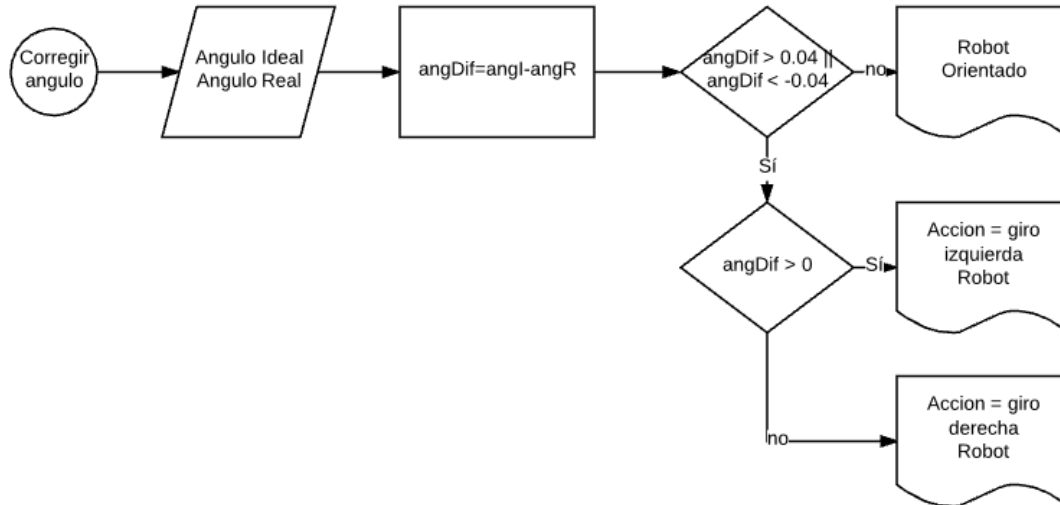


Ilustración 41 Diagrama de flujo de subprograma corrección de ángulo

3.4.4 Posición de acoplamiento

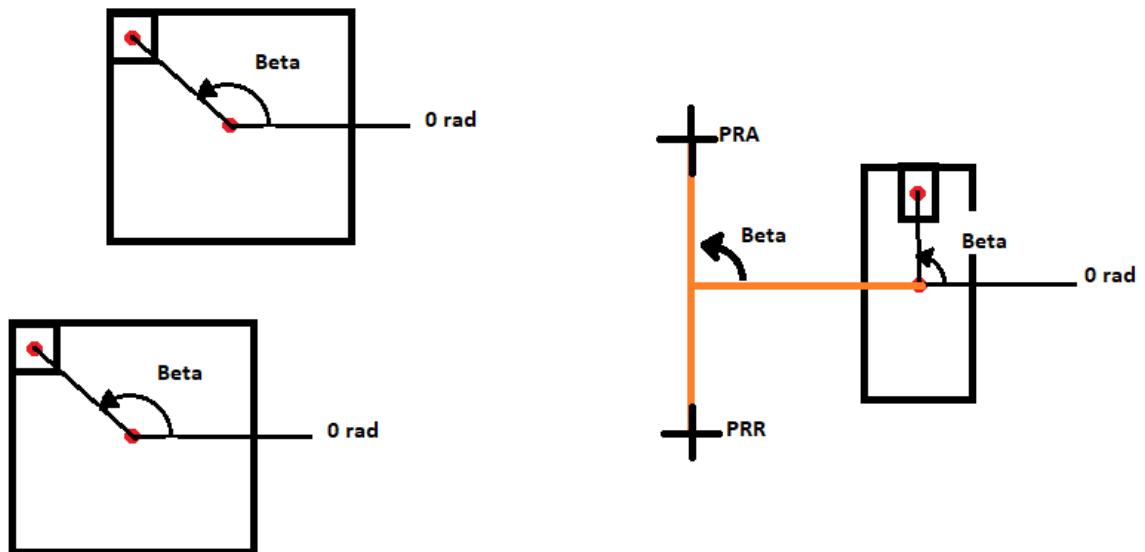


Ilustración 42 Robot y objeto en posición de inicio

Los robots deben llegar a unas posiciones de acoplamiento para que se unan haciendo una plataforma para recibir el objeto (Ilustración 42). Teniendo las posiciones iniciales se determina la distancia hacia el objeto, a partir de ahí se deben tener los siguientes requerimientos:

- Determinar el ángulo del objeto: el ángulo determinara las dos posiciones de acoplamiento y la orientación de los dos robots.
- Determinar las dos posiciones de acoplamiento de los robots, que cuenta con una distancia prudencial para evitar que las trayectorias de cada robot se crucen.
- Las posiciones de acoplamiento deben ser empleadas como puntos de llegada, al llegar a los puntos deben orientarse paralelos al objetivo para que puedan ser cargados, e iniciar con la subrutina de acoplamiento.

Los datos suministrados por la visión de maquina son: La posición de los robots y su correspondiente marcador: posición del robot Azul $PRA (x1, y1)$, marcador del robot Azul $PMA (x2, y2)$, posición del robot Rojo $PRR (x3, y3)$, posición del marcador del robot Rojo $PMR (x4, y4)$ y posición del objeto $PO (x3, y3)$ con su marcador $PMO (x6, y6)$.

La función puntos de acoplamiento, recibe del Supervisor el punto de ubicación del objeto $PO (x3, y3)$ y su ángulo de inclinación, empleando la función $pol2cart$ (ángulo, distancia) de Matlab®, con un ángulo fijo y una distancia fija la ubicación del punto base para que, a partir de ahí obtener los dos puntos de acoplamiento de los robots, puntos que determinaran la trayectoria que deben tomar, como está representado en el diagrama de flujo y las funciones que a continuación se emplean, teniendo como base la distancias fijas y ángulos determinados ecuación (6):

1. Pmx y Pmy ángulo π , distancia de 23,7 cm.
2. PAX y PAY ángulo β , distancia de 17,9 cm.
3. PRX y PRY ángulo ϑ , distancia de 17,9 cm.

$$\vartheta = \beta + \pi \quad (6)$$

Las funciones que se presentan a continuación (7,8,9) determinan, la obtención de los puntos y como se realimenta la función para obtenerlos, ubicados en la representación de la ilustración 7.

$$\begin{aligned} [Pmx, Pmy] &= pol2cart(\pi, dis1) & (7) \\ PmX &= Ox + x \\ PmY &= Oy + y \end{aligned}$$

$$\begin{aligned} [PAX, PAY] &= pol2cart(\beta, dis2) & (8) \\ PmX &= Pmx + x \\ PmY &= Pmy + y \end{aligned}$$

$$\begin{aligned} [PRX, PRY] &= pol2cart(\vartheta, dis3) & (9) \\ PmX &= Pmx + x \\ PmY &= Pmy + y \end{aligned}$$

El diagrama de flujo (Ilustración 43) representa, la ejecución del programa de la obtención de los puntos de acoplamiento, determinando los puntos finales para el desarrollo de la trayectoria de los robots.

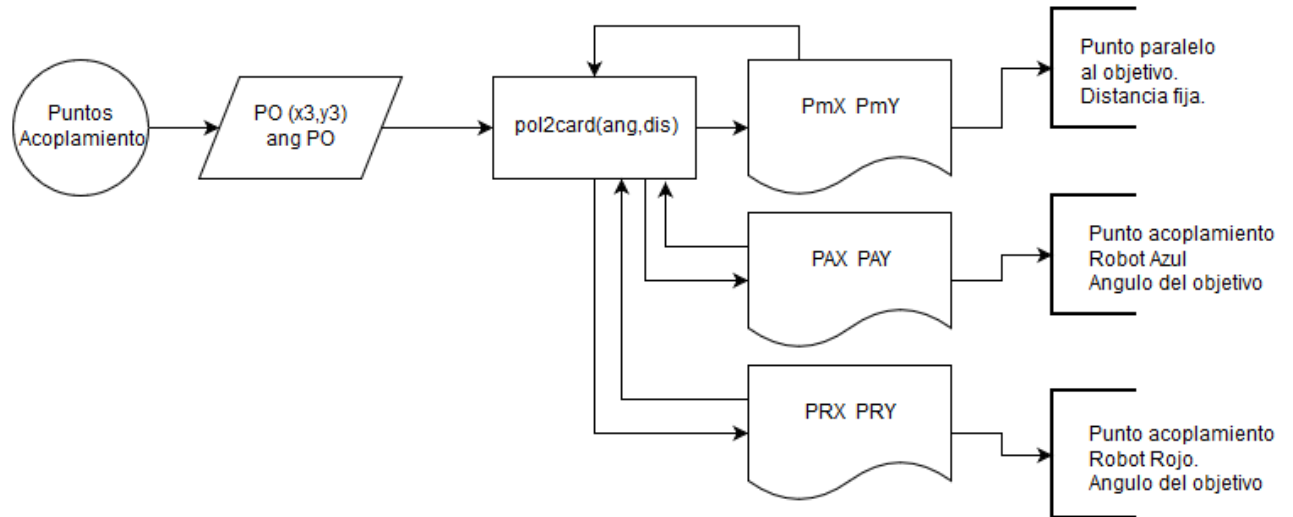


Ilustración 43 Diagrama de flujo de los puntos de acoplamiento

3.4.5 Subprograma de acoplamiento

La función está determinada por los siguientes requerimientos:

- Los robots deben estar orientados igual que el objeto a ser cargado.
- El acoplamiento debe ser organizado, por lo cual un solo robot se desplazará y acoplará al otro robot.
- Se debe verificar el acoplamiento y la unión de los dos robots.
- Debe ser detectado el objeto, para el transporte del objeto.

Los robots cuentan con 4 sensores de contacto, 2 están en la base de la plataforma inferior para que al hacer contacto con la otra plataforma acciones las pinzas y queden anclados y 2 en las pinzas para corregir la ubicación del robot en caso de que no estén completamente alineados para la realización del acoplamiento. La Ilustración 44 muestra los robots en los puntos de acoplamiento respectivamente y como deben estar orientados con respecto a la orientación del objeto a transportar.

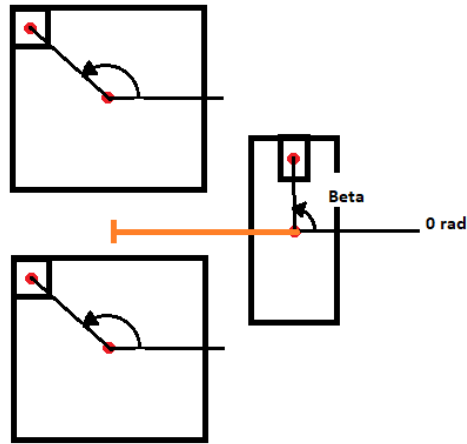


Ilustración 44 Robots en posición de acoplamiento

El acoplamiento de los robots es realizado por un sistema mecánico y electrónico conformado por 2 servomotores y dos pinzas, que de acuerdo con la programación se cerrarán acoplándose, en la Ilustración 45 se observa cómo deben quedar los dos robots acoplados, para la recepción del objeto e iniciar el movimiento colaborativo transportando el objeto a un punto de entrega.

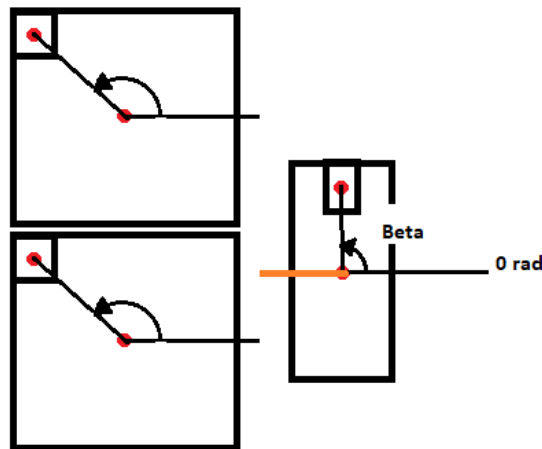


Ilustración 45 Robots acoplados

El diagrama de flujo de la Ilustración 46, está determinado por las señales de los sensores de contacto del robot que acoplara, el cual es el único que se moverá y empleando los pulsadores podrá determinar si está alineado y si ya se acoplo.

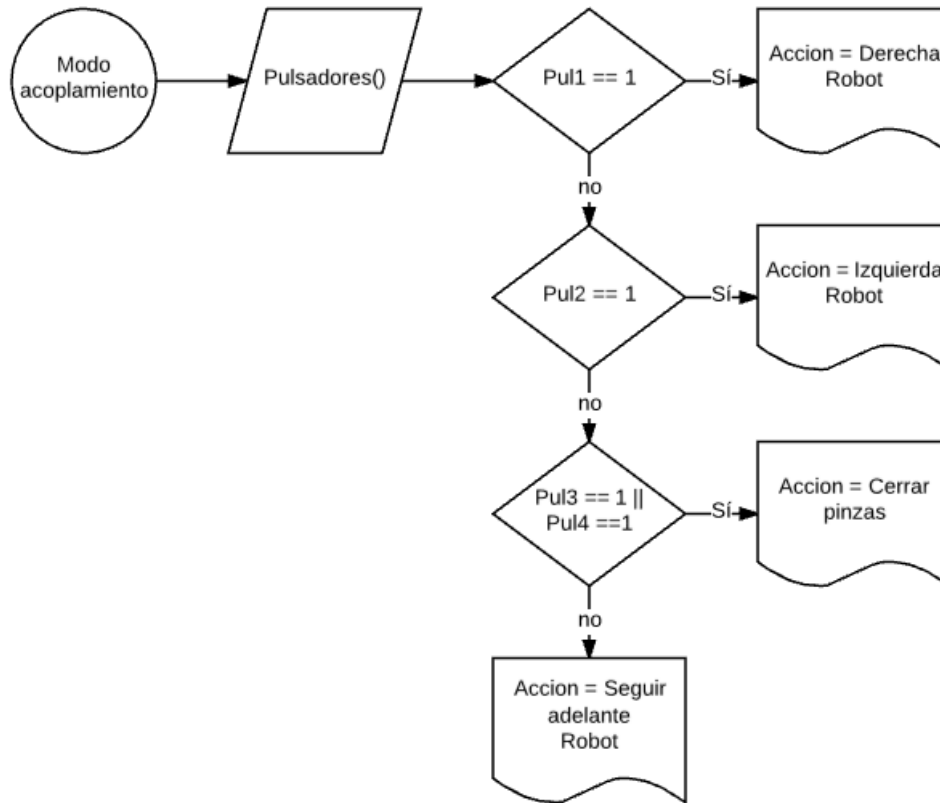


Ilustración 46 Diagrama de flujo del subprograma de acoplamiento

3.4.6 Subprograma de la trayectoria de los robots al objetivo

Los robots deben desplazarse hacia los puntos de acoplamiento, estas trayectorias no deben cruzarse teniendo en cuenta que los robots no cuentan con sensores que les permitan identificar obstáculos, ellos son orientados por el agente Supervisor el cual emplea la visión de máquina para determinar su ubicación y determinar la trayectoria que debe tomar al punto de acoplamiento.

Los datos suministrados por la percepción visual (visión de maquina) son: la posición de los robots y sus puntos de acoplamiento, posición del robot Azul $PRA(x1,y1)$, posición del robot Rojo $PRR(x3,y3)$ y posición del objeto $PO(x3,y3)$, con las posiciones de acoplamiento Robot Azul $PA(PAX,PAY)$ y Robot Rojo $PR(PRX,PRY)$, esta información es utilizada para determinar la distancia que existe entre cada uno de los robots con su respectivo punto de llegada, como está representado en la Ilustración 47.

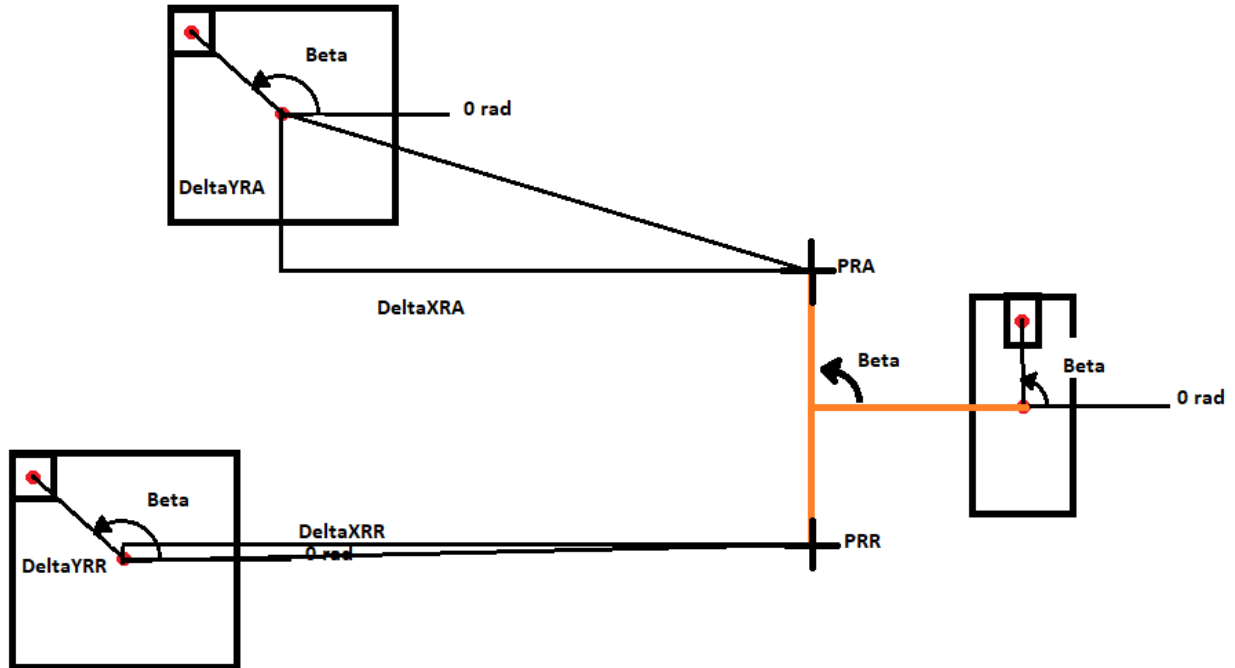


Ilustración 47 Distancia de los robots a sus puntos de acoplamiento

Se obtiene de la posición de llegada y la del robot una diferencia de distancias tanto en X como en Y llamados DeltaX y DeltaY, que para cada robot son:

1. Robot Azul: DeltaXRA y DeltaYRA.
2. Robot Rojo: DeltaXRR y DeltaYRR.

Representados en las ecuaciones 10, este valor determinara la orientación de los robots en el desarrollo de su trayectoria.

$$\begin{aligned}
 \Delta X_{RA} &= PAX - x_1; \\
 \Delta Y_{RA} &= PAY - y_1; \\
 \Delta X_{RR} &= PRX - x_3; \\
 \Delta Y_{RR} &= PRY - y_3;
 \end{aligned}
 \tag{10}$$

En la Ilustración 48, se determina el comportamiento de la función que desarrolla la trayectoria de cada uno de los robots, determinado por el valor de la variable Delta tanto en X como en Y, implementando los tipos de movimiento del robot omnidireccional, en relación con la diferencia de su punto de ubicación y el de acoplamiento.

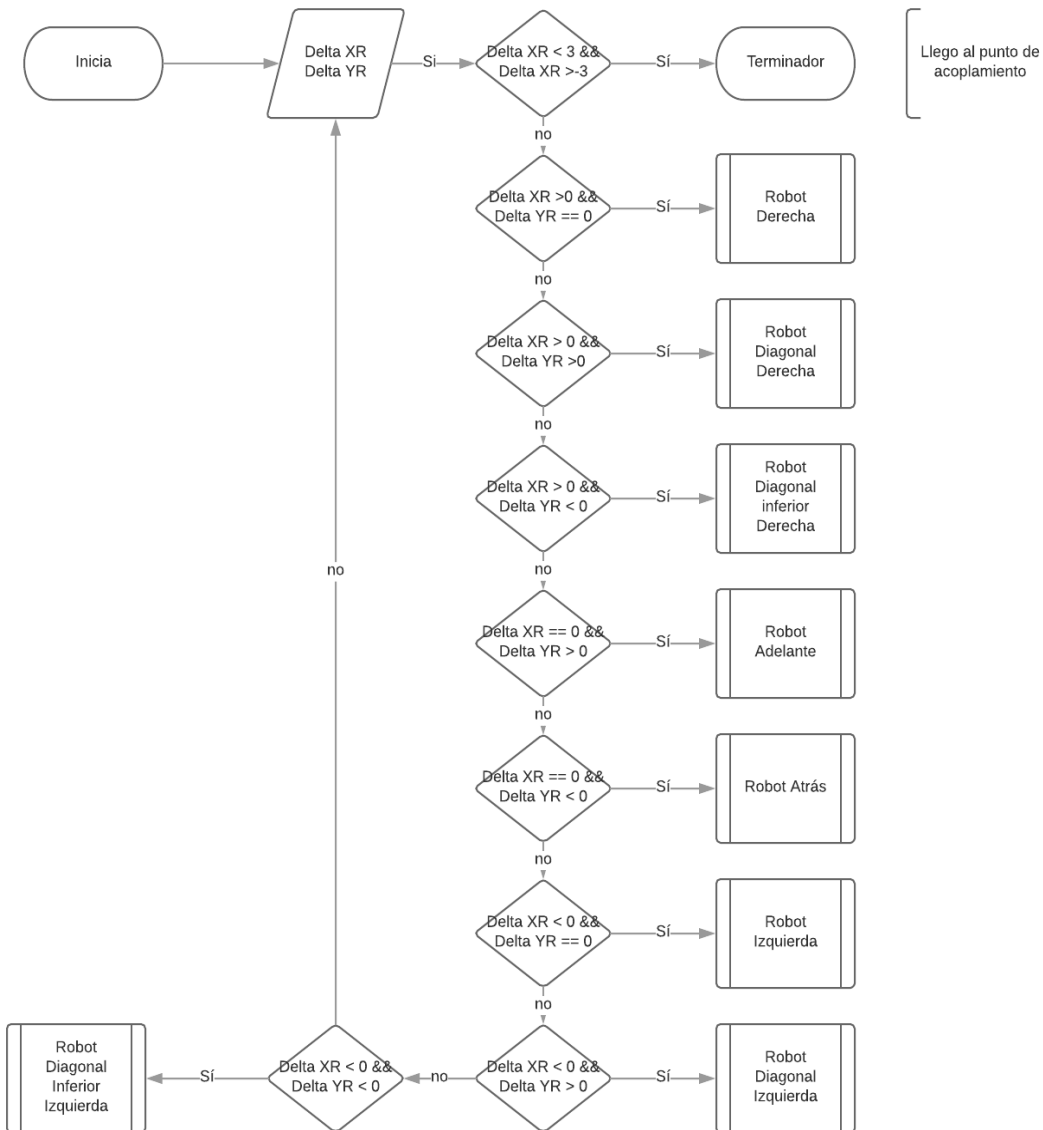


Ilustración 48 Diagrama de flujo de la trayectoria del robot

3.4.7 Subprograma de Orientación de la plataforma

Requerimientos para el control de la orientación de la plataforma durante su desplazamiento:

- Debe poder utilizar los diferentes tipos movimientos de cada robot, teniendo en cuenta que son omnidireccionales.
- Mantener la orientación durante su desplazamiento al punto de entrega, orientación de un ángulo de $\pi/2$.

- Debe moverse de forma colaborativa.
- La plataforma determina su orientación por medio de la visión de maquina suministrada por el agente Supervisor (software).

Para determinar la orientación de la plataforma se tomaron las coordenadas de los centroides de los dos robots (Ilustración 49), tomando como base el robot inferior, que durante el desarrollo es el Robot Móvil Rojo y el superior sería el Robot Móvil Azul, tomando los centroides se determina el ángulo real α que y el ángulo ideal β , que en la ilustración serian iguales, teniendo estos dos ángulos se emplea de nuevo la función MDelta ecuación (5), que haría el llamado a la función corrección cuando la diferencia de los ángulos sea mayor a 0.07 rad.

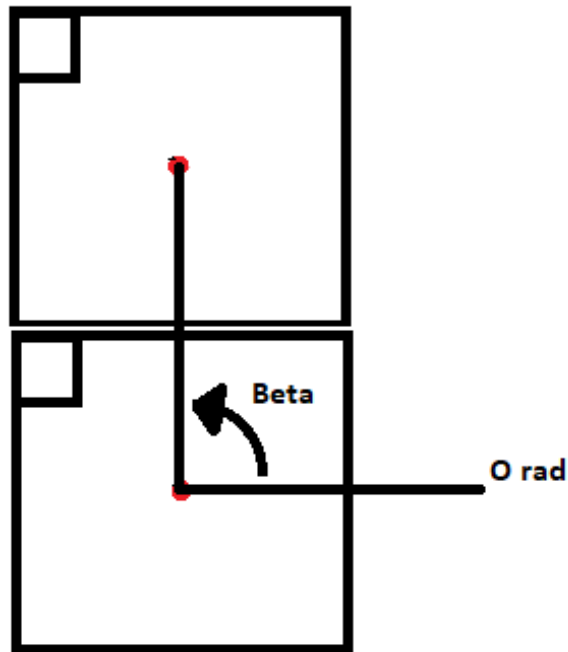


Ilustración 49 Robos acoplados con su orientación

La función que corrige la orientación está determinada en el diagrama de flujo de la Ilustración 50, en la cual se evalúan el ángulo real α y el ángulo ideal β enviando al robot la acción que debe realizar para la corrección de su orientación.

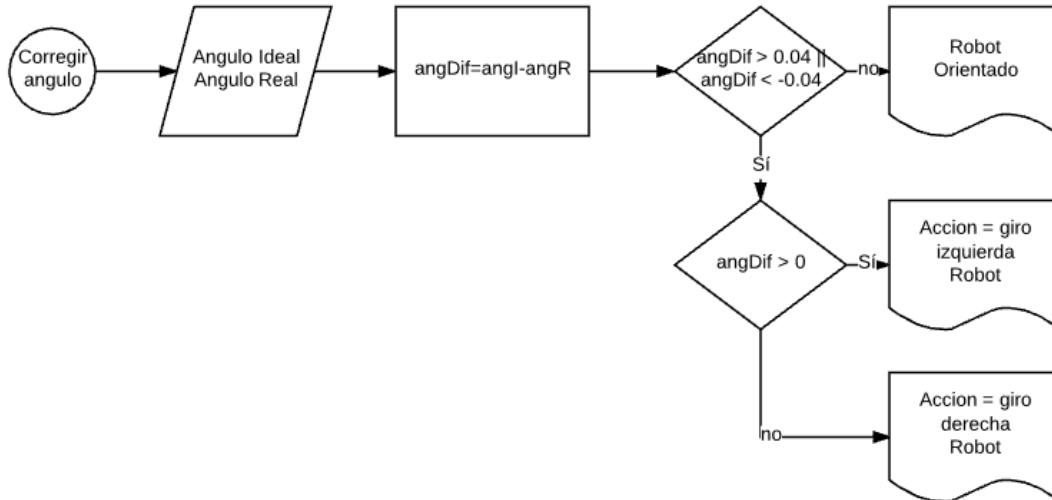


Ilustración 50 Diagrama de flujo de la función corregir ángulo

3.4.8 Subprograma de la trayectoria de la plataforma al punto de entrega

La trayectoria está determinada por la distancia entre dos puntos: el centroide del Robot Móvil Rojo $PP(x2,y2)$ y el punto de entrega suministrado por el usuario $PEnt(PuntoFX, PuntoFY)$. Las distancias obtenidas empleando las ecuaciones (11), ΔX y ΔY , en consecuencia, el tipo de acción que deben realizar los robots en el modo de plataforma para ir reduciendo la distancia al punto de entrega.

$$\begin{aligned} \Delta X &= PuntoFX - x2 \\ \Delta Y &= PuntoFY - y2 \end{aligned} \quad (11)$$

En la Ilustración 51 se ve gráficamente los vectores que componen la distancia del robot plataforma al punto de entrega, teniendo como origen el centroide del Robot Móvil Rojo.

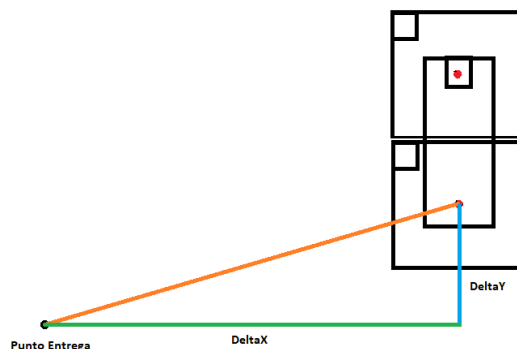


Ilustración 51 Trayectoria de la plataforma hasta el punto de entrega

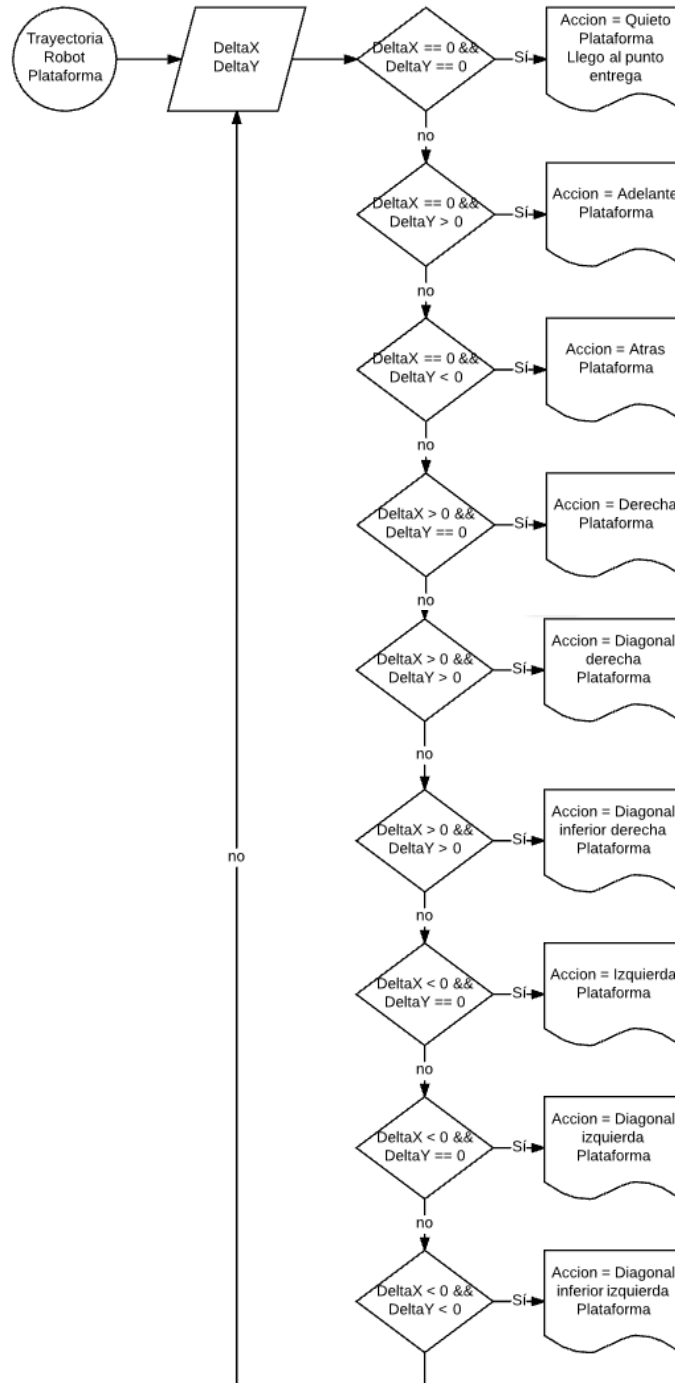


Ilustración 52 Diagrama de flujo de la función trayectoria al punto de entrega

El diagrama de flujo (Ilustración 52) evalúa los valores de las componentes del vector distancia de la plataforma y el punto de entrega. Las diferentes combinaciones de magnitudes determinan que acción deben tomar los dos robots, teniendo en cuenta que el movimiento de los motores de la plataforma no es igual al movimiento del robot individual, pero si mantiene el movimiento sin restricciones siendo un robot omnidireccional.

CAPITULO IV

Simulación

4.1 Software de Simulación de Robótica Webots®

El software empleado en la simulación de los robots y su entorno es Webots®, diseñado especialmente para la robótica, para el desarrollo de entornos virtuales, modelamiento de robot y el diseño de controladores, permitiéndonos la simulación integral en robótica para así buscar un comportamiento muy cercano al real, la interacción robot, entorno y su comportamiento.

Este software permite importar modelos de elementos mecánicos, en este proyecto se importará el modelo del Robot Móvil heterogéneo diseñado en SolidWorks®, este programa fue dado por la universidad para el desarrollo de esta investigación. En la Ilustración 53 se ve la vista del programa, sus componentes para el desarrollo de la simulación.

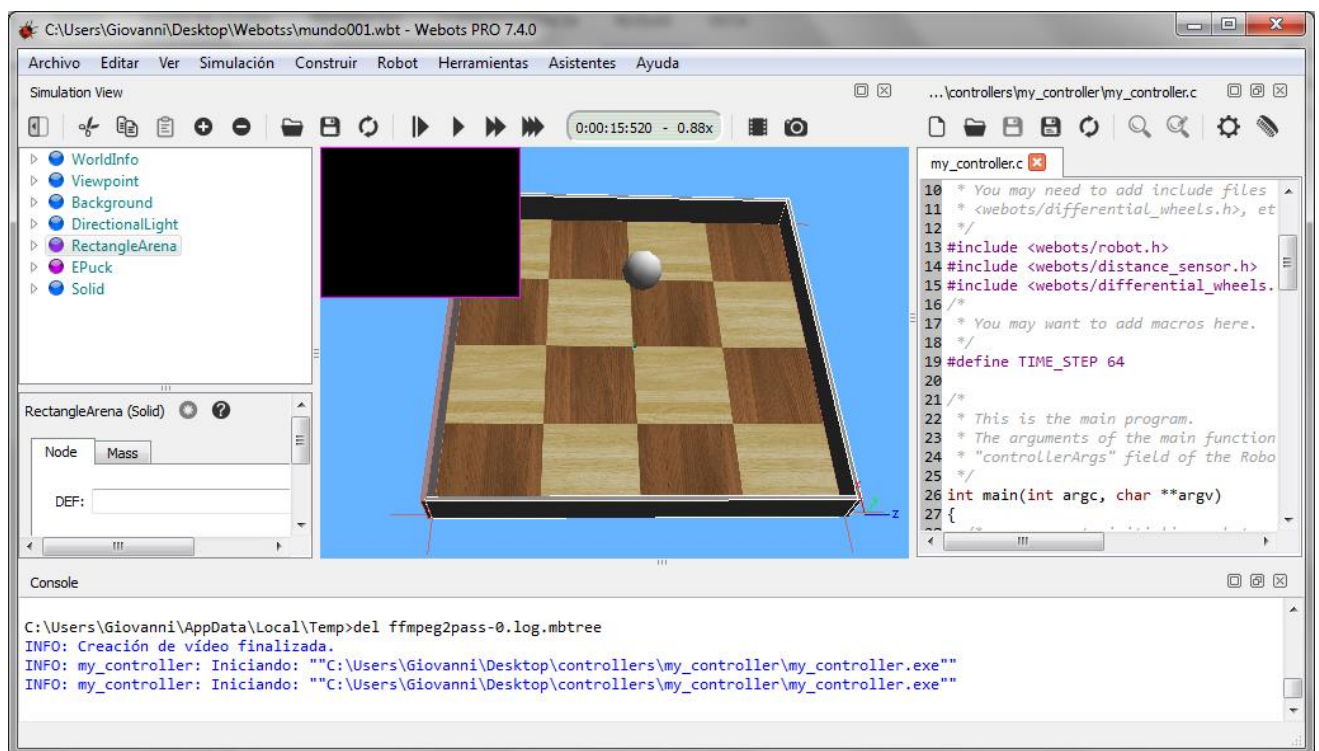


Ilustración 53 Interfaz del programa.

4.2 Tutoriales desarrollados Webots®

a. Creación de un mundo nuevo

En la Ilustración 54 muestra la ventana de la creación de un mundo nuevo que está compuesto por inicialmente por un área de trabajo llamado RectangleArena, que por defecto está constituida por un piso de unas medidas iniciales de 1m por 1m.

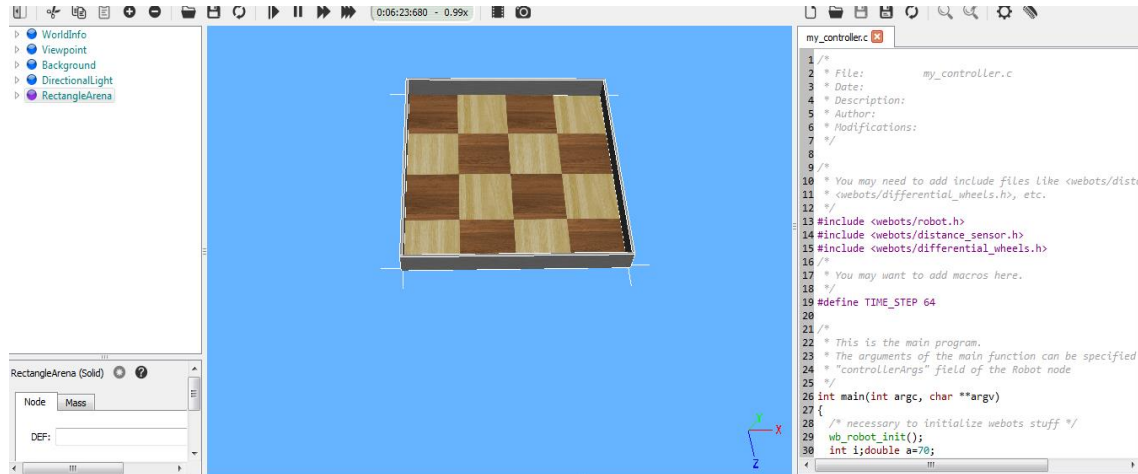


Ilustración 54 Creación del mundo nuevo.

b. Agregar un robot e-puck al mundo creado

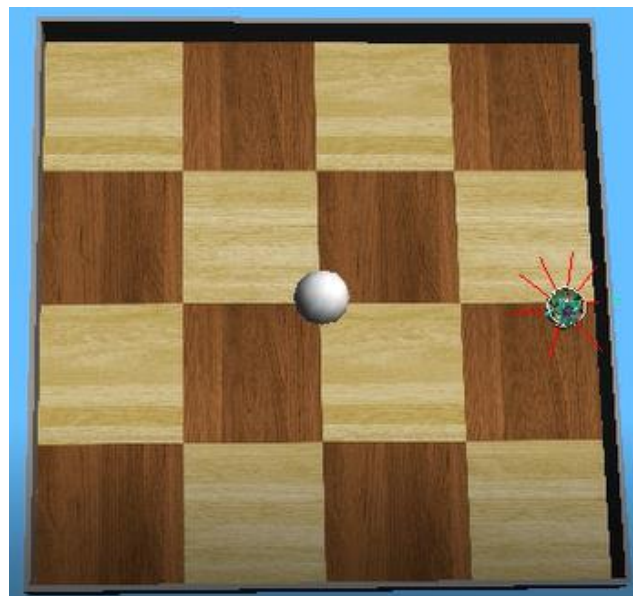


Ilustración 55 Programa del e-puck.

En el segundo tutorial se implementó el mundo, un robot llamado e-puck, el programa por defecto tiene varios robots, en su parte física y sus controladores básicos para que sean empleados de manera rápida para el desarrollo de simulaciones visto en la Ilustración 56

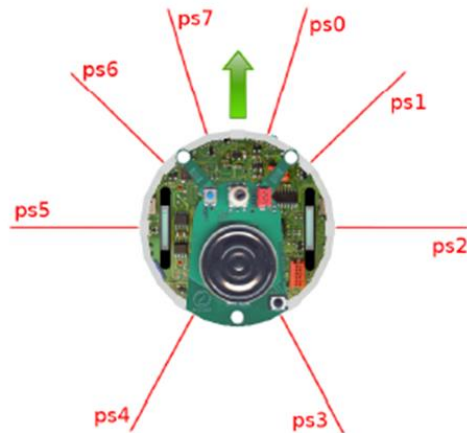


Ilustración 56 Diagrama de sensores.

El robot e-puck fue programado el controlador para que siguiera el contorno del mundo formado por 4 paredes, el robot cuenta con 8 sensores infrarrojos en su contorno, que fueron empleados para detectar las paredes evitando chocar con ellas y mantener una distancia mínima, es un robot diferencial por lo cual se controlaron los dos motores para poder mover el robot, lo cual se ve en la Ilustración 56.

c. Creación de objetos simples para el ambiente

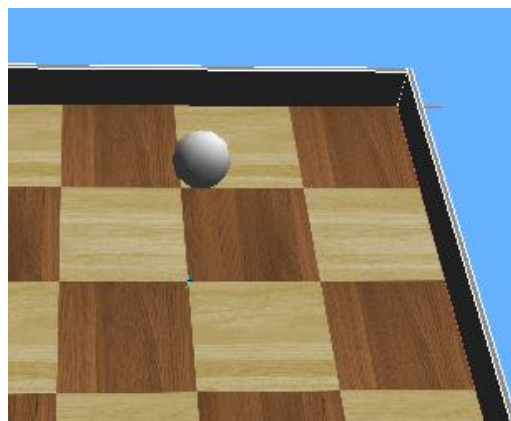


Ilustración 57 Creación de una esfera en el mundo.

El programa cuenta con una organización en forma de árbol, en él se pueden ingresar nodos, que a su vez representan cada uno de los elementos que podemos ingresar a la

simulación, como lo son los robots, elementos solidos objetos con los cuales el robot podrá interactuar y desarrollar el programa, así mismo elementos que harán parte del mundo en el cual se desarrollará la simulación como el terreno, montañas y más, en la Ilustracion 57 se ve el elemento que es una esfera la cual tiene componentes físicos como lo son la gravedad, el peso, las dimensiones reales y la interacción con los demás elementos de la simulación, la simulación permite ver también los puntos de contacto del elemento.

d. Desarrollo de dos robots homogéneos de 4 ruedas

Se creó un mundo en el cual, se adicionaron dos robots empleando los módulos de elementos de disponibles en el software (Ilustracion 58), teniendo gran cantidad de elementos como sensores de diferentes usos, así como elementos que enriquezcan la simulación de acuerdo con las necesidades.

Sensores empleados en la simulación:

1. Sensores de proximidad infrarrojos 2, empleados para detectar objetos presentes durante su desplazamiento.
2. Sensores de infrarrojos para detección de línea 4, la forma en la cual los robots pueden identificar el mundo por medio de los sensores, la forma en la cual el robot puede navegar en el mundo es por medio de una línea negra, la cual es detectada por los sensores, los cuales detectan la ubicación del robot con línea, el controlador determina las diferentes opciones en la lectura de los sensores, identificando la ubicación del robot si está a la derecha o izquierda de la línea, si llego al final de una, si esta fuera de la línea y generar el movimiento necesario para encontrarla.
3. Objetos de identificación que están divididos en dos: los objetos de interés cilindros para su recolección y los obstáculos cubos, que serán removidos del camino para continuar con el movimiento.

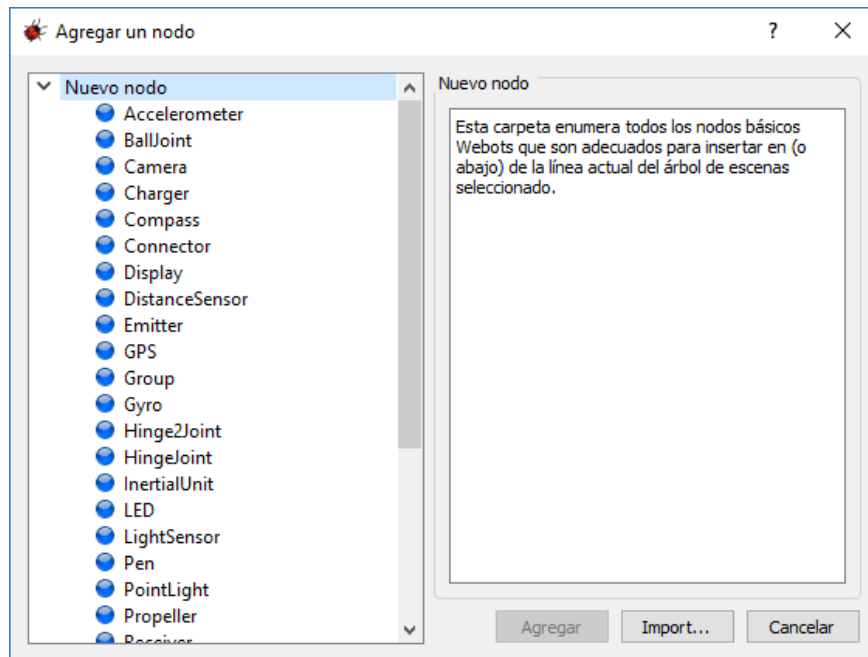


Ilustración 58 Opciones de nodo del programa.

4. Sensores de transmisión y recepción de información, los cuales permiten la comunicación entre los dos robots, que serían dos agentes físicos y un tercer agente que sería un software que sería el encargado de generar el mapa de ubicación de los obstáculos y de los objetivos, información transmitida por los dos robots.
5. Los actuadores son 4 motores que son programados con relación, a la percepción de los sensores para orientar el robot y mantenerlo en la línea, así mismo en caso de perder la línea volver a retomarla.
6. Un cuerpo en el cual es la base del robot, en donde se ubica los señores que tiene una parte de programación y una parte física que puede ser manipulada para que sean identificados, como lo es el color.

En la Ilustración 58 se identifica el árbol de nodos del robot Reco1, en el cual se ven los sensores que lo componen y sus partes físicas, cada uno de los nodos tiene también sus características como lo son: su ubicación, rotación, escala, nombre el cual es muy importante porque determina de qué forma puede ser llamado en el controlador, sus características físicas, como el color, peso, identificación del tipo de sensor etc., se identifica el nombre del controlador, lo cual permite que desarrollemos varios controladores, con relación a las necesidades de la tarea a desarrollar.

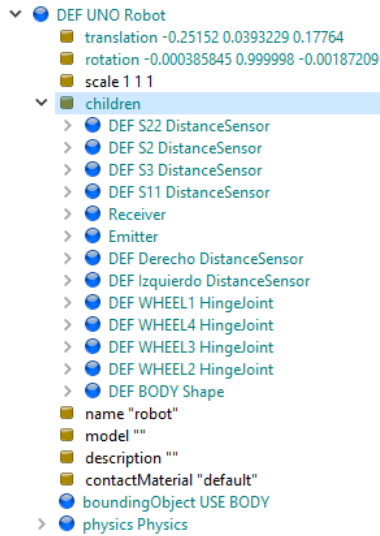


Ilustración 59 Árbol del programa principal.

En la Ilustración 60 se puede ver todos los elementos presentes en la simulación que hacen parte de la programación, se ven los tres objetos a identificar, los de interés, un elemento obstáculo el cubo rojo que deberá ser retirado por los robots, el mapa que está representado por la línea negra, cada robot debe seguirlo para poder realizar la navegación del mundo, es un mapa que está compuesto por nodos que son representados por las entradas y salidas, que son orientadas a los puntos cardinales (Norte, sur, oriente y occidente).

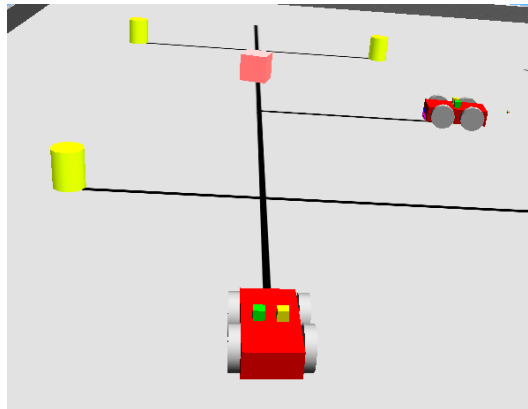


Ilustración 60 Mundo de dos robots heterogéneos.

El mapa es un tipo de laberinto en el cual los robots deben navegar por lo cual el controlador, fue diseñado para:

- a. Generar un árbol de nodos, donde cada nodo del mapa genera un nodo en el programa, que es presentado por un vector el cual guarda la cantidad de salidas del nodo, identificando cada una de ellas.
- b. En cada nodo identifica el ingreso y que opciones tiene para salir del nodo, llevando el historial.

- c. Él debe identificar su ubicación con relación al norte, el algoritmo determina las opciones de salida y da prioridad a la izquierda, llenado el nodo para buscar en las diferentes entradas y salidas la búsqueda de los objetos y obstáculos.
- d. El controlador identifica los objetos con los sensores del robot, generando un mapa de la ubicación de los elementos, lo cual permitirá su ubicación rápida, optimizando la ruta para llegar a en el caso de recolección, ocurre lo mismo con los obstáculos, compartiendo la información con el otro robot, los dos son orientados para la navegación y la toma de decisiones abarcando de forma más rápida y optima y así poder encontrar todos los elementos del mundo.
- e. El programa principal identifico la ubicación de los obstáculos y objetos e identifico la ruta más rápida a ellos, el programa finaliza con la llegada a un punto final y el retorno por la ruta más rápida al inicio (Ilustracion 61).

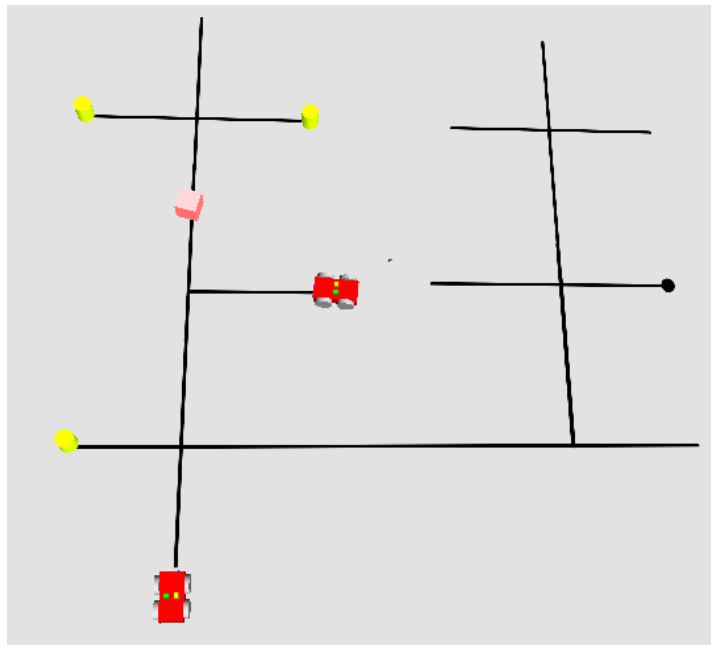


Ilustración 61 Mundo completo

e. Desarrollo de 2 robots heterogéneos

Se diseñaron dos robots de arquitecturas diferentes, empleando como modelo las piezas de los robots Bioloid®, creando dos robots una Arana de seis patas y una pinza y un robot móvil de 4 ruedas y una pinza, para recorrer el laberinto creado en la simulación anterior, orientado este programa para la recolección de los elementos del laberinto (Ilustracion 62).

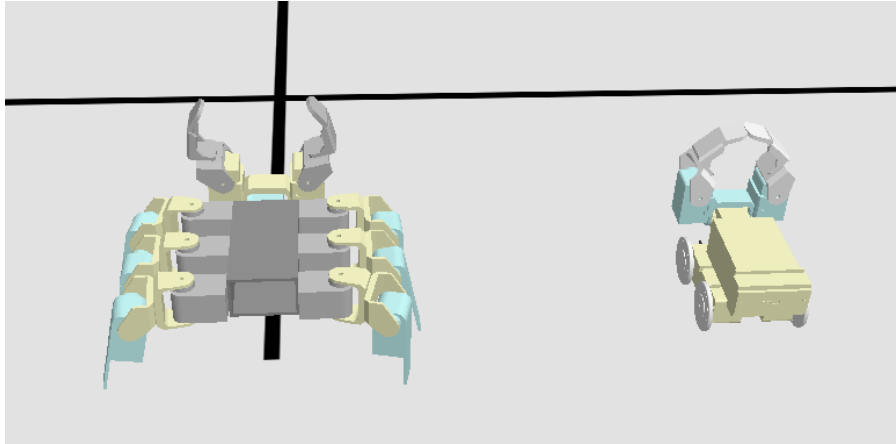


Ilustración 62 Robots heterogéneos

El control de cada uno de los robots es diferente, teniendo en cuenta su configuración, los robots fueron diseñados en SolidWorks®, teniendo con base las piezas de los robots Bioloid, kit que la universidad tiene disponibles para investigación y desarrollo para los alumnos.

➤ Exportación de CAD de SolidWorks® a Webots®

Importación de los Robots de SolidWorks® a Webots®, los modelos diseñados para ser exportados de SolidWorks® deben contar con las siguientes características:

- a. Los modelos deben ser simplificados, tratando de hacer los modelos con la menor cantidad de detalles teniendo en cuenta, que el programa Webots® disminuye su capacidad de procesamiento al aumentar los detalles de los elementos del robot por lo cual, el programa tiende a bloquearse, si se tiene en cuenta que lo importante de la simulación es que podamos desarrollar el controlador con relación a un mundo que ha sido creado.
- b. El modelo diseñado en SolidWorks® debe ser guardado en VRML, especificando la versión 97, teniendo en cuenta que debe ubicar un centro de coordenada del modelo a exportar en el centro del mismo, porque de este eje depende la correcta ubicación del modelo en Webots® y llenado las medidas en metros, en la Ilustración 63 se observa la ventana de guardar como en VRML.
- c. En Webots® se busca en archivo la opción importar VRML, llevando a si al programa el modelos que se necesita importar, el modelo no como uno solo en el árbol de nodos, sino aparece la cantidad de piezas como transforma, pero en la imagen de la simulación aparece todo el modelo, por lo cual debemos empezar generando un nodo robot, en el cual empezamos a adicionar cada pieza por pieza, pero debemos ir identificando, al varia la escala o el color y así colocarles el nombre e ir adicionándola al nodo robot que ha sido creado.

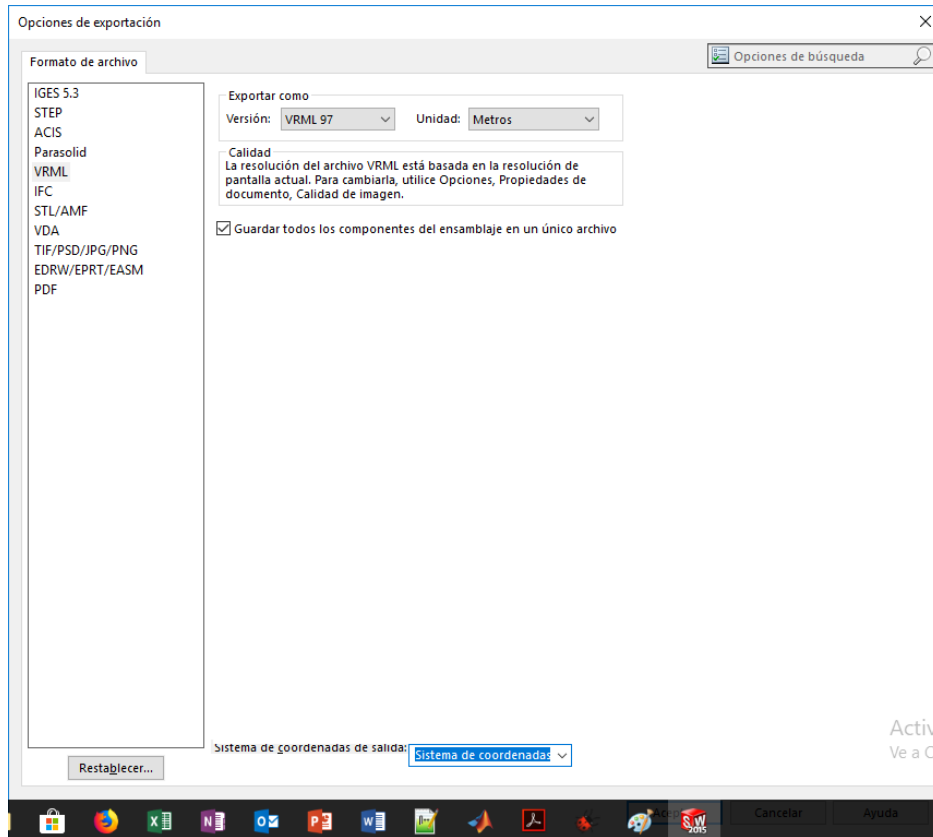


Ilustración 63 Ventana para guardar modelo de SolidWorks®.

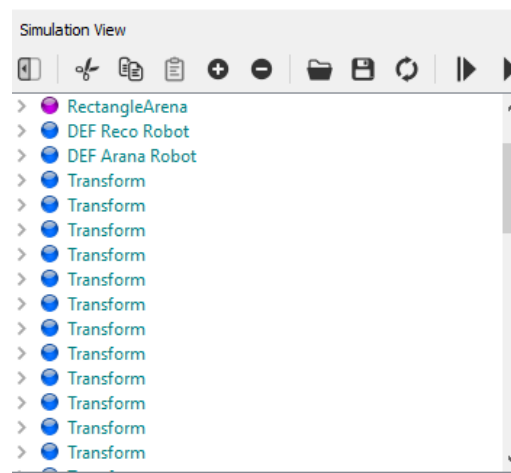


Ilustración 64 Nodo creados a partir de la importación del modelo a Webots®

El modelo al ser importado a Webots® pasa con la cantidad de elemento que fue diseñada y en la misma ubicación, pero no determina el nombre, lo cual se debe ser identificado para poder unirlos al nodo robot Ilustracion 64.

4.3 Desarrollo de la simulación de trabajo colaborativo

Los prototipos diseñados en SolidWorks®, va a ser implementados en el programa de simulación de Webots® para verificar su comportamiento, su respuesta para así poder determinar el prototipo para ser elegido para su construcción.

4.3.1 Simplificación del modelo en SolidWorks

Para la importación de los modelos CAD, se debe tener en cuenta que el programa de simulación, al importarle un modelo con todos los detalles y partes hace que el coste computacional aumente y va directamente relacionado con la capacidad del computador en el cual realiza la simulación, razón por la cual el modelo debe ser simplificado para poder optimizar el rendimiento del programa y poder aumentar la complejidad de la simulación.

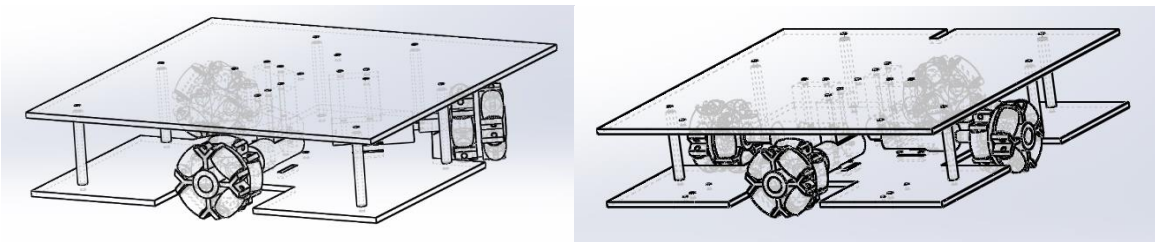


Ilustración 65 Modelo robot móvil 002 y 004

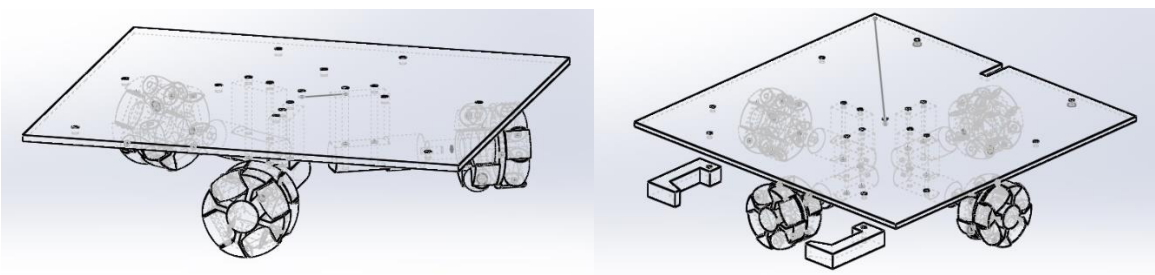


Ilustración 66 Modelo Robot móvil 002 y 004 simplificado

En la Ilustración Ilustración 65, permite ver los prototipos que vas a ser modificados, las modificaciones realizadas son:

- Se modificaron la cantidad de elementos que componente tratando de reducirlo a la menor cantidad, quitando la placa inferior y los tonillos de unión de las dos placas, pero sin cambiar la ubicación de las ruedas y su orientación.
- Se simplificaron las ruedas omnidireccionales se redujo de 17 partes a 9 partes, para reducir el coste computacional, a mayor cantidad de partes mayor cantidad de cálculos para el programa (Ilustración 66).
- Se redujeron los detalles de las partes con orificios, partes rectas, cortes que no sean relevantes para la funcionalidad del modelo.

4.3.2 Nuevo sistema de coordenadas del CAD

Se debe crear un nuevo sistema de coordenadas en el CAD antes de convertirlo a VRML 97, porque el programa Webots® al momento de importar el robot, toma el eje de coordenadas el que viene por defecto en el ensamblaje y tomara como eje la ubicación del robot durante la simulación, razón por la cual lo ideal es que el robot tenga su eje de coordenadas en el centro del mismo, por lo cual se creó un nuevo eje de coordenada en el CAD (Ilustración 67), teniendo en cuenta las unidades a metros y en especial el nuevo sistema de salida.

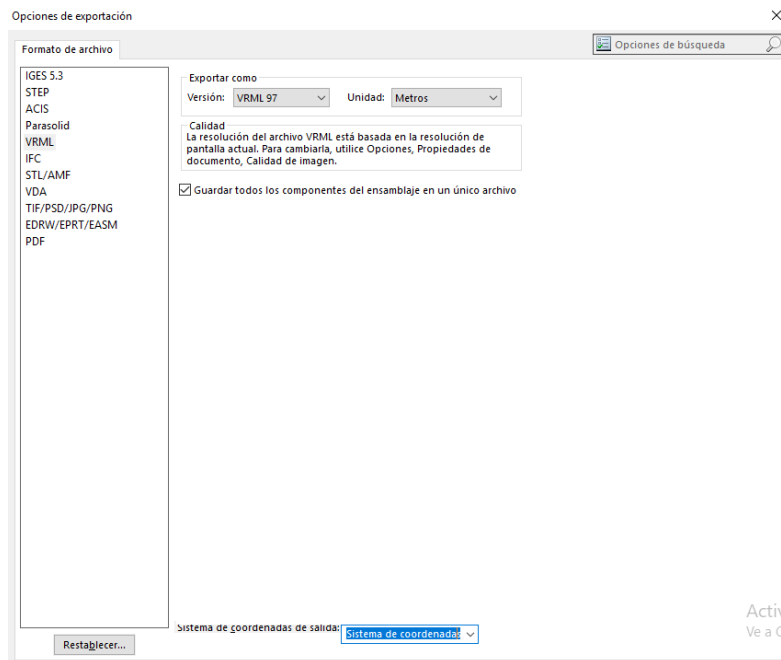


Ilustración 67 Ventana de guardar en formato VRML 97

El CAD del robot para Webots debe conservar las características esenciales para así, poder crear una simulación que nos permita evaluar correctamente los prototipos iniciales.

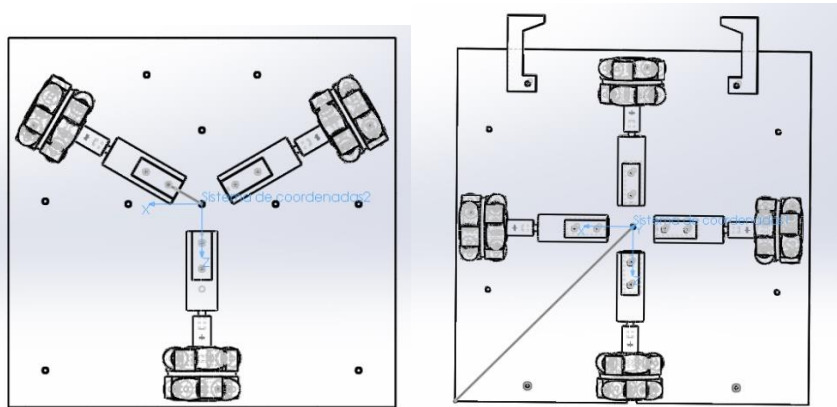


Ilustración 68 Prototipos con nuevo sistema de coordenadas

4.3.3 Importación del CAD a software de simulación

Para la importación del robot nos dirigimos a File y después escógenos la opción Import VRML 2.0 y escogemos el archivo del robot antes guardado, el ensamble importado no va a contar con las relaciones entre elementos, ni sus materiales o apariencias, pasan todas las partes y son vistas como transform Ilustracion 68.

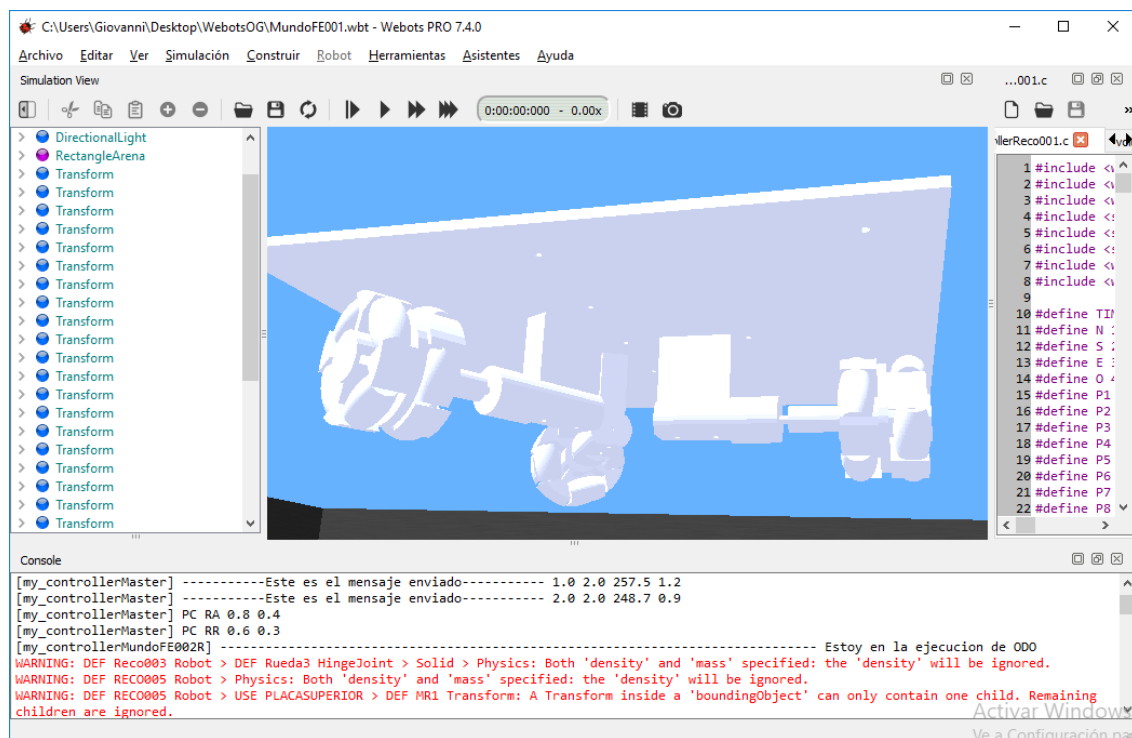


Ilustración 69 Importación del robot a Webots

El robot es importado al mundo, lo primero es identificar cada una de las piezas ya sea modificando su color o su escala e ir asignando el nombre correspondiente, se deben identificar todas las piezas del modelo importado, como se ve en la Ilustración 70 pero como la totalidad de partes debe ser asignado a un nodo robot e ir relacionando las piezas.

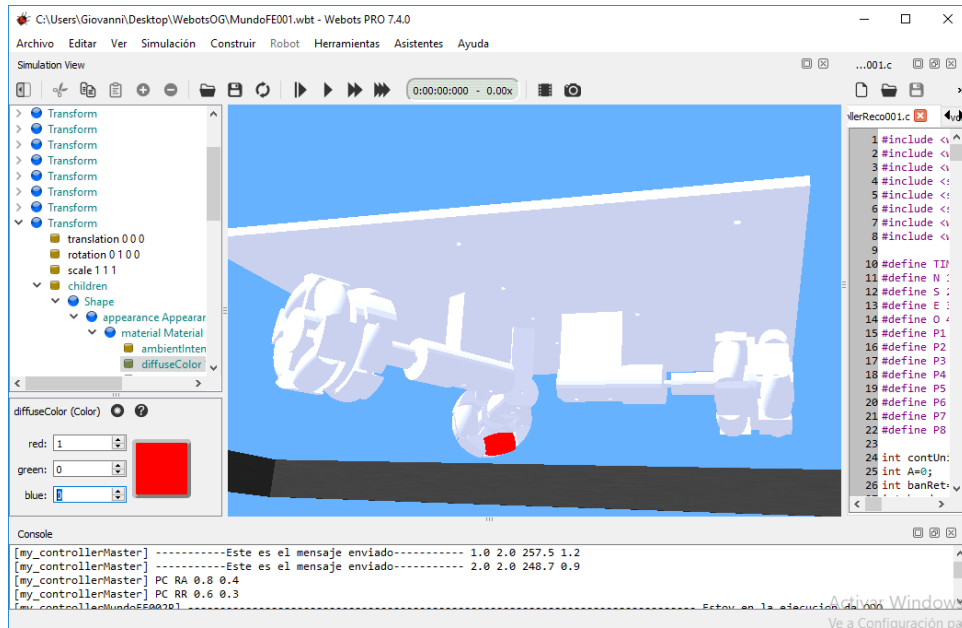


Ilustración 70 Identificación de las piezas por color

Al identificar la pieza por color determinamos que elemento es y determinamos la característica de la pieza si es un sólido, motor o sensor (Ilustración 71).

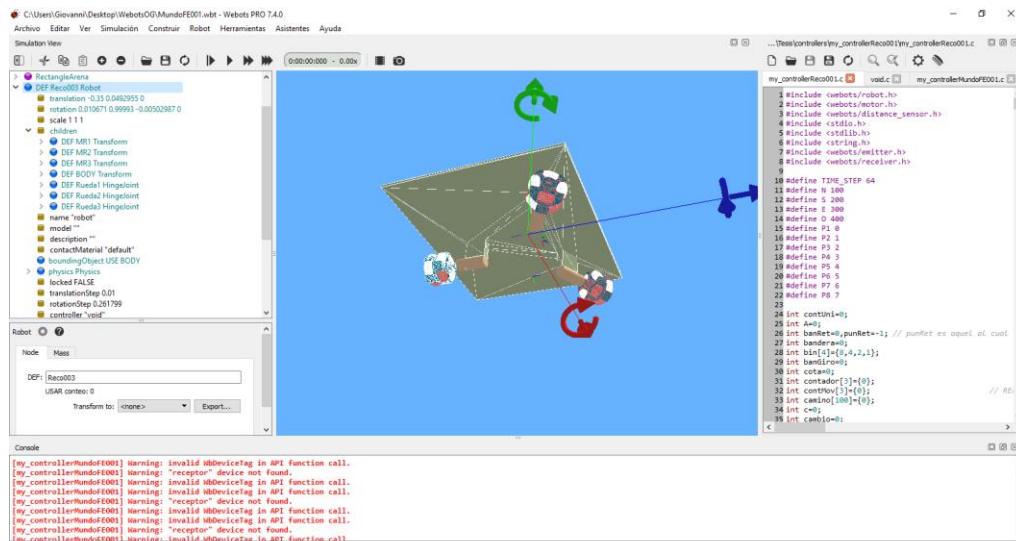


Ilustración 71 Robot identificado y construido

Creamos un nodo robot en el cual creamos un nodo children en el cual dependiendo de la parte la implementamos como parte de la estructura o como Hingejoint en la cual

generalmente se utiliza para colocar las ruedas o la parte que realizara el movimiento (Ilustración 72).

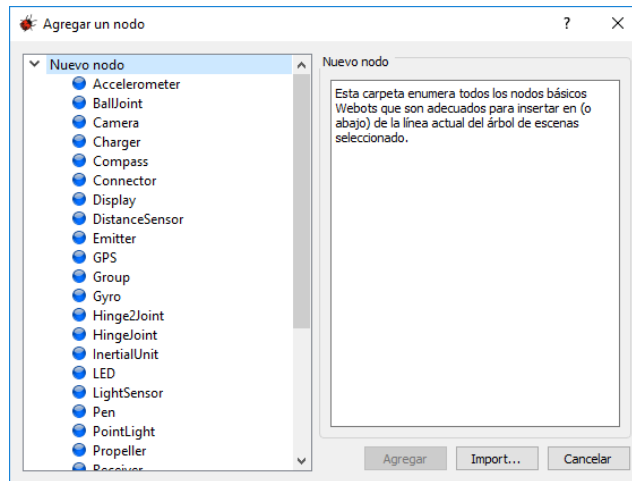


Ilustración 72 Opción de nodos para el robot

En la Ilustración 72 identificamos los diferentes nodos que pueden ser incorporados al nodo children del nodo robot, dentro del nodo robot es muy importante determinar cuál elemento se va a comportar como base para ser adicionado al nodo boundingObject, lo cual determina las líneas de delimitación de la pieza, mientras el children se tiene el sólido completo. Al tener definido este nodo se procede a adicionarles las características físicas del elemento al adicional al nodo physics el nodo Physics (Ilustración 73).

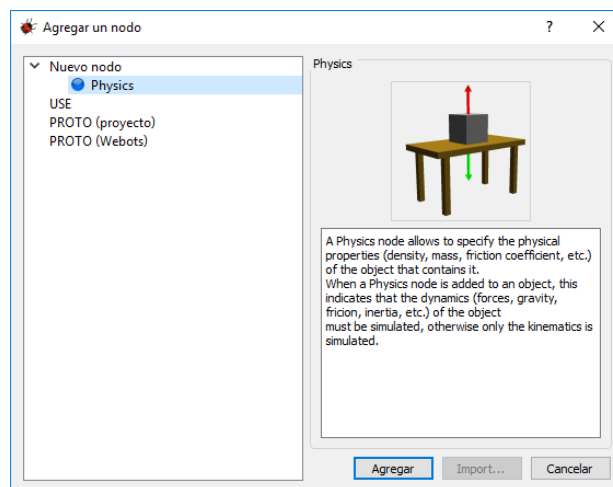


Ilustración 73 Nodo Physics

El nodo physics nos permite agregar al modelo el peso, la densidad, centro de masa, importante para que el robot, pueda interactuar con el mundo creado en su parte física, existiendo contacto y fricción entre los diferentes elementos del mundo. A partir de ahora podemos agregar los diferentes nodos que sean necesarios para que comportamiento del modelo sea lo más cercano a la realidad en la simulación Ilustración 74.

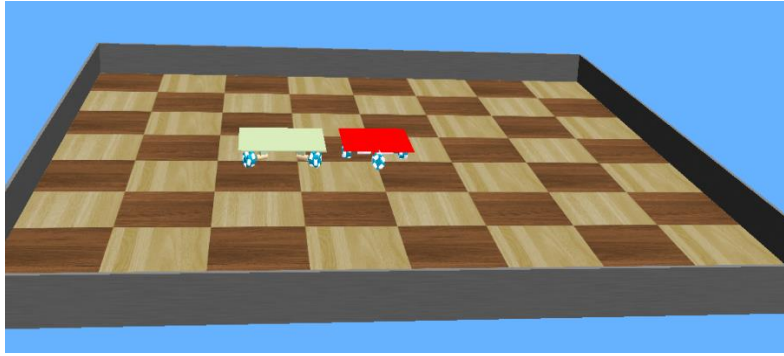


Ilustración 74 Robots importados a Webots

➤ Selección del robot base del proyecto

Tabla 8 Representa la comparación de las características evaluadas en el software de simulación Webots® de los dos prototipos.

Características	Robot móvil 002	Robot móvil 004
<i>Tipos de movimiento</i>	Tiene todos los tipos posibles de movimiento.	Tiene todos los tipos posibles de movimiento.
<i>Movimiento en diagonal</i>	Presenta menor estabilidad al moverse en diagonal en la simulación.	Presenta mayor estabilidad en el movimiento en diagonal.
<i>Acoplamiento</i>	No permite acoplamiento al tener dos llantas en el mismo lado lo cual dificultaría el mecanismo de anclaje.	Permitiría el mecanismo de anclaje a coincidir las ruedas en la mitad de cada lado.
<i>Cantidad de piezas</i>	Se reduce la cantidad de piezas en una rueda.	Tiene la mayor cantidad de piezas de los dos robots.
<i>Estructura</i>	La disposición de las tres ruedas a 120 grados hace que el control de dirección sea más complejo en relación que necesita el movimiento de las tres ruedas en toda clase de movimiento.	la disposición de las 4 ruedas permite controlar fácilmente los diferentes tipos de movimiento, así mismo hay motores que no funcionarían en algunos movimientos y esto genera mayor estabilidad, el ángulo entre llantas es de 90 grado.
<i>Valor</i>	Tendría un valor inferior al robot móvil 004	Tendría el valor más alto.

Los dos robots fueron importados al software de simulación (Ilustración 74 Robots importados a Webots), para la realización de pruebas y determinar su comportamiento en el desarrollo de la tarea colaborativa en sus tres etapas. La Tabla 8 presenta la comparación de los dos robots en el software de simulación. El robot que se desempeñó de mejor

manera en el desarrollo de la simulación fue el Robot Móvil 004, su estructura de cuatro ruedas permite que su desplazamiento ya acoplado sea más estable y manipulable en comparación con la estructura de 3 ruedas.

4.3.4 Ensamble de los robots en la simulación

Ya definido el prototipo se importan a la simulación y se comienza a desarrollar programa, la simulación contará con una cantidad de elementos:

1. Agentes físicos: estará conformado por los dos robots homogéneos, que son desarrollados a partir del prototipo seleccionado que se llamarán RECOAZUL005 y RECOROJO007, que serán modificados para su adaptación a la simulación (Ilustración 75 Escenario de la simulación).

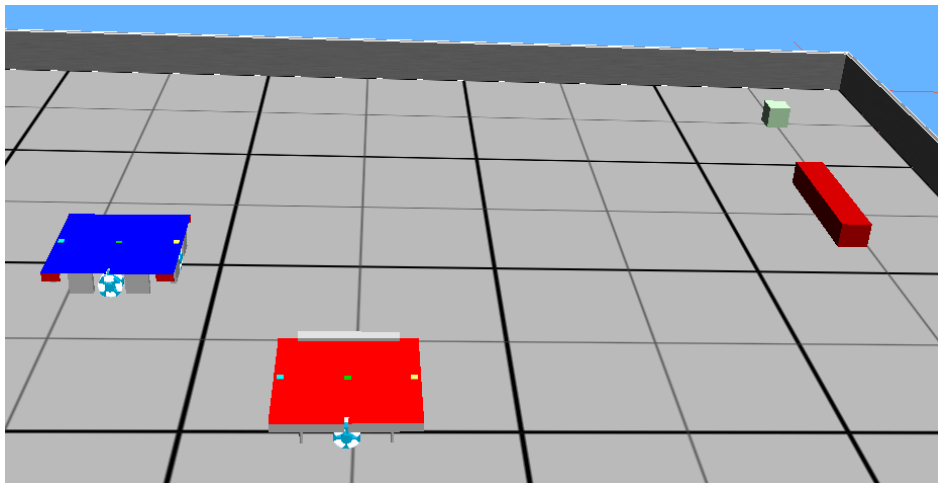


Ilustración 75 Escenario de la simulación

2. Agente supervisor: es el programa que por defecto cuenta el simulador que permite controlar, comunicarse e interactuar con los robots físicos de la simulación, en la Ilustración 76 están los diferentes componentes de la simulación.
3. Objeto: es el elemento el cual los dos robots tendrán que transportar, la ubicación la obtendrá el Agente Supervisor.
4. Punto de inicio y punto final: son los puntos en los cuales los dos robots inician su movimiento y en el cual ellos finalizan de forma colaborativa en su movimiento y ejecución.

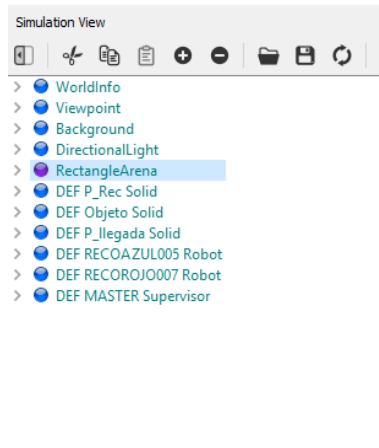


Ilustración 76 Árbol principal de la simulación

La simulación está dividida en tres fases que determinan el programa que controlaran a cada robot y al supervisor, el algoritmo está dividido en una capa reactiva y una capa deliberativa respectivamente para cada robot, los robots Azul y robot Rojo utilizaran estrategias de control reactivo y deliberativo y el supervisor empleara una estrategia de control deliberativo.

Configuración de los robots y del supervisor está organizada en su árbol de nodos respectivamente, en los cuales aparece cada una de las partes que los componen, partes sólidas, sensores y actuadores, Ilustración 77 robots (a) y supervisor (b).

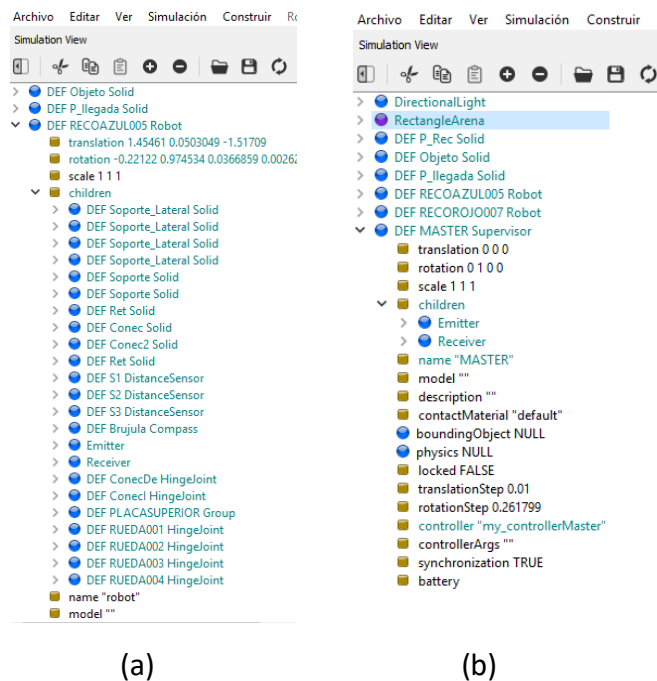


Ilustración 77 Árbol de nodos de los robots y el supervisor

Tabla 9 Nodos que componen a los robots y al supervisor.

Agentes Robots	Agente Supervisor
<i>Sensor Emisor y Receptor: la comunicación entre los agentes.</i>	<i>Sensores Emisor y Recepto: comunicación con los robots.</i>
<i>Sensores de distancia 5: empleados para el acoplamiento y la detección del objetivo.</i>	
<i>Brújula: le permite determinar su orientación con respecto al norte magnético.</i>	
<i>Motorreductores 4: son los encargados de mover las ruedas omnidireccionales.</i>	
<i>Pinzas de acoplamiento 2: son las que acoplan los robots para ser una sola plataforma.</i>	

La Tabla 9 representa los componentes de los agentes del sistema multiagente diseñado compuesto por una Agente Supervisor y dos Agentes Robots.

- Grafica de la brújula de los robots

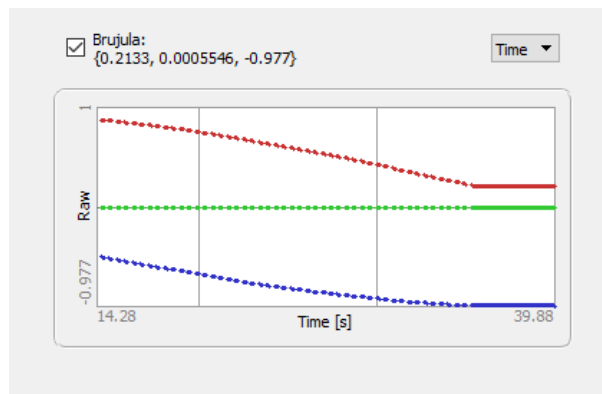


Ilustración 78 Grafica de la brújula de los robots

La grafica de la brújula, permite observar como grafica el cambio de dirección al orientar el robot hacia el ángulo ideal para llegar al punto de acoplamiento (Ilustracion 78).

- Grafica de los motores

El robot está en movimiento hacia adelante con la azimut o ángulo que apunta hacia el punto de acoplamiento, se observa cómo solo están activos dos motores 2 y 4 que van en sentido contrario, por lo cual su desplazamiento es hacia adelante y los motores 1 y 3 están detenidos, teniendo un comportamiento ideal del robot durante el programa de la simulación (Ilustracion 79).

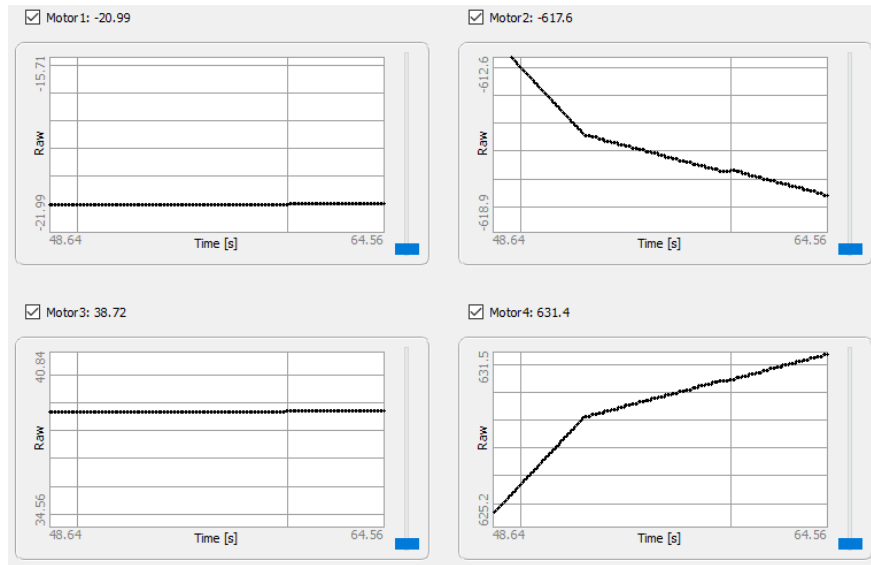


Ilustración 79 Grafica de los motorreductores

➤ Grafica de los servos

Los servos van en posición, para el acoplamiento por lo cual podemos ver en la gráfica que su posición es contraria y está de acuerdo con la posición indicada en la programación (Ilustración 80).

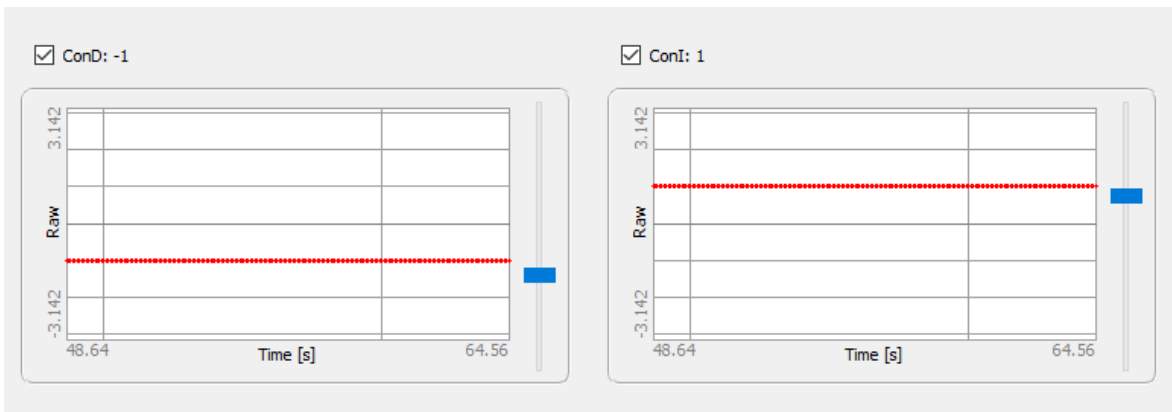


Ilustración 80 Grafica de los servomotores de las pinzas

4.3.5 Capas de desarrollo del software

La programación de cada uno de los agentes del sistema está dividida en 3 fases diferentes que dividen la tarea colaborativa de detección, recolección, acoplamiento y traslado del objeto a un punto de entrega (Ilustración 81).

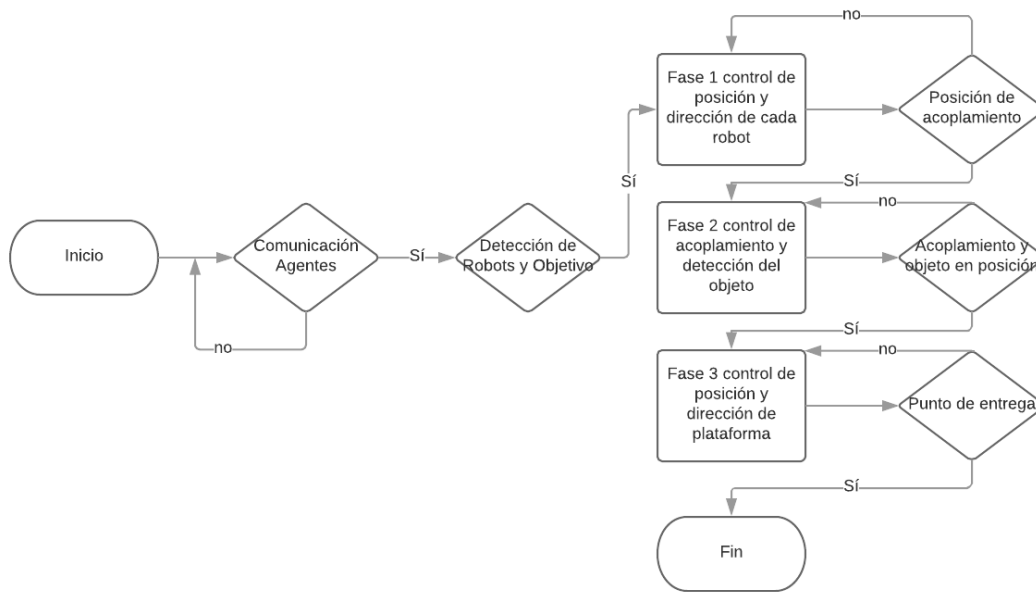


Ilustración 81 Diagrama de flujo de las fases de simulación

4.3.5.1 Fase 1

Los robots realizarán un desplazamiento inicial hacia unos puntos de acoplamiento, en los cuales harán alto y esperarán que órdenes del supervisor que le confirme que están en posición de acoplamiento Ilustración 82.

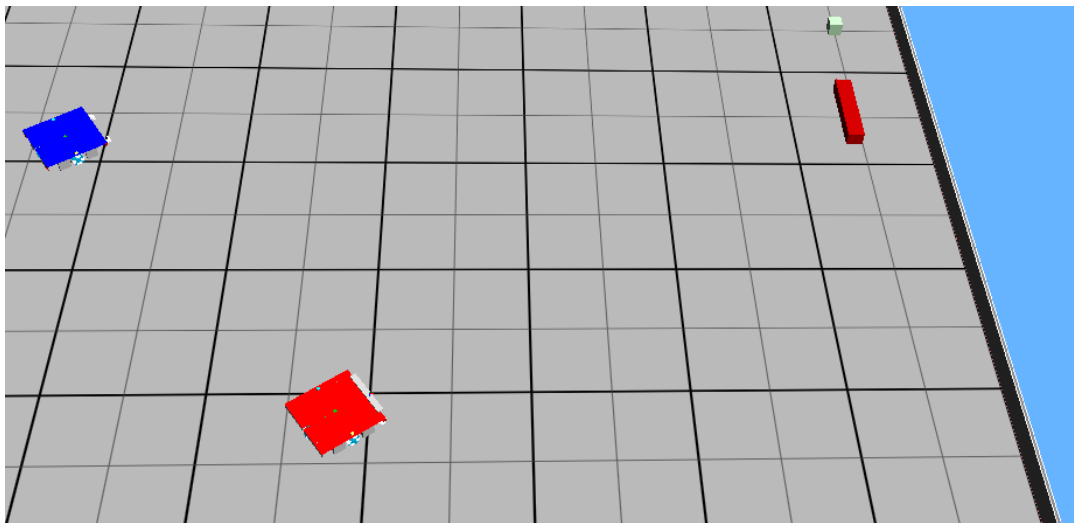


Ilustración 82 Robots en fase 1

4.3.5.1.1 Capa reactiva de los robots

1. Inicia la comunicación con el supervisor, informado su activación y recibe la orden de que tome el azimut o el ángulo hacia los puntos de acoplamiento en las inmediaciones del objetivo.
2. Recibe los datos de su brújula, comprara con el ángulo recibido y empieza a corregir e inicia con el movimiento.
3. Durante el desplazamiento recibe informes del supervisor para ir corrigiendo su dirección y verificando la distancia al objetivo.

4.3.5.1.2 Capa deliberativa del supervisor

1. Inicia la comunicación con los dos robots para confirmar su participación.
2. Detecta las posiciones de los robots, del objetivo determinando las posiciones de acoplamiento, determina la distancia y el ángulo necesario de cada uno de los robots.
3. Les envía la distancia y el ángulo para que ellos inicien con el movimiento, donde cada uno tiene una trayectoria determinada.
4. Durante el recorrido verifica las trayectorias tomada y les envía correcciones, porque su objetivo es orientarlos hasta las posiciones de acoplamiento.

4.3.5.2 Fase 2

Los robots al estar en la posición de acoplamiento inician con el programa, para que se conviertan en una sola plataforma y esperan que el objeto sea ubicado en su plataforma, para iniciar con el movimiento con el objeto a un punto de entrega, empleando el algoritmo colaborativo Ilustracion 83.

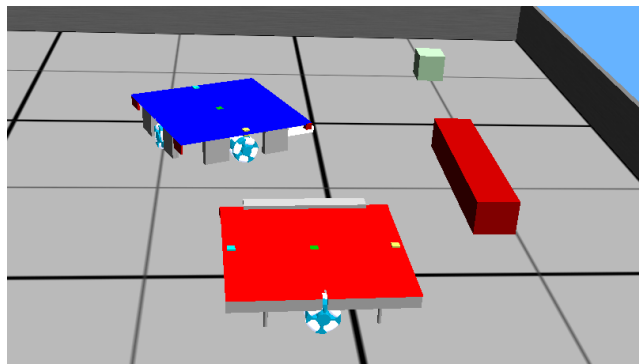


Ilustración 83 Llegada de los robots al punto de acoplamiento

4.3.5.2.1 Capa reactiva de los robots

1. Reciben la información que los dos están en posición e inician con el subprograma de acoplamiento, el robot a ser acoplado se queda quieto y el que acopla inicia movimiento hacia el otro robot (Ilustración 83).
2. Los sensores que tiene en las pinzas de acoplamiento le permiten corregir la dirección en caso de ir corrido con respecto al otro robot (Ilustración 84).

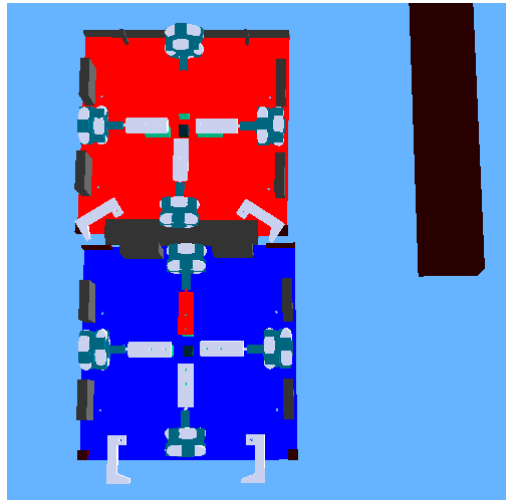


Ilustración 84 Momento del acoplamiento

3. Los sensores del Robot Rojo, determina la distancia ideal para el acoplamiento a si mismo las pinzas están diseñadas para que corrija cualquier desviación en el acoplamiento Ilustración 85(a).
4. Los sensores determinan el acoplamiento e informan al supervisor que están acoplados esperando el objeto Ilustración 85 (b).

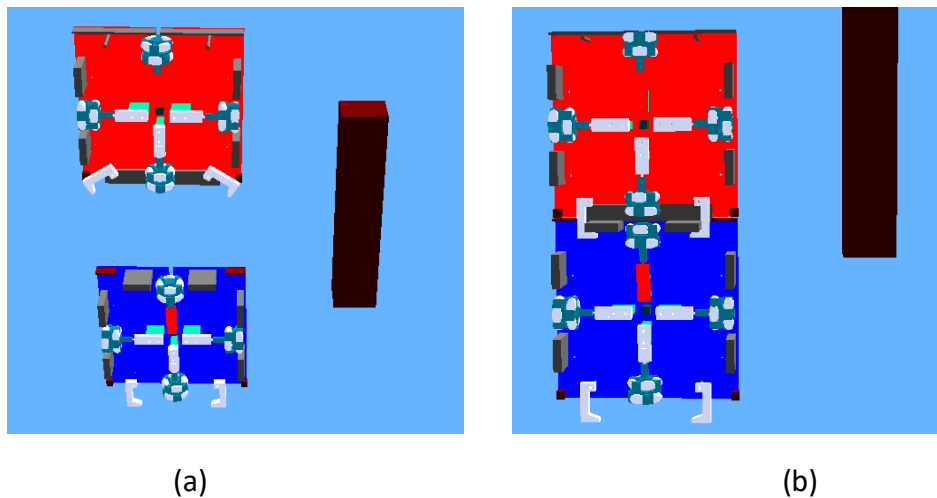


Ilustración 85 Robots en la fase de acoplamiento

4.3.5.2.2 Capa deliberativa del supervisor

1. El supervisor monitorea el movimiento de los robots y espera la confirmación del acoplamiento y así poder pasar al subprograma de carga del objeto y paso a la Fase 3 para iniciar el desplazamiento al punto de entrega (Ilustración 86).

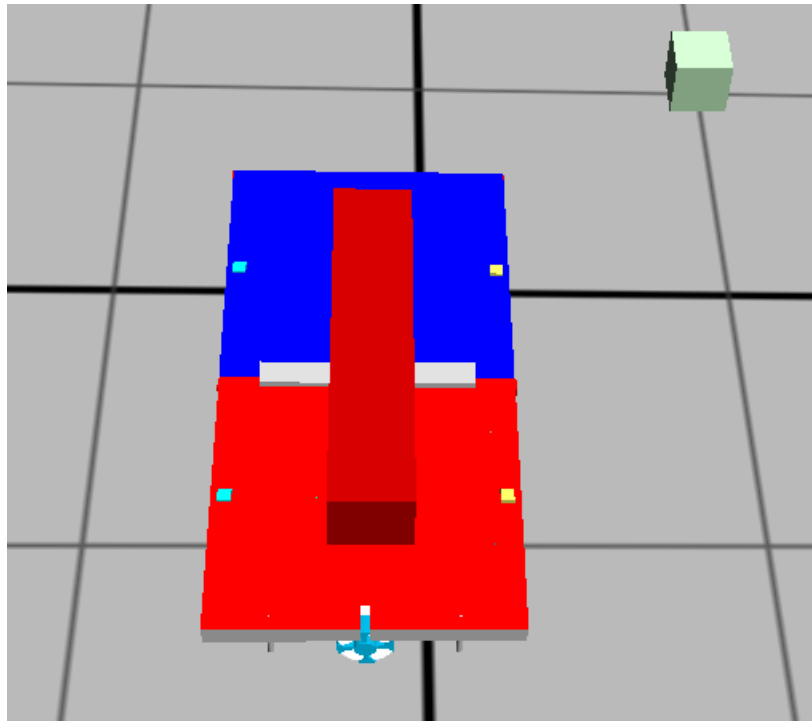


Ilustración 86 Robots acoplados y cargados con el objeto

4.3.5.3 Fase 3

Los robots inician la tarea de forma colaborativa y controlada hacia la posición de entrega en la cual finalizará la tarea colaborativa en el momento que llevaron el objeto al final.

4.3.5.3.1 Capa reactiva de los robots

1. Los robots en su movimiento como una sola plataforma se mueven conservando el norte empleando el modo de movimiento colaborativo, por la estructura de las llantas puede moverse de forma deseada.
2. Los robots van corrigiendo el ángulo durante el desplazamiento y recibiendo la información del supervisor, sobre su ubicación y como llegar al punto indicado.

4.3.5.3.2 Capa deliberativa del supervisor

1. Monitoreando el desplazamiento de la plataforma con el objeto y determinando puntos de control en los cuales verifica el ángulo y la distancia para seguir orientando a los robots al punto de entrega (Ilustración 87).

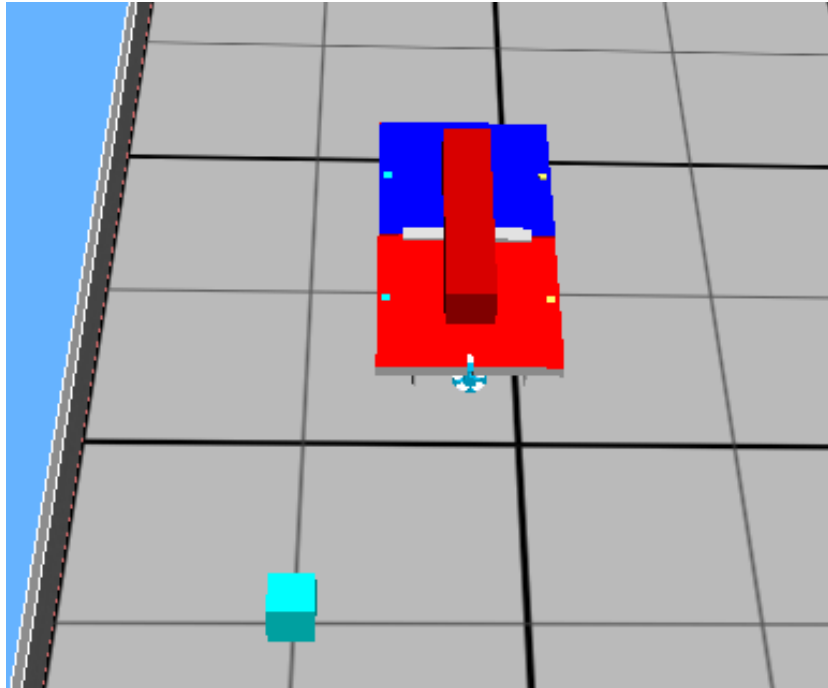


Ilustración 87 Llegada del robot acoplado a punto llegada

CAPITULO V

Implementación

5.1 Construcción del modelo estructural, mecánico y electrónico.

Modelo estructura, se mandaron a cortar los diferentes componentes de la estructura, las placas superior e inferior en acrílico de 3mm (Ilustración 88 e Ilustración 89), las piezas de los soportes de los motorreductores cortando un total de 20 piezas en su totalidad con un valor de corte y acrílico de \$ 60.000 pesos.

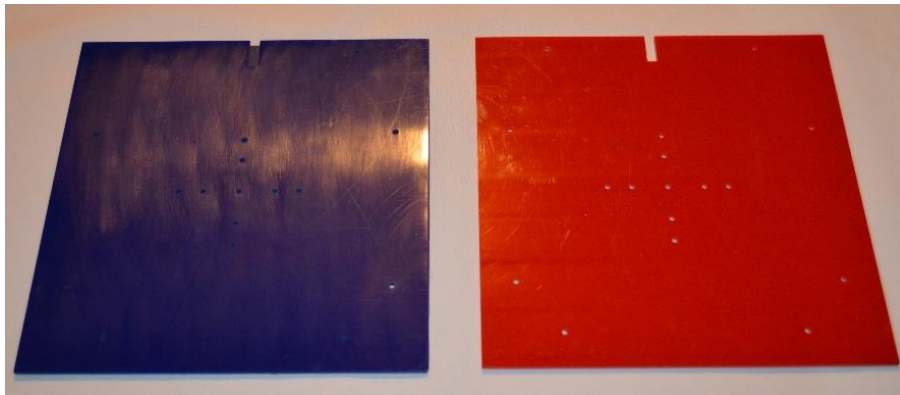


Ilustración 88 Placas superiores en acrílico

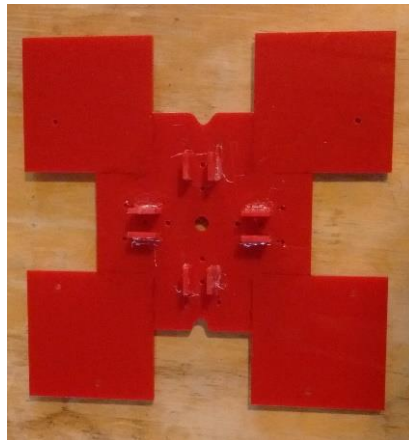


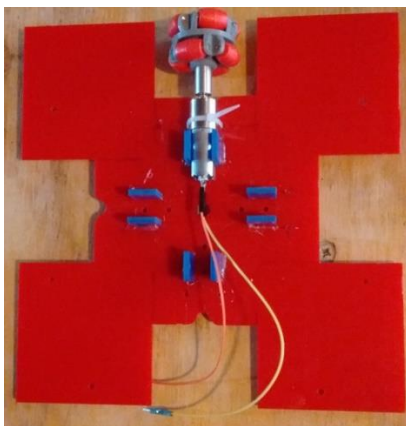
Ilustración 89 Placa inferior de los robots

Se procedió a ensamblar la parte electromecánica de las ruedas, uniendo los motorreductores el eje de las llantas y las llantas omnidireccionales (ilustración 74), verificando su ensambles y tolerancias para ser instaladas a la placa inferior de los robots, como de evidencia en la Ilustracion 90.

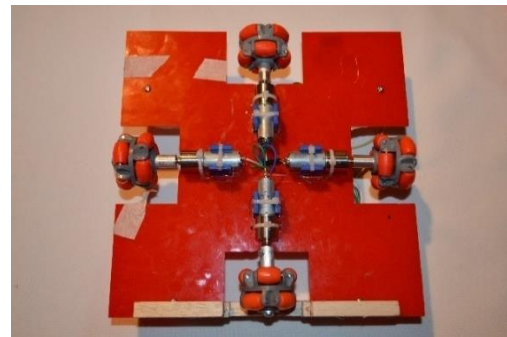


Ilustración 90 Ensamble motorreductor eje y llanta omnidireccional

La instalación de los ensambles de las ruedas de cada uno de los robots, verificando su correcta posición y dirección (Ilustración 91 (a) y (b)).



(a)



(b)

Ilustración 91 Placa inferior con motorreductor y rueda omnidireccional (a) y placa inferior del Robot Azul con los motorreductores (b)

Los ensambles de los motores fueron instalados en la placa del robot Azul, utilizando las piezas de acoplamiento y manejando el cableado en la parte central del mismo como se identifica en la ilustración 82, instalación de las partes electrónicas, tarjeta de control de los motores y tarjeta de desarrollo implementada para el sistema, sensor de comunicación (Ilustración 92).

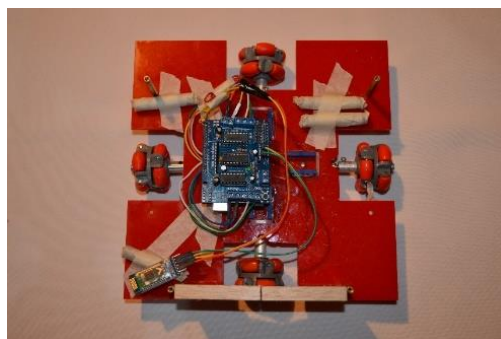
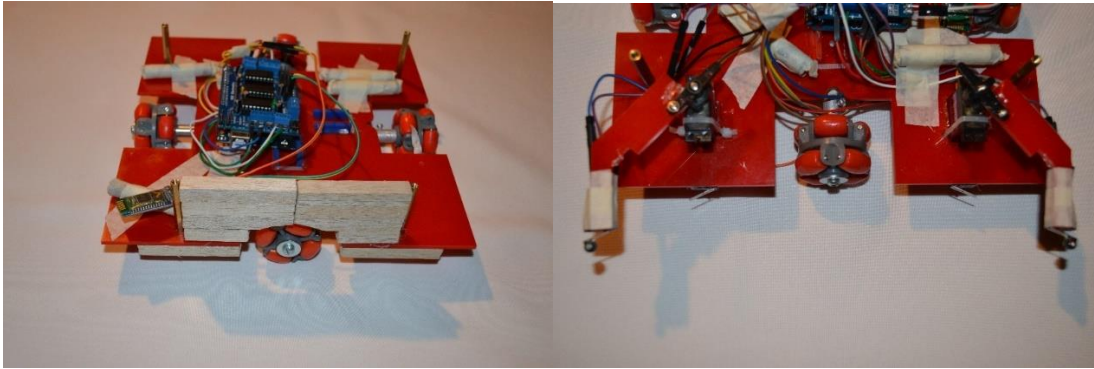


Ilustración 92 Robot Azul con la parte electrónica



(a)

(b)

Ilustración 93 Robot Azul piezas del acoplamiento hembra y Robot Rojo piezas de acoplamiento macho

El sistema de acoplamiento está dividido en dos partes que son:

Parte 1: el robot Azul tiene la pieza hembra del sistema, que está compuesto por los dos tornillos de unión de placas y un soporte entre los dos tornillos para que encado de que el robot acoplador pueda corregir su posición (Ilustracion 93(a)).

Parte 2: el robot Rojo tiene un sistema eléctrico y mecánico, compuesto por dos servomotores, dos piezas en forma de pinza y 4 sensores de contacto, los sensores de las pinzas son los encargados de detectar la mala posición de acoplamiento del robot Rojo y los sensores inferiores confirman el contacto con el robot Azul para ejecutar el subprograma que ajusta las pinzas al robot Azul, confirmando así el acoplamiento. Ilustracion 93 (b).

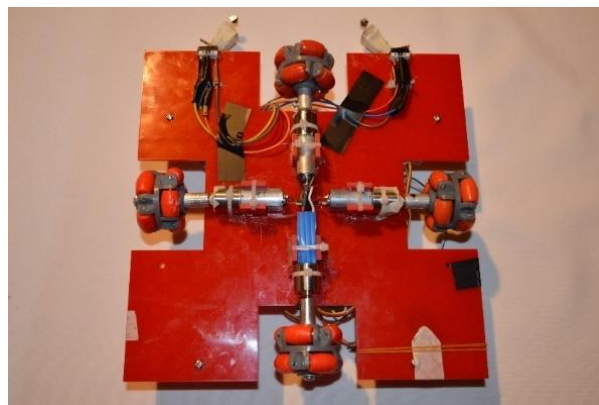


Ilustración 94 Placa inferior del Robot Rojo con los motorreductores

Al construir los dos robots se verifico la posición de los motorreductores y de la parte electrónica así mismo los elementos del sistema de acoplamiento antes mencionado, podemos verlo en la Ilustracion 94 e Ilustracion 95 , parte inferior y parte superior respectivamente.

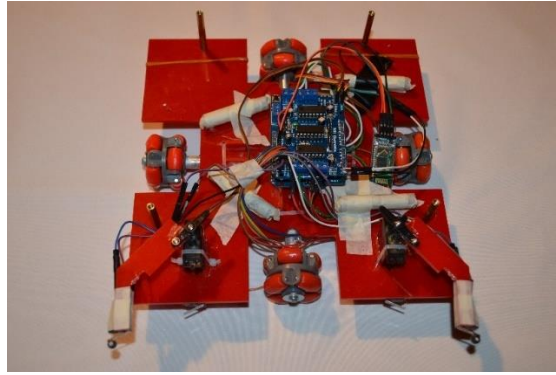


Ilustración 95 Robot Rojo parte superior con el sistema electrónico

5.2 Desarrollo del Sistema Multi-Agente

El presente sistema Multi-Agente se desarrolló para la implementación de la estrategia de solución de la tarea, que consiste en detectar un objeto, ubicar su posición y generar el algoritmo que, por medio de técnicas de control de Inteligencia Artificial, llevarlo a un punto de entrega, disponiendo de tres agentes (Ilustración 96), dos Agentes Robot y un Agente Supervisor (Programa).

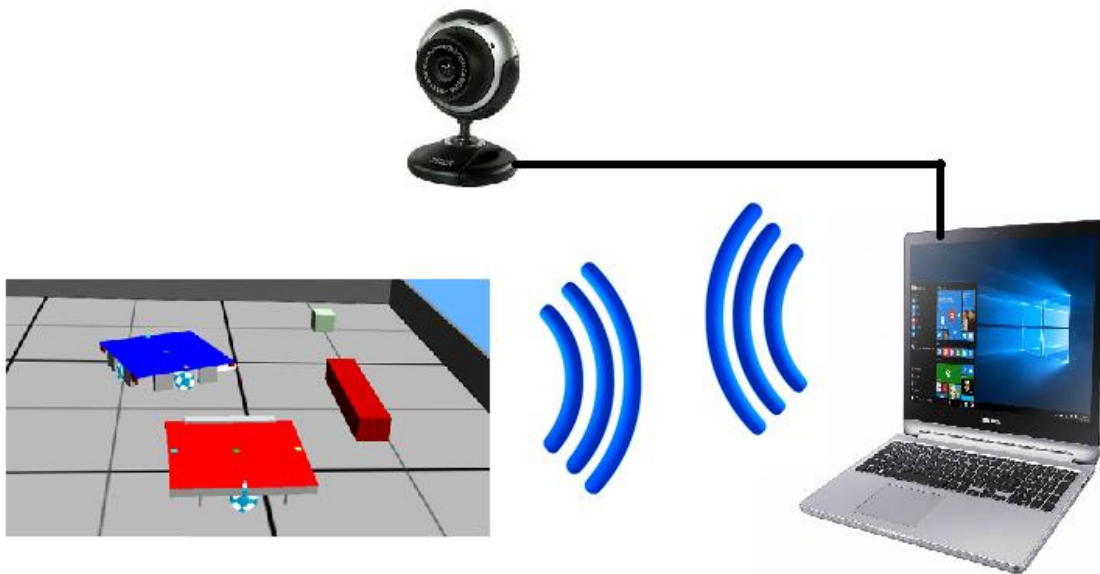


Ilustración 96 Sistema multi-agente

5.2.1 Programa del Agente Supervisor

El supervisor es el encargado de la capa deliberativa del sistema, siendo el tercer Agente del sistema Multi-Agente (MAS), este agente será de tipo software desarrollado en el programa Matlab®, con don componentes físico que son:

- a. El portátil que será el encargado de ejecutar el software y empleando el Bluetooth empleando dos canales para la comunicación con los dos robots.
- b. La cámara web que estará en tiempo real tomando las imágenes y estará conectada con el computador, teniendo en cuenta que los robots no poseen visión de máquina, el software contará con ella por medio de la cámara.

5.2.1.1 Forma de captura de la posición de los robots en la visión de maquina

El método empleado para la detección de los robots por medio de la visión de maquina será el empleo de la técnica de umbralizacion de colores que consiste en separar los elementos de la imagen que nos interese, se trata de asignar a cada pixel aun cierto grupo, llamado segmento, teniendo en cuenta los valores RGB de cada pixel, la pertenencia de cada pixel a un cierto grupo, se decide mediante la comparación de su nivel de gris con un cierto valor de umbral con lo cual se podrá identificar las seis variables, que determinan cada uno de los colores empleando este sistema para:

1. Detectar cada uno de los colores, por medio de un programa de umbralizacion realizar la búsqueda de los valores.
2. Determinar qué color está relacionado con cada robot y con su marcador así mismo determinar el objeto y su orientación.
3. Detectar el centroide del color lo cual representara el centro del objeto que necesitamos saber su ubicación.

5.2.1.2 Capa deliberativa del Agente Supervisor

Fase 1

1. Inicia la comunicación con los dos robots para confirmar su participación empleando los canales asignador por el bluetooth para cada robot.
2. Detecta las posiciones de los robots, empleando la umbralizacion detectando sus correspondientes marcadores, posición del objeto y su marcador determinado de cada uno su orientación y creando la trayectoria para el movimiento de cada uno de los robots, determinando las posiciones de acoplamiento y el ángulo con respecto

al ángulo del objeto a recoger, determina la distancia y el ángulo necesario de cada uno de los robots.

3. Les comienza a verificar la posición y orientación para que puedan corregirla durante su desplazamiento.

Fase 2

1. Al momento del llegar a las posiciones de acoplamiento, verificar que tengan la orientación indicara, efectúen el acoplamiento y la recolección del objeto.

Fase 3

2. Monitoreando el desplazamiento de la plataforma con el objeto y determinando puntos de control en los cuales verifica el ángulo y la distancia para seguir orientando a los robots al punto de entrega.

5.2.2 Programa de los Agentes Robot Azul y Robot Rojo

5.2.2.1 Capa reactiva de los robots

Fase 1

1. Inician con el inicio de la red de comunicaciones por medio de los sensores bluetooth que tiene cada uno con el Agente software (supervisor), al tener confirmación de la conexión esperan ordenes, para que les indique que acción van a realizar.
2. Reciben la orden de iniciar con el movimiento e inician con el desplazamiento, tomado la trayectoria indicada por el Agente software, durante el trayecto recibe acciones de corrección de dirección propia y de orientación al objetivo.

Fase 2

1. Reciben la información que los dos están en posición e inician con el subprograma de acoplamiento, el robot a ser acoplado se queda quieto y el que acopla inicia movimiento hacia el otro robot.
2. Los sensores que tiene en las pinzas de acoplamiento le permiten corregir la dirección en caso de ir corrido con respecto al otro robot.
3. Los sensores del Robot Rojo, determina la distancia ideal para el acoplamiento a si mismo las pinzas están diseñadas para que corrija cualquier desviación en el acoplamiento.

4. Los sensores determinan el acoplamiento e informan al supervisor que están acoplados esperando el objeto.

Fase 3

1. Los robots inician la tarea de forma colaborativa y controlada hacia la posición de entrega en la cual finalizara la tarea colaborativa en el momento que llevaron el objeto al final.
1. Los robots en su movimiento como una sola plataforma se mueven conservando el norte empleando el modo de movimiento colaborativo, por la estructura de las llantas puede moverse de forma deseada.
2. Los robots van corrigiendo el ángulo durante el desplazamiento y recibiendo la información del supervisor, sobre su ubicación y como llegar al punto indicado.

5.3 Costos de la implementación de los robots

Los costos de Implementación de los robots y el sistema multi-agente, con sus correspondientes elementos están representados en la Tabla 10 que a continuación se relaciona, con un valor total invertido de \$ 882.000 pesos.

Tabla 10 Gastos del proyecto

Detalle	cantidad	Valor Unid.	Valor
Robots diferenciales	2	\$ 40.000	\$ 80.000
Placa superior acrílico 3mm x 220 mm x 220 mm	2	\$ 12.000	\$ 24.000
Placa inferior acrílico 3mm x 220 mm x 220 mm	2	\$ 12.000	\$ 24.000
Pinzas acrílico 3 mm	4		\$ 6.000
Soportes motores acrílico 3 mm	16		\$ 8.000
Tornillos 4 cm	8		\$ 7.000
Motorreductores	8	\$ 25.000	\$ 200.000
Ruedas omnidireccionales de 40 mm diámetro	8	\$ 30.000	\$ 240.000
Tarjeta desarrollo	2	\$ 30.000	\$ 60.000
Pulsadores	8	\$ 500	\$ 4.000
Módulos de comunicación HC-05	2	\$ 18.000	\$ 36.000
Tarjeta controladora de motores L293	2	\$ 18.000	\$ 36.000
Baterías recargables AA	16		\$ 120.000
Cableado			\$ 12.000
Varios			\$ 25.000
Valor Total			\$ 882.000

CAPITULO VI

Pruebas y Resultados

6.1 Programa de identificación de colores

El programa capta la imagen de la cámara, después de detectar la imagen seleccionamos iniciar muestreo y luego comenzamos a variar las variables de la umbralización, buscando detectar cada color en la Ilustración 97, vemos que detecto el color asignado al Robot Rojo, el color azul claro, en la imagen lo ubica y mediante una función se obtiene el centroide de la figura, el centroide nos determinara la posición real del objeto en la visión de máquina, el diagrama de flujo del programa Ilustración 98.

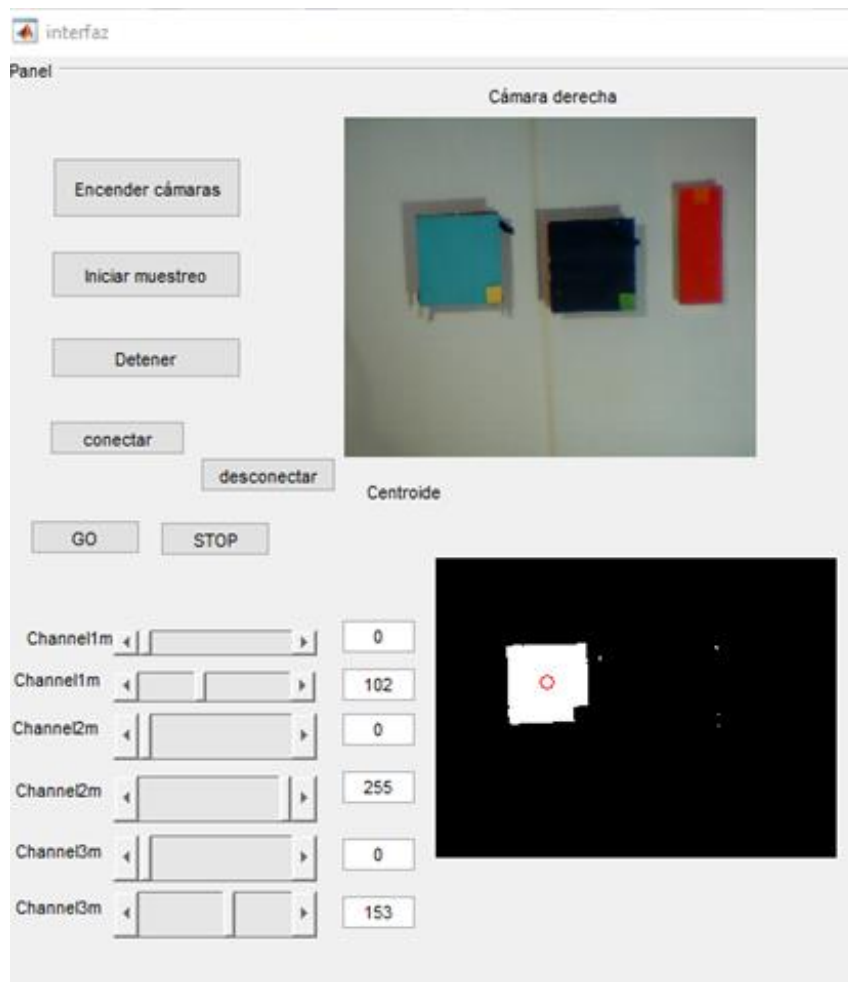


Ilustración 97 Ventana del programa de umbralización de colores

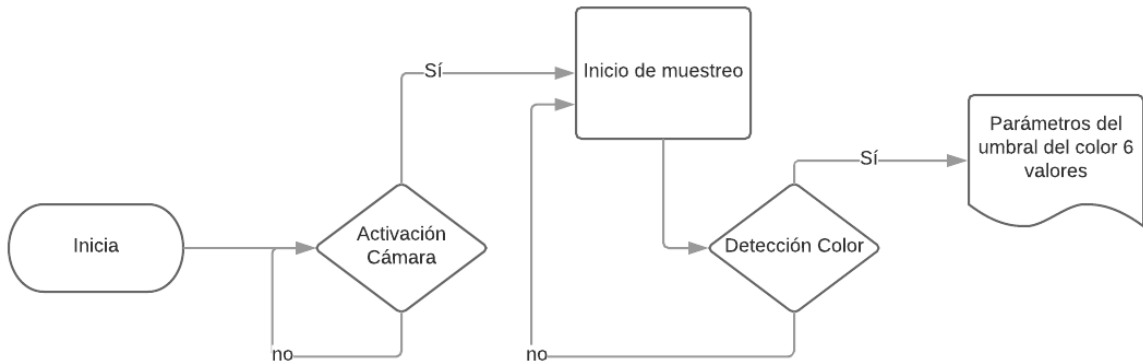


Ilustración 98 Diagrama de flujo del programa de detección de colores

6.2 Programa del Agente Supervisor

Este programa será el encargado por medio de la visión de máquina y la red de comunicación que emplea el bluetooth, monitorear las posiciones de los robots, los marcadores de cada color y el objeto, información que será procesada para obtener la orientación de cada uno de los elementos del sistema, el programa cuenta con 3 imágenes que son: la imagen de la cámara, la imagen detección de los robots, objeto y marcadores, información de las posiciones en tiempo real y los deltas de los robot y el estado en el cual se encuentran (Ilustración 99).

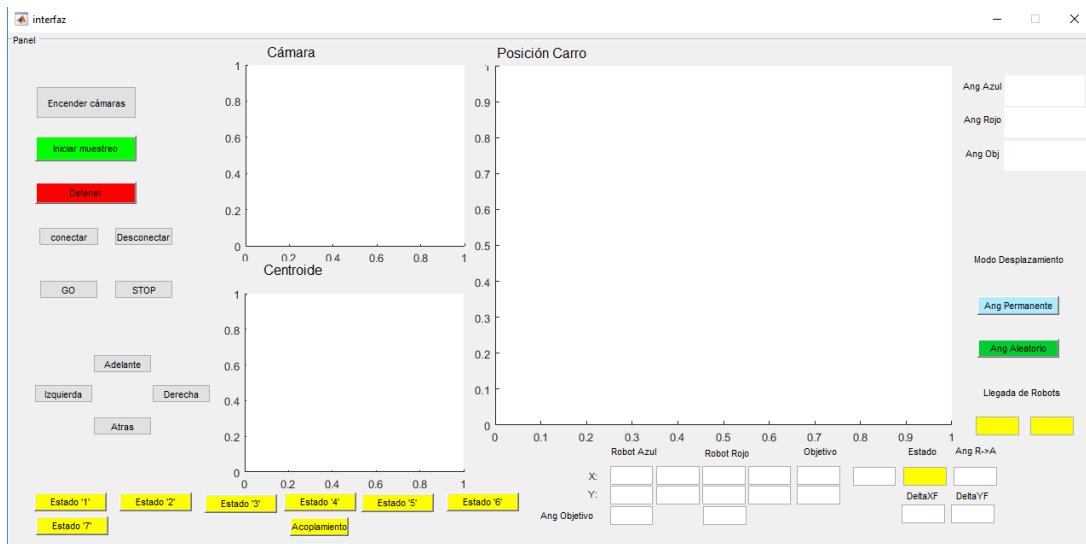


Ilustración 99 Ventada del programa del Agente software

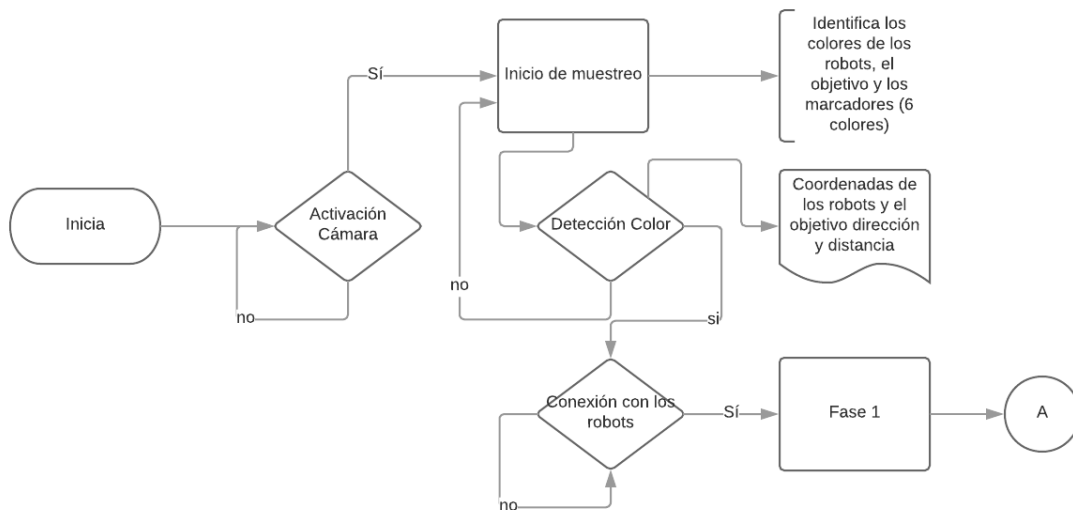


Ilustración 100 Diagrama de flujo del Agente Supervisor

El diagrama de flujo de la ilustración 100 muestra el inicio del programa del agente supervisor, ubicando los robots y el objetivo para dar inicio a la Fase 1 del programa, como punto de conexión del diagrama de flujo la letra A.

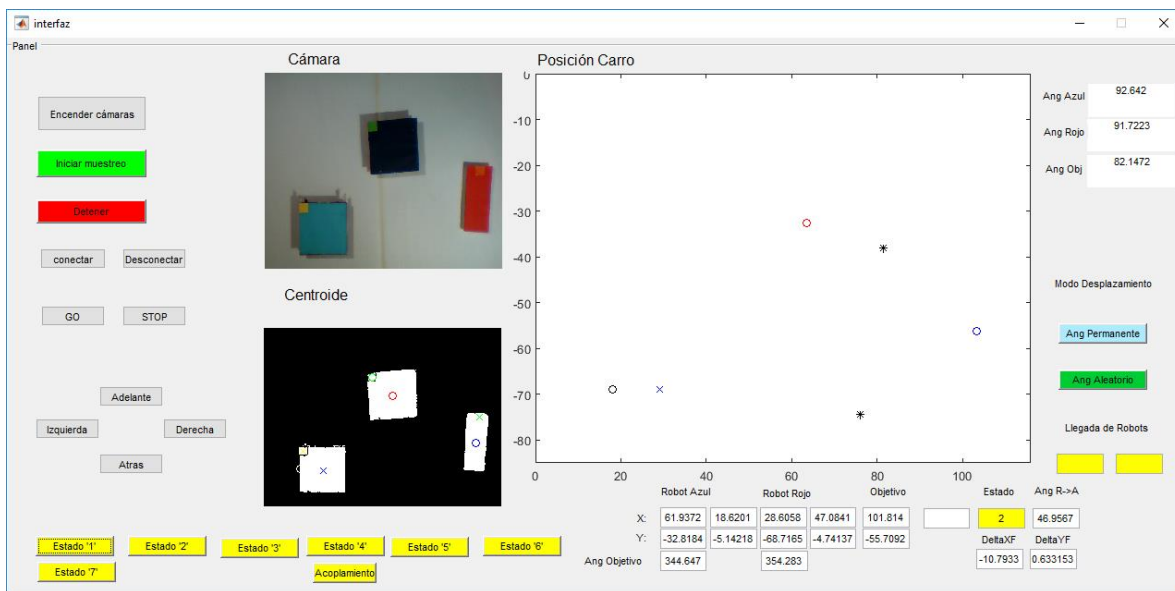


Ilustración 101 Programa en ejecución

En la Ilustración 101 verificamos como el ubica los robots, el objeto y los marcadores, realiza la ubicación en la ventana posición carro y espera órdenes para iniciar con el planeamiento de trayectoria de los robots hacia el objetivo ejecutando la primera Fase.

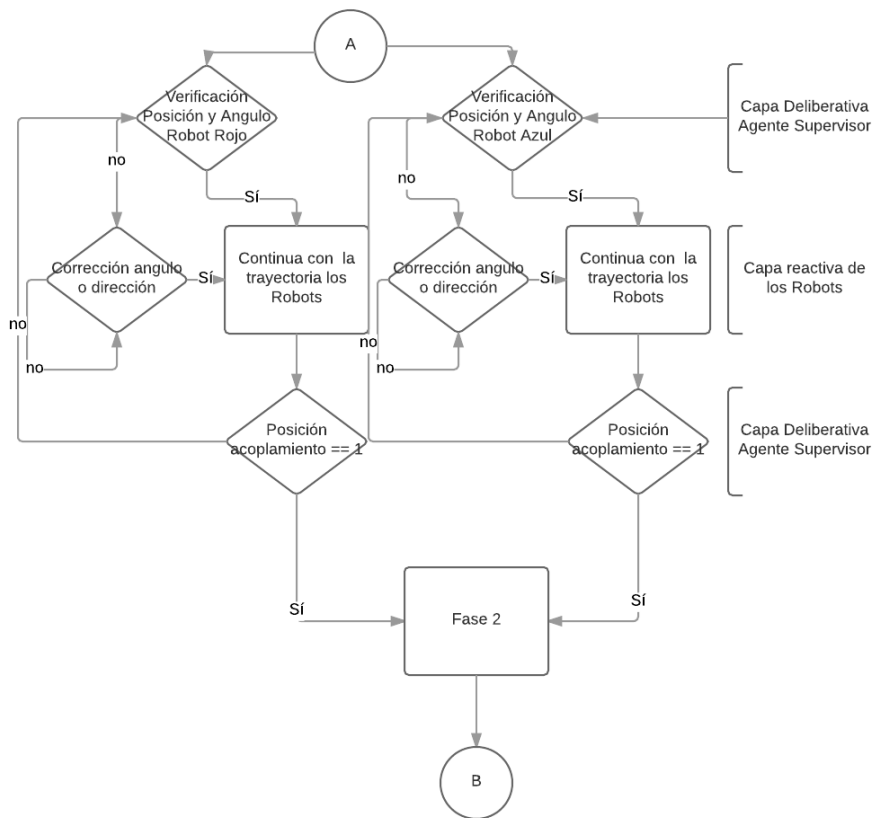


Ilustración 102 Diagrama de flujo fase 1

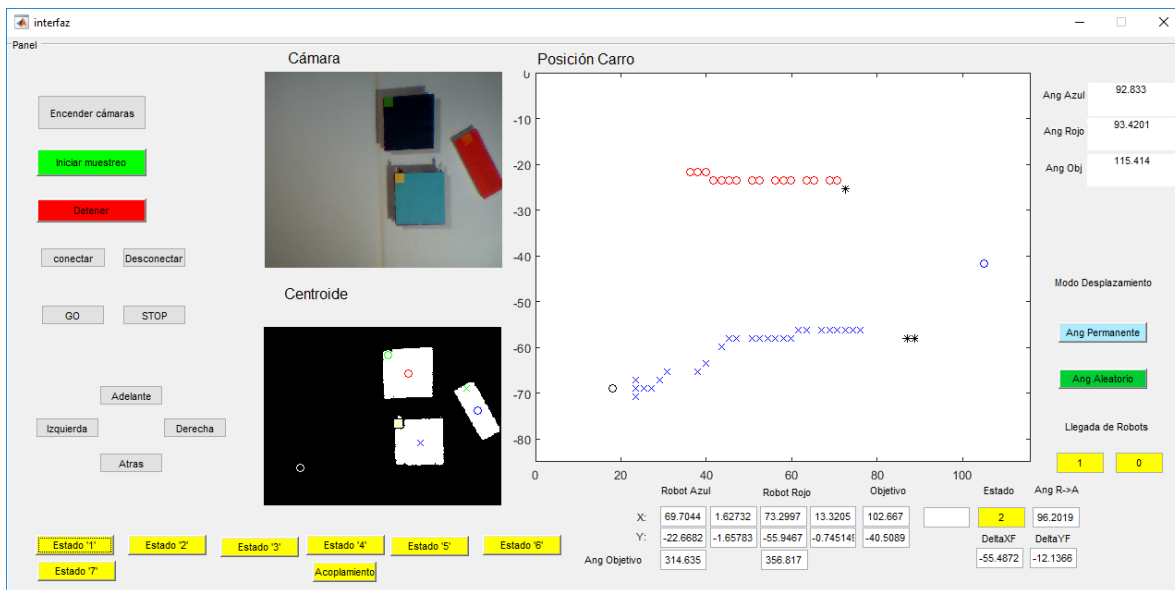


Ilustración 103 Programa en ejecución del desplazamiento de los robots

El Agente software va graficando la trayectoria que van tomando los robots durante el desplazamiento a las posiciones de acoplamiento, en el diagrama de flujo de la Ilustración 102 determinamos el comportamiento del control reactivo de los Agentes Robots y el comando deliberativo del Agente Supervisor Ilustración 103.

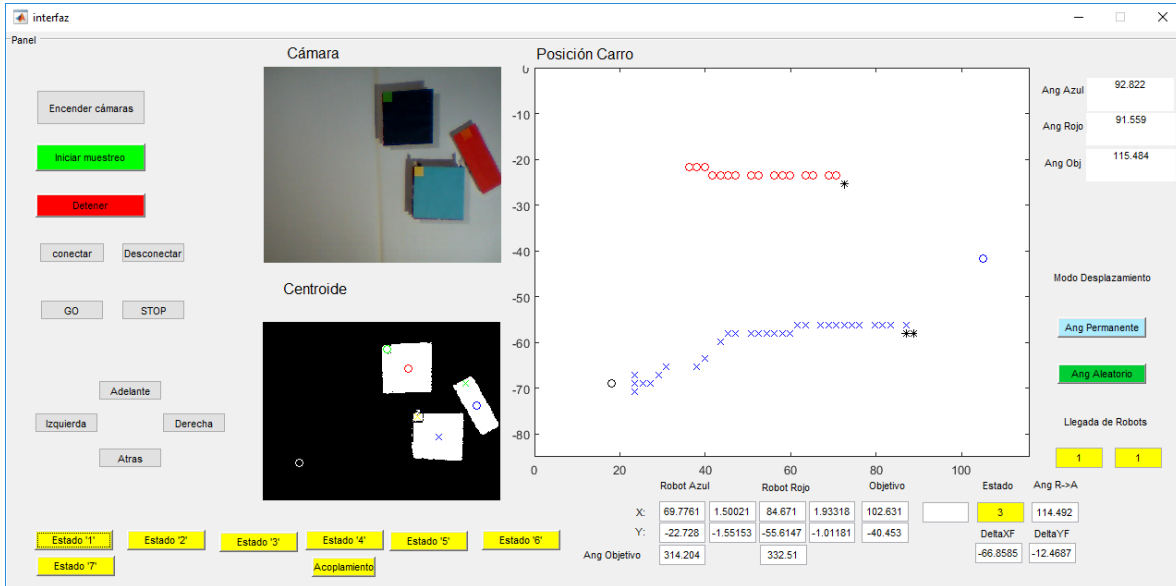


Ilustración 104 Programa en Fase 2

En la Fase dos los dos robots llegaron a las posiciones de acoplamiento y se disponen a iniciar con el subprograma de acoplamiento, el Agente software monitorea que cada uno de ellos llegue a su correspondiente ubicación y confirma que tomen la dirección paralela al objeto a recoger (Ilustración 104).

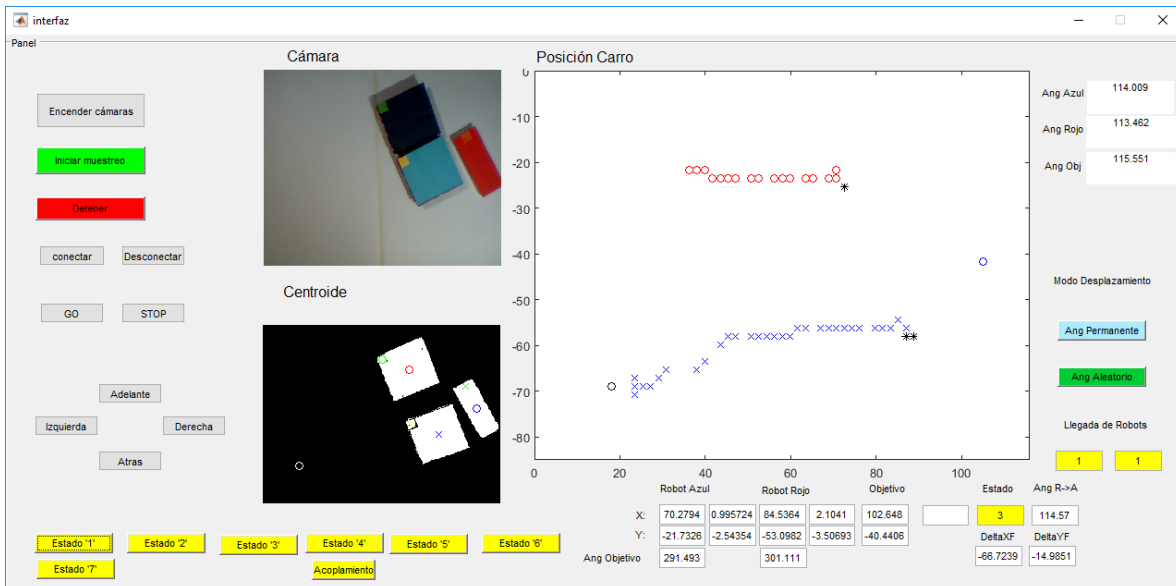


Ilustración 105 Fase 2 acoplamiento

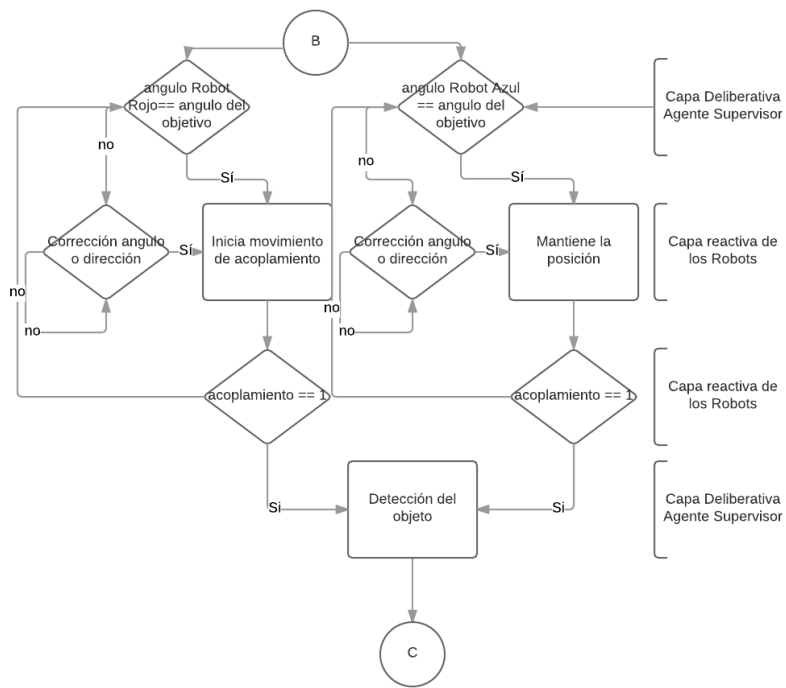


Ilustración 106 Diagrama de flujo de la Fase 2

En la Ilustración 105 vemos como los robots se acoplaron y entraron en el modo de recepción del objeto, el cual será colocado en la plataforma, el objeto es mayor al lado de los robots, lo cual nos determina el acoplamiento de los robots, diagrama de flujo de la Fase 2 Ilustración 106.

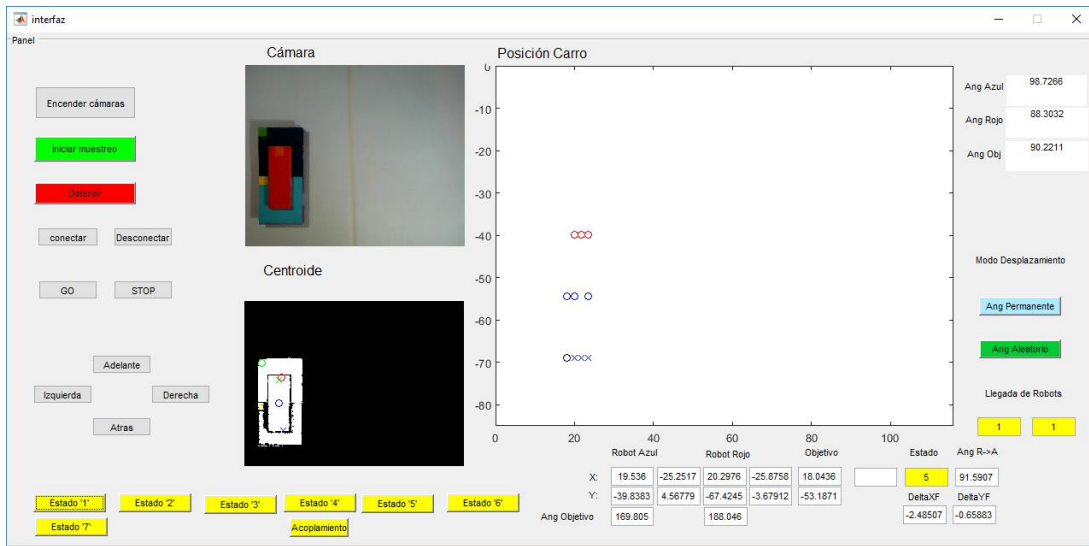


Ilustración 107 Fase 3

La Fase 3 está compuesta por el movimiento de la plataforma al punto de entrega puesto de manifiesto en la Ilustración 107, manteniendo el ángulo toda la plataforma, empleando el modo de movimiento colaborativo de los robots, diagrama de flujo de la Fase 3 Ilustración 108.

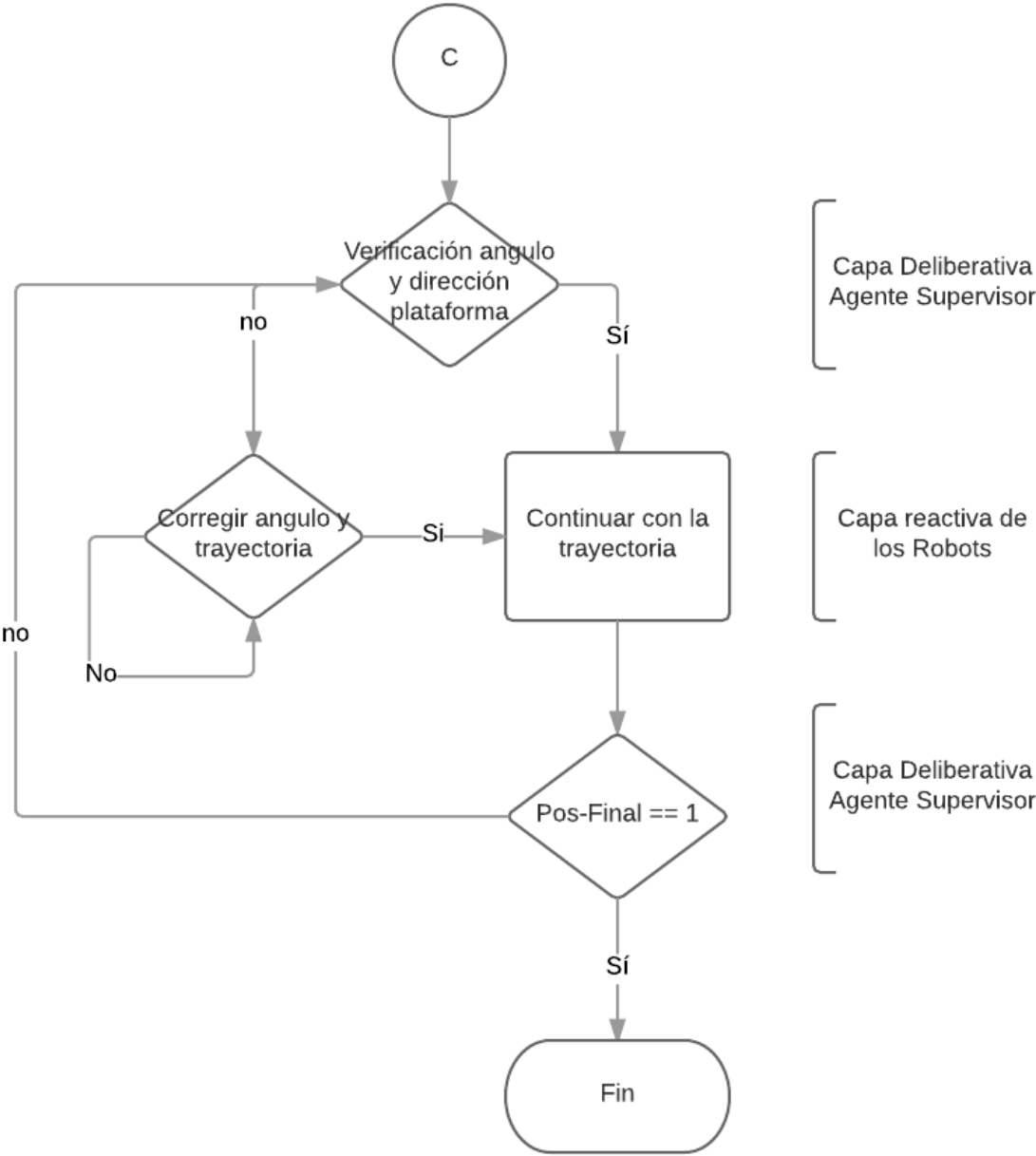


Ilustración 108 Diagrama de flujo Fase 3

6.3 Resultados de las trayectorias de los Robots Vs tiempo

6.3.1 Trayectorias directas

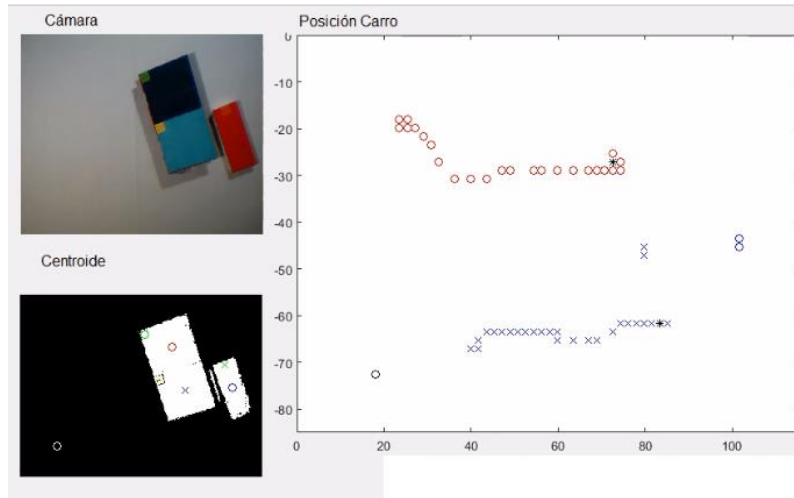


Ilustración 109 Grafica de la trayectoria de los robots en la interfaz de usuario

La ilustración 109 muestra la trayectoria de los robots durante la Fase 1 en el movimiento a los puntos de acoplamiento y la Fase 2 de acoplamiento, círculo rojo son los puntos de la trayectoria del robot Azul y la x azul los puntos de la trayectoria del robot Rojo, el círculo azul es la ubicación del objeto y los dos asteriscos negros son los puntos de acoplamiento.

Trayectoria del robot Azul

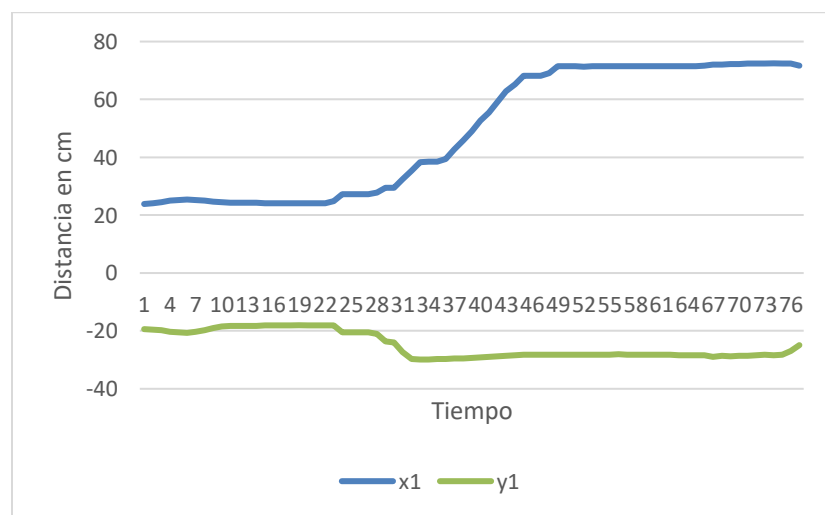


Ilustración 110 Grafica del desplazamiento en X Y del robot Azul

El desplazamiento del Robot Azul y Rojo en X Y está representado en la ilustración 110 y 111, los datos solo los obtenidos por medio de la interfaz de usuario, datos tomados en tiempo real, la gráfica muestra los momentos en los cuales el robot corrige el ángulo la gráfica muestra que no hay desplazamiento apreciable, la pendiente de la recta tiende a 0 y el movimiento cuando la pendiente es mayor a 0.

Trayectoria del robot Rojo

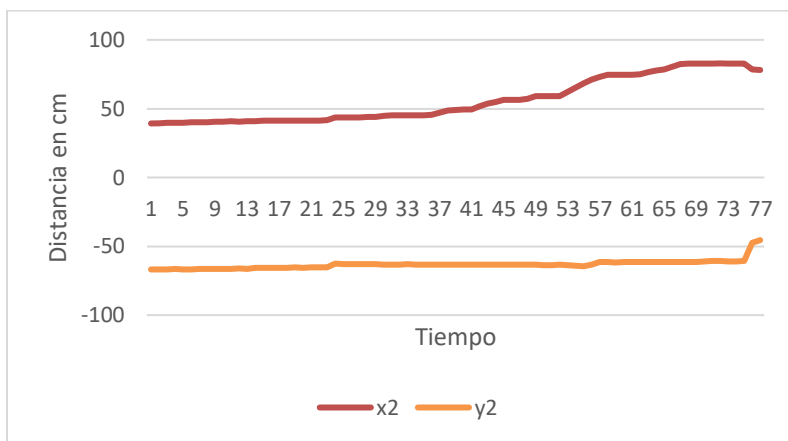


Ilustración 111 Grafica del desplazamiento en X Y del robot Rojo

Grafica de los ángulos

El algoritmo mantiene el ángulo ideal de los dos robots de $\pi/2$ o 90 grados, como muestra la gráfica de la ilustración 112, línea roja Robot Rojo y línea azul Robot Azul. Los robots iniciaron en un ángulo diferente a $\pi/2$, la gráfica representa el movimiento del robot para encontrar el ángulo ideal y como durante su desplazamiento va corrigiéndolo, al final de la gráfica los robots están tomando el ángulo del objeto para realizar el acoplamiento.

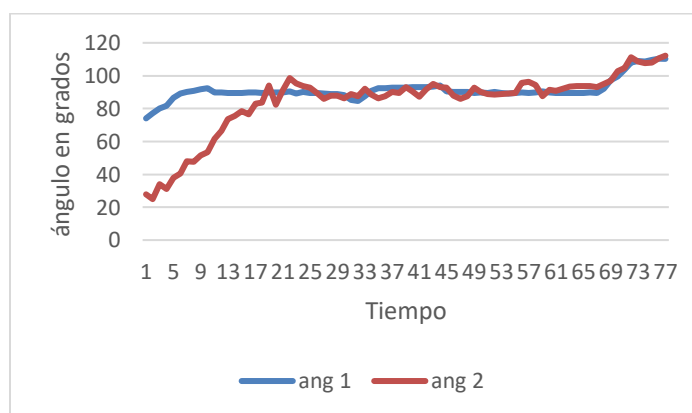


Ilustración 112 Grafica de la orientación de los robots

6.3.2 Cruce de trayectorias

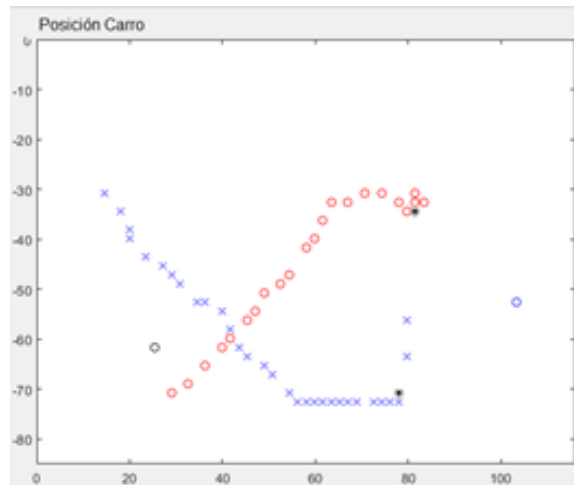


Ilustración 113 Grafica de posiciones de los robots al objeto

Los robots en la ilustración 113 inician desde puntos que hacen que su trayectoria se cruce en su camino a los puntos de acoplamiento, el algoritmo al determinar su posición cruzada da prioridad el robot más cercano a su punto de acoplamiento para que inicie su movimiento mientras el otro se mueve solo en el eje y, reduciendo su DeltaY hacia su punto de acoplamiento evitando el contacto entre los robots.

La ilustración 114 y 116 representan el movimiento de los robots en el eje X con relación al tiempo, teniendo en cuenta que son omnidireccionales le permite moverse de forma estable y directa en cualquier dirección hacia el punto de acoplamiento.

Las gráficas corresponden a los valores reales tomados de interfaz de usuario, datos que fueron tomados por el supervisor, donde el comportamiento de los robots está determinado por el algoritmo diseñado.

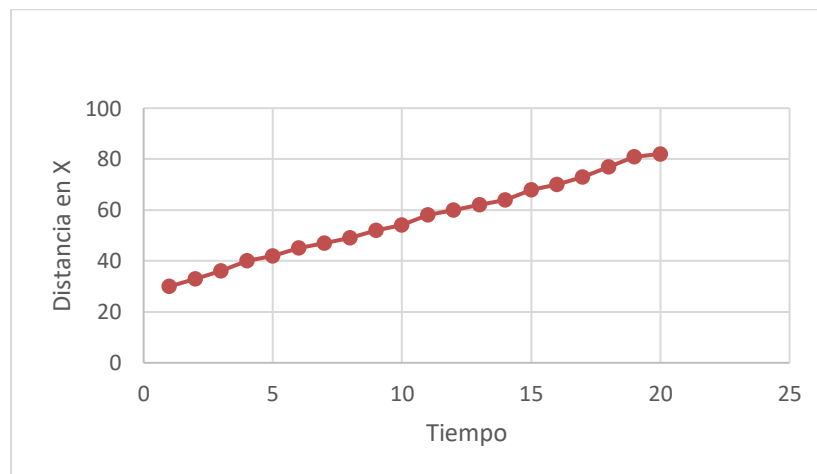


Ilustración 114 Grafica de desplazamiento del Robot Móvil Rojo distancia en X vs tiempo

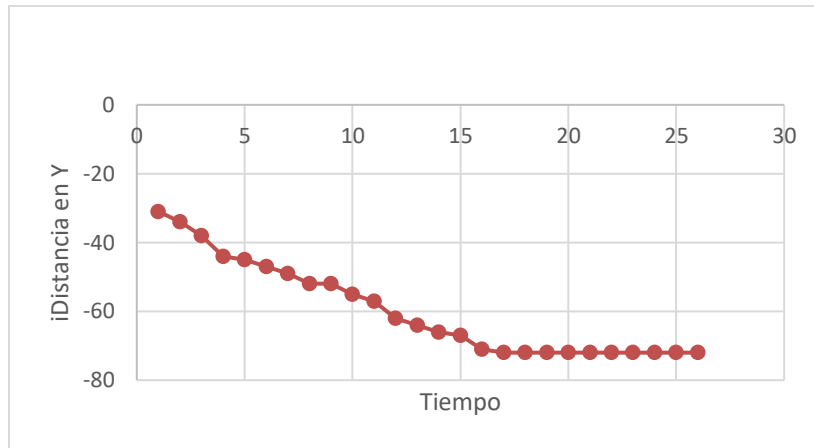


Ilustración 115 Grafica de desplazamiento del Robot Móvil Rojo en Y vs tiempo

Las gráficas de las ilustración 115 y 117 son la representación de los punto en Y de la trayectoria de los dos robots y se observa que las trayectorias se cruzan en la gráfica, pero en el desarrollo de los robots, el algoritmo determinó el cruce y ejecutó el programa para evitarlo, visto en el desarrollo de los puntos con relación al tiempo de ejecución, los valores de tiempo están relacionados directamente con 0.6 seg de toma de imagen, correspondiendo a la gráfica 1 -> 0.6 seg, 2 -> 1.2 seg.

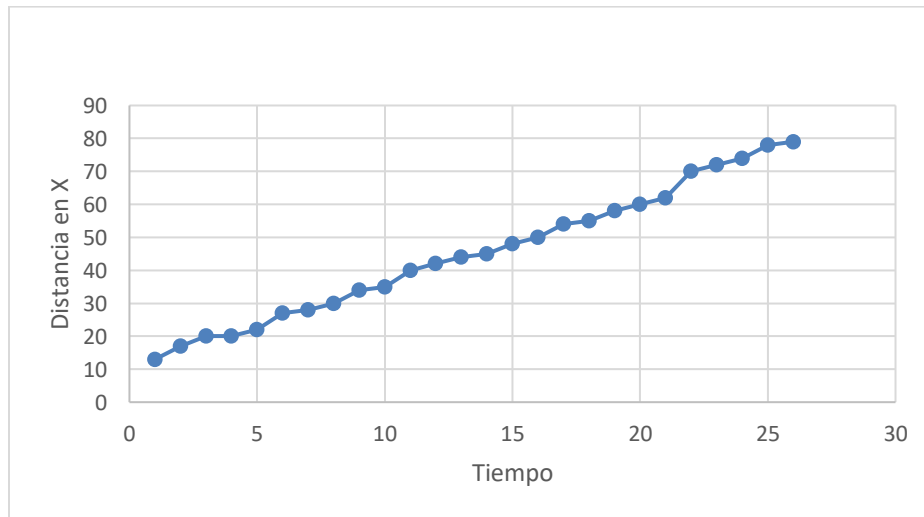


Ilustración 116 Grafica del Robot Móvil Azul distancia en X vs tiempo

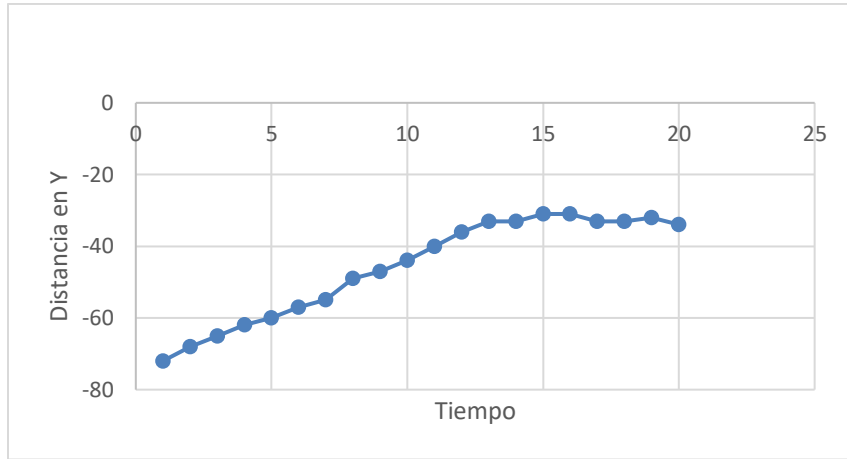


Ilustración 117 Grafica del Robot Móvil Azul distancia en Y vs tiempo

Conclusiones

- La tarea seleccionada de transportar un objeto del cual se determinaría posición y orientación, teniendo unas dimensiones superiores al largo de los Robots, generó una gran cantidad interrogantes que permitió el desarrollo del robot prototipo en cuanto a la estructura, disposición de los motores, sistema de acoplamiento, desarrollo de la arquitectura del software, haciendo del proyecto un desarrollo muy interesante. Todo esto dio como resultado dos robots omnidireccionales que pueden trabajar de forma colaborativa y la construcción de un sistema Multi-agente, que fue parte fundamental en el trabajo colaborativo.
- En la investigación se plantearon 3 prototipos, un robot diferencial y dos robots omnidireccionales Móvil 002 y 004; fue descartado el robot diferencial porque presenta restricciones al movimiento para su desplazamiento en el área de trabajo y el sistema de acoplamiento de los robots sería de mayor complejidad; en cambio los robots omnidireccionales permiten un movimiento sin restricciones, al ser acoplados se comportarían como un solo robot omnidireccional siendo una gran ventaja para la investigación.
- El diseño del prototipo del Robot Móvil 002 y 004, permitieron acercarse a las respuestas físicas y dinámicas de los robots, al ser importados y manipulados en el programa Webots®. Esas pruebas contribuyeron a tomar la decisión más pertinente en lugar de iniciar con la construcción de dos prototipos que sería evaluados físicamente, reduciendo no solo el gasto económico, si no el tiempo invertido en su construcción.
- El comportamiento de los robots en el mundo simulado fue muy cercano al real, ejecutando las tres fases de la tarea colaborativa, lo que permitió evaluar la configuración elegida del robot omnidireccional (cuatro ruedas). Se utilizaron métodos de inteligencia artificial clásica, los cuales ayudaron a definir el algoritmo principal del sistema colaborativo para ser implementado. Todo esto contribuyó a la reducción de tiempos de desarrollo, y a la evaluación de las diferentes situaciones que se podían presentar dando como resultado un sistema multi-agente compuesto por dos agentes homogéneos y un agente software.
- El estudio de la robótica móvil permite conocer los diferentes robots que podrían ser implementados en la solución del problema de la investigación; al analizar la estructura, los tipos de ruedas, tipos de entornos y de locomoción, se estableció los prototipos diferencial y omnidireccional.

- Los robots móviles omnidireccionales al tener en cuenta su estructura, configuración, capacidades y usos; son de un gran potencial en el ámbito de la investigación, donde generalmente se emplean robots móviles diferenciales incentivando así el uso de estos robots en próximas investigaciones.
- El CAD de las ruedas omnidireccionales, es muy cercano a las reales, en cuanto elementos, medidas y diseño, pero al ser implementados en la simulación de Webots® los rodillos no pudieron ser ajustado a su comportamiento real, su giro infinito en la dirección perpendicular del movimiento de la rueda generó que en la simulación una fuerza de fricción contraria al movimiento del robot perdieran fricción, el software lo identificaba como un sólido y el robot debía arrastrar la rueda, aumentando los tiempos de simulación y prueba, la solución del problema fue aumentar la velocidad de las ruedas para que se redujera la fricción dinámica y permitiera el movimiento deseado.
- El procesamiento computacional del programa Webots® es alto en relación con las características físicas de los robots esto hace que, si no se simplifica al máximo el CAD al ser importado, el programa presentara fallas y errores que limitaran la simulación, razón por la cual se simplifican las piezas en sus características físicas suprimiendo piezas pequeñas que no hagan parte del movimiento del robot como las baterías y tornillos. Los elementos que no pueden suprimirse dando su importancia son los que producen movimiento: llantas y pinzas, y los elementos electrónicos como los sensores que son necesarios para interactuar con el mundo creado.
- El sistema electrónico compuesto por la tarjeta de desarrollo, tarjeta de control de motores, sensores de contacto, módulo y las baterías; cumplieron con los requerimientos del sistema para la implementación del robot en el desarrollo de la tarea, ocupando casi todos los puertos de la tarjeta, permitiendo que la programación de la capa reactiva de los robots cumpliera con su propósito.
- El Robot móvil 004 fue la configuración adecuada para ser empleada en el desarrollo del proyecto, al utilizar las cuatro ruedas esto permitió que el acoplamiento se efectuará de mejor forma, permitiendo controlar el movimiento durante la integración de los dos robots.
- En el desarrollo de la programación de las capas deliberativa y reactivas, parte fundamental de su comportamiento fue la división de la ejecución en tres Fases, Fase 1: movimiento hacia el objetivo a puntos de acoplamiento, Fase 2: acoplamiento de los robots y carga del objeto a transportar y Fase 3: movimiento a punto de entrega.

Fases que permitieron mejorar el programa de cada uno de los agentes del sistema, y delimitar las dos capas de la programación.

- El material seleccionado para la estructura fue el ideal en su comportamiento y valor, siendo una estructura fuerte en el acople de los dos robots. El espacio entre placas permitió colocar el sistema electrónico en un lugar seguro y estético, permitiendo que los robots se vieran bien y organizados.
- El sistema Multi-Agente, involucró un tercer agente (Supervisor), ya simulado inicialmente en Webots®. Este agente adiciono al sistema la visión de maquina fundamental y comunicaciones, a un sistema que tenía solo dos robots móviles, que no podían ubicarse fácilmente en el mundo creado para interactuar y desarrollar una tarea de búsqueda y transporte de un obstáculo.
- El Agente Supervisor utilizo una librería de Matlab® para realizar la visión de máquina para la identificación de colores, empleo la umbralizacion de los pixeles de la imagen tomada por la cámara y la identificación de sus centroides, esta función permitió la identificación del 6 colores diferente que a su vez permitieron la identificación de la ubicación y orientación de los robots y el objeto, valores que fueron muy importantes en el desarrollo del programa, como en el planeamiento de trayectorias.
- Las capas de programación del sistema, capa deliberativa y reactiva, fueron fundamentales para el desarrollo de la programación determinando cada paso a ser programado en cada una de las fases, dándole más capacidades a cada uno de los agentes dando una jerarquía u organización, que fue fundamental en el desarrollo de la ejecución del programa.

BIBLIOGRAFÍA

- [1] Hayward, V., Hayati, S. (1988). KALI: an environment for the programming and control of cooperative manipulators. American Control Conference, pp. 473-478.
- [2] Acosta, G., (2010). Ambiente multi-agente robótico para la navegación colaborativa en escenarios estructurados.
- [3] Martínez, M., (2011). *Sistema de control colaborativo de robots autónomos basado en planificación automática*. Universidad Carlos III de Madrid.
- [4] Merino, L., Capitán, J. y Ollero, A. «Robótica cooperativa e integración con sensores en el ambiente. Aplicaciones en entornos urbanos.» [En línea]. Disponible en: <http://www.upo.es/isa/lmercab/papers/robot09Spanish.pdf>. [Accedido: 14-may-2014].
- [5] Bano, N., Roppel, T. y Gokhale, I. (2010) «Use of mobility models for communication in collaborative robotics», in 2010 42nd Southeastern Symposium on System Theory (SSST), , pp. 143-146.
- [6] Easton, K. y Martinoli, A. (2002) «Efficiency and optimization of explicit and implicit communication schemes in collaborative robotics experiments», in IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2795-2800 vol.3.
- [7] Builes, J., Carranza, D., y Bedoya, J. «Comunicación en sistemas de múltiples robots desde la metodología MAD-Smart», Ing. E Investigación., vol. 28, no. 2, pp. 59-65, 2008.
- [8] Estepa, R., Rosário, J. y Torres, G. (2013) «Estrategia de Coordinación y Comunicación para Sistemas Robóticos Colaborativos», Sci. Tech., vol. 18, no. 1, pp. 101-107,
- [9] Ducatelle, F., Caro, G., Gambardella, L. Cooperative Self-Organization in a Heterogeneous Swarm Robotic System, 2010.
- [10] Beni, G. (2005) «From swarm intelligence to swarm robotics», in Proceedings of the 2004 international conference on Swarm Robotics, Berlin, Heidelberg, pp. 1-9.
- [11] Ferrante, E. «The Design of a Modular Behavior-based Architecture». [En línea]. Disponible en: http://www.academia.edu/2559073/The_Design_of_a_Modular_Behavior-based_Architecture. [Accedido: 12-may-2014].
- [12] Weiss, G. «Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence: Weiss, G.: 9780262731317: Amazon.com: Books». [En línea]. Disponible en: <http://www.amazon.com/Multiagent-Systems-Distributed-Artificial-Intelligence/dp/0262731312>. [Accedido: 15-oct-2013].
- [13] «Intelligent Software Agents». [En línea]. Disponible en: <http://www.cs.cmu.edu/~softagents/multi.html>. [Accedido: 12-may-2014].

- [14] «Vak-informati ekunde». [En línea]. Disponible en: <http://www.cs.uu.nl/education/vak.php?vak=INFOMAS>. [Accedido: 08-jul-2013].
- [15] Vlassis, N. (2007) «A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence», *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 1, no. 1, pp. 1-71, ene. 2007.
- [16] Amaya, G. (2010). *Ambiente Multi-agente Robótico para la Navegación Colaborativa en Escenarios Estructurados*. Medellín.
- [17] Sánchez, J. M. (2016). *ABC Tecnología: Yumi: la evolución de los robots colaborativos*. Madrid: Diario ABS, SL. <http://www.abc.es>.
- [18] Salinas, R. (2011). *Usach: Investigadores de la Usach dan vida a una comunidad de Robots*. Santiago de Chile. <http://www.usach.cl>.
- [19] Sutter, J.D. (2011). *Expansión: Los Robots de ayuda en desastres, una consecuencia del 11-S*. México. Expansión. <http://www.expansion.mx>.
- [20] Ovalle, D., Jiménez, J., Acosta, G., (2006). *Arquitectura de ambiente multi-agente Robótico para la navegación colaborativa*. Medellín.
- [21] Rodríguez, G., (2015). *Robótica móvil colaborativo*. Universidad de la Laguna, La laguna.
- [22] Il. Bambino (2008). *Una introducción a los Robots Móviles*. Recuperado http://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.pdf
- [23] Suarez E., Sánchez A. (2015). *Plataforma móvil omnidireccional de cuatro llantas suecas (Mecanum) en configuración "AB"* (tesis de pregrado). Universidad Nacional Autónoma de México.
- [24] Emaze. (2017). *Dron hibrido*. <https://www.emaze.com/@AOZLRFIW/dron1>
- [25] Kuka. (2017). *Plataforma omnidireccional*. <http://www.directindustry.es/prod/kuka-roboter-gmbh/product-17587-1714586.html>.
- [26] Medina J., Martínez R., Jaramillo F., Álava A. (2016). *Robot Omnidireccional Omnibot*. <https://www.youtube.com/watch?v=ZBvbgq1CAWs>.
- [27] (2009). *Robots Omnidireccionales*. <https://www.fing.edu.uy/inco/grupos/mina/pGrado/easyrobots/>.
- [28] Cristina T. (2010). *Robots Híbridos*. <http://cristiantabu.blogspot.com.co/2010/09/>.
- [29] Arduino. (2017). <http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files>
- [30] Método V. (2017). https://es.wikipedia.org/wiki/M%C3%A9todo_en_V

[31] Evan, A (2012). Robotic Airplane, Boat, and Submarine Team Up to Monitor Coral Reefs. *IEEE SPECTRUM*. Recuperado de <https://spectrum.ieee.org/automaton/transportation/marine/iros-2012-robotic-airplane-boat-and-submarine-team-up-to-monitor-coral-reefs>.

[32] Cioarca, R., Nalatan, I., Tura-Bob, S., Micea, M., Cretu, V., Biriescu, M. y Groza, V., (2008). Emergent Exploration and Resource Gathering in Collaborative Robotic Environments. Rose 2008 – IEEE International Workshop on Robotic and Sensors Environments.

Anexos

Anexo 1: Plano de robots y piezas.

Anexo 2: Programa en C de los Robots Azul y Rojo

```
#include <AFMotor.h>
#include <Servo.h>
#define ABRIR 1
#define CERRAR 2
// Definiciones del programa de los motores
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);
Servo servo1;
Servo servo2;

// variables de los servos
int servo_position = 0;
int servo_position1 = 0;
// Variables del puerto analogo
int analogPin2 = 2;
int analogPin3 = 3;
int analogPin4 = 4;
int analogPin5 = 5;
int Pul1 = 0;
int Pul2 = 0;
int Pul3Con = 0;
int Pul4Con = 0;

// variables de la transmision por bluethoo
char rxChar; // Variable para recibir datos del
puerto serie
int ledpin = 13; // Pin donde se encuentra
conectado el led (pin 13)

// Variable de acoplamiento
int Acople = 0;

// Funciones
void mov4motores(int x,int y);
void recibir();
void pulsadores();
void modoAcoplamiento();
void controlServos(int x);

void setup() {

// motores servo
servo1.attach(9);
servo2.attach(10);
controlServos(ABRIR);
// Comunicacion por Bluethoo
pinMode(ledpin, OUTPUT); // Pin 13 como salida

Serial.begin(9600); // Comunicación serie a
9600 baudios
}

void loop() {

//Control de los pulsadores
//pulsadores();

recibir();
//controlServos(ABRIR);
if ( rxChar == '1'){
for( int i=0 ; i<5 ; i++){
mov4motores(1,200);
delay(100);
//Serial.println("ON");
}
}
else if ( rxChar == '2'){
for( int i=0 ; i<5 ; i++){
mov4motores(2,200);
delay(100);
//Serial.println("OFF");
}
}
else if ( rxChar == '3'){
for( int i=0 ; i<5 ; i++){
mov4motores(3,200);
delay(100);
// Serial.println("OFF");
}
}
else if ( rxChar == '4'){
for( int i=0 ; i<5 ; i++){
mov4motores(4,200);
delay(100);
//Serial.println("OFF");
}
}
else if ( rxChar == '5'){
for( int i=0 ; i<5 ; i++){
mov4motores(5,200);
delay(100);
// Serial.println("OFF");
}
}
else if ( rxChar == '6'){ // Giro Derecho
for( int i=0 ; i<1 ; i++){
mov4motores(6,100);
delay(100);
}
}
}
}
```

```

        // Serial.println("OFF");
    }
}
else if ( rxChar == '7'){
    for( int i=0 ; i<5 ; i++){
        mov4motores(7,200);
        delay(100);
        //Serial.println("OFF");
    }
}
else if ( rxChar == '8'){
    for( int i=0 ; i<5 ; i++){
        mov4motores(8,200);
        delay(100);
        //Serial.println("OFF");
    }
}
else if ( rxChar == '9'){
    for( int i=0 ; i<5 ; i++){
        mov4motores(9,200);
        delay(100);
        // Serial.println("OFF");
    }
}
else if ( rxChar == 'A'){
    for( int i=0 ; i<1 ; i++){
        mov4motores(10,100);
        delay(100);
        // Serial.println("OFF");
    }
}
else if ( rxChar == 'B'){
    for( int i=0 ; i<5 ; i++){
        mov4motores(11,200);
        delay(100);
        // Serial.println("OFF");
    }
}
else if ( rxChar == 'C'){
    for( int i=0 ; i<5 ; i++){
        mov4motores(12,200);
        delay(100);
        // Serial.println("OFF");
    }
}
else if ( rxChar == 'D'){
    for( int i=0 ; i<5 ; i++){
        mov4motores(13,200);
        delay(100);
    }
}

// // Robot Rojo
// else if ( rxChar == 'a' && Acople == 0 ) {
//     controlServos(ABRIR);
//     delay(100);
//     while ( Acople == 0 ){
//         modoAcoplamiento();
//         delay(100);
//     }
// }
// // Robot Azul
// else if ( rxChar == 'a' && Acople == 0 ) {
//     controlServos(ABRIR);
//     delay(100);
//     for( int i=0 ; i<5 ; i++){
//         mov4motores(11,200);
//         delay(100);
//         // Serial.println("OFF");
//     }
// }
// else {
//     mov4motores(11,0);
//     delay(100);
// }

}

void recibir(){

    // Si hay datos disponibles en el buffer
    if( Serial.available() )
    {
        // Leer un byte y colocarlo en variable
        rxChar = Serial.read();

        // // Procesar comando de un solo byte
        // if( rxChar == 'A' )
        // {
        //     digitalWrite(ledpin, HIGH);
        //     Serial.println("ON");
        // }
        // else if ( rxChar == 'a' )
        // {
        //     digitalWrite(ledpin, LOW);
        //     Serial.println("OFF");
        // }

        // Podemos hacer otras cosas aquí
        delay(100);
    }
}

```

```

}

void mov4motores(int x,int y){

    int                                     atras=1,
    adelante=2,der=3,dder=4,dider=5,giroder=6,izq=
    7,dizq=8,diizq=9,giroizq=10,quieto=11;
    // son iguales atra-adelante-izquierda-detecha
    int CgDer = 12, Cglzq = 13;
    int speed = 0;
    speed = y;

    motor1.setSpeed(speed); // 0-255 RPM
    motor2.setSpeed(speed);
    motor3.setSpeed(speed);
    motor4.setSpeed(speed);

    if(x==atras) {
        motor1.run(RELEASE);
        motor2.run(FORWARD);
        motor3.run(RELEASE);
        motor4.run(BACKWARD);
    }
    else if(x==adelante) {
        motor1.run(RELEASE);
        motor2.run(BACKWARD);
        motor3.run(RELEASE);
        motor4.run(FORWARD);
    }
    else if(x==izq) {
        motor1.run(FORWARD);
        motor2.run(RELEASE);
        motor3.run(BACKWARD);
        motor4.run(RELEASE);
    }
    else if(x==der) {
        motor1.run(BACKWARD);
        motor2.run(RELEASE);
        motor3.run(FORWARD);
        motor4.run(RELEASE);
    }
    else if(x==dder) {
        motor1.run(BACKWARD);
        motor2.run(BACKWARD);
        motor3.run(FORWARD);
        motor4.run(FORWARD);
    }
    else if(x==dizq) {
        motor1.run(FORWARD);
        motor2.run(BACKWARD);
        motor3.run(BACKWARD);
        motor4.run(FORWARD);
    }
    else if(x==diizq) {
        motor1.run(FORWARD);
        motor2.run(FORWARD);
        motor3.run(BACKWARD);
        motor4.run(BACKWARD);
    }
    else if(x==dider) {
        motor1.run(BACKWARD);
        motor2.run(FORWARD);
        motor3.run(FORWARD);
        motor4.run(BACKWARD);
    }
    else if(x==giroder) {
        motor1.run(FORWARD);
        motor2.run(FORWARD);
        motor3.run(FORWARD);
        motor4.run(FORWARD);
    }
    else if(x==giroizq){ // 1
        motor1.run(BACKWARD);
        motor2.run(BACKWARD);
        motor3.run(BACKWARD);
        motor4.run(BACKWARD);
    }
    // Robot Azul
    else if( x == CgDer ) {
        motor1.run(BACKWARD);
        motor2.run(RELEASE);
        motor3.run(FORWARD);
        motor4.run(RELEASE);
    }
    // // Robot Rojo
    // else if( x == CgDer ) {
    //     motor1.run(FORWARD);
    //     motor2.run(RELEASE);
    //     motor3.run(BACKWARD);
    //     motor4.run(RELEASE);
    // }
    // Robot Azul
    else if( x == Cglzq ) {
        motor1.run(FORWARD);
        motor2.run(RELEASE);
        motor3.run(BACKWARD);
        motor4.run(RELEASE);
    }
    // Robot Rojo
    // else if( x == Cglzq ) {
    //     motor1.run(BACKWARD);
    //     motor2.run(RELEASE);
}

```

```

//  motor3.run(FORWARD);
//  motor4.run(RELEASE);
//  }
}

void pulsadores(){

  Pul1 = analogRead(analogPin5);
  Serial.println(Pul1);
  Pul2 = analogRead(analogPin4);
  Serial.println(Pul2);
  Pul3Con = analogRead(analogPin3);
  Serial.println(Pul3Con);
  Pul4Con = analogRead(analogPin2);
  Serial.println(Pul4Con);
}

void modoAcoplamiento(){

  pulsadores();
  if ( Pul1 > 500 ){
    for( int i=0 ; i<5 ; i++){
      mov4motores(3,200);
      delay(100);
      Serial.println("Derecha");
    }
  }
  else if ( Pul2 > 500){
    for( int i=0 ; i<5 ; i++){
      mov4motores(7,200);
      delay(100);
      Serial.println("izquierda ");
    }
  }
  else if ( Pul3Con > 500){
    for( int i=0 ; i<5 ; i++){
      mov4motores(11,200);
      delay(100);
      Serial.println(" Quieto cerrar pinzas ");
      controlServos(CERRAR);
      Acople = 1;
    }
  }
  else if ( Pul4Con > 500){
    for( int i=0 ; i<5 ; i++){
      mov4motores(11,200);
      delay(100);
      Serial.println(" Quieto cerrar pinzas ");
      controlServos(CERRAR);
      Acople = 1;
    }
  }
}

else {
  mov4motores(2,200);
  delay(100);
  Serial.println(" Seguir adelante ");
}
}

void controlServos(int x){

  // control de los servos      // Inicio 1-50 2-120
  Acoplamiento 1-90 2-80
  if (x == 1){
    servo1.write (50);
    delay (10);
    servo2.write (120);
    delay (10);
  }
  else if ( x == 2 ){
    servo1.write (88);
    delay (10);
    servo2.write (75);
    delay (10);
  }

  // for (servo_position = 0; servo_position <=180;
  servo_position +=1){
  //      servo1.write (servo_position);
  // G-> AQUI QUEDA DESTERMINADO LA POSICION
  // DE LOS SERVOS CON LA FUNCION
  // delay (10);
  // }
  //
  // for (servo_position = 180; servo_position >=0;
  servo_position -=1){
  // servo1.write (servo_position);
  // delay (10);
  // }
  // for (servo_position1 = 0; servo_position1
  <=180; servo_position1 +=1){
  // servo2.write (servo_position1);
  // delay (10);
  // }
  //
  // for (servo_position1 = 180; servo_position1
  >=0; servo_position1 -=1){
  // servo2.write (servo_position1);
  // delay (10);
  // }
}

```

```
// sentido motores DC  
//FORWARD=ADELANTE  
//BACKWARD=ATRAS  
//RELEASE=DETENIDO
```