



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA SEMI-AUTÓNOMO DE ROBOTS
MÓVILES COLABORATIVOS PARA EL
MANEJO DE MATERIALES**

T E S I S
Que para obtener el título de
INGENIERO MECATRÓNICO

P R E S E N T A N

**ALFREDO MIGUEL ÁNGELES GARCÍA
DANIEL LIMA ROBLEDA**

**DIRECTOR DE TESIS:
DR. VÍCTOR JAVIER GONZÁLEZ VILLELA**



MÉXICO D.F.

ENERO 2011

Dedicatoria.

A mis papás por regalarme su infinito amor y sabiduría, compartir mi vida e incontable felicidad.

A mi hermana Lau por la grandiosa vida que hemos pasado juntos y a mi nuevo hermano César por su enorme calidez y bondad.

A mis ti@s y prim@s por su imprescindible apoyo y seguimiento de mis pasos con mucho afecto.

A Eva por llenar de inmenso amor mi corazón.

A todos mis amigos por duplicar mi felicidad y dividir las angustias por la mitad.

A mi amigo Daniel Lima por acompañarme con inteligencia y paciencia en esta grandiosa etapa.

*Alfredo Ángeles
Enero 2011*

Dedicatoria.

*A mis padres **Mónica** y **Gabino** por su apoyo, su ayuda incondicional en todo momento, porque son mi fuente de inspiración, por su esfuerzo derramado día a día y por aguantarme durante todos estos años.*

*A mi hermano **Gabriel** porque siempre ha estado conmigo en las buenas y en las malas.*

*A mi amigo **Alfredo** porque sin él gran parte de este trabajo no se habría podido realizar.*

Y a todos aquellos que han sido parte de mi vida universitaria.

*Daniel Lima Robleda.
Enero 2011.*

Agradecimientos.

A la Universidad Nacional Autónoma de México por la excelente educación recibida.

A los profesores de la Facultad de Ingeniería, en especial a los del Departamento de Ingeniería Mecatrónica, a quienes debemos nuestra formación profesional.

Al Dr. Víctor Javier González Villela por la motivación y confianza brindada durante la realización de este proyecto.

A la DGAPA con respecto al proyecto PAPIIT IN108308 con nombre "Investigación y desarrollo en Robótica Móvil, Robótica Paralela y Sistemas Mecatrónicos" por el apoyo brindado para la elaboración de este trabajo.

Resumen

En esta tesis se presenta el diseño de un sistema semiautónomo de robots móviles colaborativos (collaborative robot, cobot) para el manejo de materiales, que mediante un software de visión llamado ReactIVision se obtienen valores de posición y ángulo en tiempo real. Con esos valores, se adquieren las velocidades angulares apropiadas para cada llanta gracias a la aplicación de algoritmos de planeación basados en campos potenciales artificiales desarrollado en C#. Finalmente, las velocidades se envían a cada cobot a través del protocolo de comunicación inalámbrica Zigbee y se procesan por un microcontrolador PIC16F887 que por medio de comunicación serial I²C le envía a la tarjeta de control de los actuadores MD-25 los datos adecuados para la ejecución de la tarea.

En este trabajo se presentan tres comportamientos distintos a la solución del problema de ordenar y transportar cuatro objetos colocados aleatoriamente en cuatro estantes fijos utilizando dos cobots montacargas.

ÍNDICE GENERAL

CAPÍTULO I. INTRODUCCIÓN	1
CAPÍTULO II. ROBOTS MÓVILES COLABORATIVOS	6
2.1 Introducción a los robots móviles colaborativos.....	6
2.2 Conceptos básicos.....	6
2.2.1 <i>Definiciones</i>	6
2.2.2 <i>Agentes Inteligentes</i>	7
2.2.2.1 Coordinación entre agentes.....	8
2.3 Arquitecturas de sistemas multi-robots	8
2.3.1 <i>Centralizadas o Descentralizadas</i>	8
2.3.2 <i>Diferenciación</i>	9
2.3.3 <i>Comunicación</i>	9
2.4 Nuestros robots móviles.	9
2.4.1. <i>Prototipo</i>	9
2.4.2. <i>Tarjeta MD-25</i>	11
CAPÍTULO III. VISIÓN, COMUNICACIÓN E INTERFAZ	12
3.1 Visión.....	12
3.1.1 <i>ReactIVision</i>	12
3.1.1.1 Arquitectura.....	13
3.1.1.2 Componente de Reconocimiento.....	13
3.1.1.3 Componente de Comunicación	14
3.1.1.4 Componente Cliente.....	14
3.1.1.5 Símbolos Fiducial	14
3.1.1.6 Fiducial Amiba	14
3.1.2 <i>Cámaras y lentes</i>	16
3.1.3 <i>Interpretación de la Información</i>	16
3.2 Comunicación inalámbrica.....	17
3.2.1 <i>Introducción</i>	17
3.2.2 <i>Protocolo Zigbee</i>	18
3.2.2.1 Arquitectura.....	18
3.2.2.2 Modos de operación.....	19

3.3 Interfaz	20
CAPÍTULO IV. PLANIFICACIÓN DE TRAYECTORIAS Y TAREAS	24
4.1 Localización de un robot en el plano.	24
4.2 Tipos de Arquitecturas.	25
4.2.1 <i>Diseño funcional de la arquitectura</i>	26
4.3 Campos Potenciales.	26
4.3.1 <i>Modelo matemático de campos potenciales.</i>	28
4.3.2 <i>Limitaciones de la navegación mediante campos potenciales.</i>	29
4.4 Planificación de Caminos.	29
4.5 Planificación de la tarea.	30
CAPÍTULO V. MODELO DE CONTROL PARA EL ROBOT MÓVIL	31
5.1 Modelo de robots móviles.	31
5.1.1 <i>Modelo cinemático de robots móviles.</i>	32
5.1.2 <i>Modelo cinemático para el control del robot móvil diferencial.</i>	35
5.2 Campos Potenciales en el Modelo de Control.	37
CAPÍTULO VI. RESULTADOS	38
6.1 Caso de estudio.	38
6.2 Metodología.	38
6.3 Entorno de las pruebas.	39
6.4 Pruebas	40
6.4.1 <i>Prueba 1. Transporte de cajas.</i>	40
6.4.2 <i>Prueba 2. Cruce de trayectorias.</i>	48
6.4.3 <i>Prueba 3. Jerarquía.</i>	56
6.5 Análisis de Resultados.	64
CAPÍTULO VII. CONCLUSIONES Y TRABAJO FUTURO	66
7.1 Conclusiones	66
7.2 Trabajo futuro	67
APÉNDICES	68
Apéndice A – Código programa C#	68
Apéndice B – Código grabado en los PIC-16F887	93
Apéndice C – Datos Prueba 1.	97
Apéndice D – Datos Prueba 2	102
Apéndice E – Datos Prueba 3.	107

Apéndice F – Xbee.....	112
Apéndice G – ReactIVision	121
Apéndice H – Esquemas electrónicos tarjeta MD25	126
REFERENCIAS	128

LISTA DE FIGURAS

Capítulo 2:

<i>Figura 2. 1: Robot móvil elaborado.....</i>	<i>10</i>
<i>Figura 2. 2: Circuito montado en el cobot.....</i>	<i>10</i>
<i>Figura 2. 3: Tarjeta MD25.....</i>	<i>11</i>

Capítulo 3:

<i>Figura 3. 1: Diagrama de ReactIVision.....</i>	<i>13</i>
<i>Figura 3. 2: Fiducial amiba.....</i>	<i>15</i>
<i>Figura 3. 3: (a)fiducial de reactIVision, (b)nodos hoja blancos y negros y su centroide promedio, (c)nodos hoja negros y su centroide promedio, (d)vector usado para calcular la orientación del fiducial.....</i>	<i>15</i>
<i>Figura 3. 4: Pantalla de captura de 720x480.....</i>	<i>17</i>
<i>Figura 3. 5: Estándares de Familias inalámbricas.....</i>	<i>17</i>
<i>Figura 3. 6: (a)Arquitectura Zigbee. (b)Aplicación.....</i>	<i>19</i>
<i>Figura 3. 7: (a)Módulo Coordinador. (b)Módulo Router.....</i>	<i>20</i>
<i>Figura 3. 8: Posiciones y campos antes del inicio del programa.....</i>	<i>21</i>
<i>Figura 3. 9: Control de velocidad y Distancias.....</i>	<i>21</i>
<i>Figura 3. 10: Comunicación PC - Cobot.....</i>	<i>22</i>
<i>Figura 3. 11: (a)Comienzo del programa. (b)Programa en curso.....</i>	<i>22</i>
<i>Figura 3. 12: Interfaz para el manejo de los cobots.....</i>	<i>23</i>

Capítulo 4:

<i>Figura 4. 1: Campo atractivo correspondiente.....</i>	<i>27</i>
<i>Figura 4. 2: Campo repulsivo correspondiente a.....</i>	<i>27</i>
<i>Figura 4. 3: Campos potenciales artificiales.....</i>	<i>28</i>

Capítulo 5:

<i>Figura 5. 1: Cambio del sistema de referencia en navegación de robots móviles.....</i>	<i>32</i>
<i>Figura 5. 2: Circulo osculador.....</i>	<i>32</i>
<i>Figura 5. 3: Locomoción con guiado diferencial.....</i>	<i>35</i>

Capítulo 6:

<i>Figura 6. 1: Ciclo del sistema empleado.....</i>	<i>38</i>
<i>Figura 6. 3: Área de trabajo.....</i>	<i>39</i>
<i>Figura 6. 2: Fiducials para los cobots y las cajas.....</i>	<i>39</i>
<i>Figura 6. 4: Posiciones iniciales de los elementos que conforman la prueba.....</i>	<i>41</i>
<i>Figura 6. 5: Trayectoria completa del Cobot 1.....</i>	<i>42</i>
<i>Figura 6. 6: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 3.....</i>	<i>43</i>
<i>Figura 6. 7: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 4.....</i>	<i>43</i>
<i>Figura 6. 8: Gráfica de la velocidad lineal del Cobot 1.....</i>	<i>44</i>
<i>Figura 6. 9: Gráfica de la velocidad angular del Cobot 1.....</i>	<i>44</i>
<i>Figura 6. 10: Trayectoria completa del Cobot 2.....</i>	<i>45</i>

<i>Figura 6. 11: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 1.</i>	46
<i>Figura 6. 12: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 2.</i>	46
<i>Figura 6. 13: Gráfica de la velocidad lineal del Cobot 2.</i>	47
<i>Figura 6. 14: Gráfica de la velocidad angular del Cobot 2.</i>	47
<i>Figura 6. 15: Trayectoria de los Cobots.</i>	48
<i>Figura 6. 16: Posiciones iniciales de los elementos que conforman la prueba.</i>	49
<i>Figura 6. 17: Trayectoria completa del Cobot 1.</i>	50
<i>Figura 6. 18: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 2.</i>	51
<i>Figura 6. 19: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 3.</i>	51
<i>Figura 6. 20: Gráfica de la velocidad lineal del Cobot 1.</i>	52
<i>Figura 6. 21: Gráfica de la velocidad angular del Cobot 1.</i>	52
<i>Figura 6. 22: Trayectoria completa Cobot 2.</i>	53
<i>Figura 6. 23: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 4.</i>	54
<i>Figura 6. 24: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 1.</i>	54
<i>Figura 6. 25: Gráfica de la velocidad lineal del Cobot 2.</i>	55
<i>Figura 6. 26: Gráfica de la velocidad angular del Cobot 2.</i>	55
<i>Figura 6. 27: Trayectorias de los Cobots.</i>	56
<i>Figura 6. 28: Posiciones iniciales de los componentes que conforman la prueba.</i>	57
<i>Figura 6. 29: Trayectoria completa seguida por el Cobot 1.</i>	58
<i>Figura 6. 30: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 4.</i>	59
<i>Figura 6. 31: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 3.</i>	59
<i>Figura 6. 32: Gráfica de la velocidad lineal del Cobot 1.</i>	60
<i>Figura 6. 33: Gráfica de la velocidad angular del Cobot 1.</i>	60
<i>Figura 6. 34: Trayectoria completa del Cobot 2.</i>	61
<i>Figura 6. 35: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 2.</i>	62
<i>Figura 6. 36: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 1.</i>	62
<i>Figura 6. 37: Gráfica de la velocidad lineal del Cobot 2.</i>	63
<i>Figura 6. 38: Gráfica de la velocidad angular del Cobot 2.</i>	63
<i>Figura 6. 39: Trayectoria completa de los Cobots.</i>	64

CAPÍTULO I. INTRODUCCIÓN

La palabra *robot* tiene su origen de la palabra *robota* que significa "trabajo", "tarea" o "servidumbre" en checo y en muchos idiomas eslavos. Fue introducida por el dramaturgo Karel Čapek en 1921 en su obra teatral R.U.R (*Rossum's Universal Robots*) donde hace referencia a "seres trabajadores" creados para realizar ciertas labores y capaces de tener sentimientos como los de un humano. Actualmente la palabra robot se usa para referirse a algún dispositivo que opera automáticamente dotado de mecanismos programables que emulan comportamientos para realizar tareas humanas o animales como motivo principal. Dependiendo del dominio de la aplicación, cada robot está dotado de ciertas características mecánicas, eléctricas y de cómputo, es decir, mecatrónicas que le permiten mostrar comportamientos específicos, por ejemplo: robots manipuladores para labores en la industria, robots antropomorfos (humanoides) usados en labores domésticas y de limpieza o robots móviles utilizados en diversos dominios como la exploración de ambientes peligrosos, la vigilancia y control de áreas extensas o la recolección de sustancias nocivas entre otros.

Debido a las capacidades aumentadas de los robots "unitarios" y los avances en la Inteligencia Artificial (IA) se agruparon robots para estudiar capacidades colectivas desde inicios de los noventa. La robótica cooperativa estudia las ventajas que ha presentado tener equipos de múltiples robots como sensado y acción distribuidas, reconfigurabilidad basada en tareas, diseño simplificado de los robots individuales y un sistema confiable gracias a la redundancia de la que se dispone con los robots cooperativos.

La robótica cooperativa surge como una idea de sentido común para mejorar el desempeño en la realización de la tarea a ejecutar. Hacer un trabajo entre varios puede tener muchas más ventajas como por ejemplo, reducir el tiempo de ejecución de la tarea o realizar tareas más complejas, que hacerlo uno sólo. Basta con darse cuenta de que hay algunas tareas que simplemente no pueden ser realizadas únicamente por un individuo debido a que sobrepasa sus límites, como desplazar eficientemente cajas pesadas o manipular objetos de grandes dimensiones, es decir, el manejo de materiales. Actividad que considera el almacenamiento, transporte y entrega de materiales, pudiendo ser estos desde el tamaño de chips de silicio hasta camiones para transporte de diesel. Hay una idea no menos importante que la mejora de rendimiento, y es que si alguien del grupo falla, el resto puede sustituirle y terminar el trabajo con éxito. En la mayoría, el grado de dificultad de la tarea determina la complejidad en el diseño del robot. Por lo tanto, en aquellas que son muy complicadas, se tienen robots estructuralmente complejos, costosos y poco versátiles.

Ante esta problemática, durante más de dos décadas, ha surgido un creciente interés por el estudio de sistemas que integran varios robots conocidos con el nombre de *Sistemas Multi-Robot (SMR)* o *equipo de robots* y que en algunas tareas han demostrado ventajas

que los hacen más eficientes respecto a sistemas uni-robot. Los SMR ofrecen interesantes alternativas en la concepción de sistemas robóticos y pueden darnos muchos servicios útiles. Sin embargo, el diseño y la implementación de un equipo de robots enfrentan problemas concernientes a los recursos de cómputo, especificación de tareas, comunicación entre robots, localización, control y planificación entre muchos otros.

Estos no son problemas sencillos de resolver y hay distintas maneras de enfocarlos. Por lo tanto, durante este tiempo, la investigación en estas áreas ha progresado ampliamente. Por ejemplo, para empezar basta con definir si la arquitectura de un grupo será centralizada o descentralizada, y en caso de ser descentralizada, si el sistema será jerárquico o distribuido. Dicha selección dependerá del problema planteado y será una decisión fundamental para su correcta solución. De acuerdo al comportamiento cooperativo, se pueden nombrar trabajos de investigación estudiados desde hace diez años. Por ejemplo Theraulaz en [1] plantea una estrategia de control cooperativo para un SMR que emula la búsqueda de alimento (*foraging*) a partir del estudio de colonias de avispas. Stell en [2] presenta una simulación que permite generar comportamientos emergentes a través de una aplicación dirigida al problema de recolección de muestras de rocas en exploración interplanetaria. Mataric en [3] presenta los resultados de una implementación de un grupo de comportamientos a los que denomina *dispersión, aggregation, flocking* en un grupo de robots físicos. En [4] se presenta otra arquitectura jerárquica implementada en robots físicos cuyos comportamientos básicos son pesados o ponderados a través de coeficientes y sumados con el fin de obtener comportamientos compuestos. En [5] se presenta un conjunto de robots miniatura denominados MARS (*Micro Autonomus Robotic System*) que implementan algunas ideas de control basado en arquitectura CEBOT.

Recientemente en el año 2004 Sugawara, Kazama y Watanabe, proponen una implementación virtual de la feromona, a través de gráficas proyectadas en el suelo [6]. En 2007 Purnamadjaja, Iskandar y Russell se realiza una simulación de robots que utilizan feromonas para comunicarse pero limitada a una implementación real [7]. Finalmente existen SMR muy conocidos como son:

El CEBOT, descrita en [8] y en otros trabajos relacionados, presenta una propuesta de arquitectura jerárquica inspirada en la organización celular, en la cual se acoplan los robots unos con otros y se reconfiguran según las necesidades cambiantes del entorno. Se basada en células (robots o agentes) maestras que controlan y coordinan subtareas, además se comunican con otras células maestras. En [9], Asama *et al.* describen su sistema descentralizado de robots que permite la comunicación, asignación de tareas y planeación de movimientos entre agentes heterogéneos llamado ACTRESS (*ACTor-based Robot and Equipments Synthetic System*), se pretende que hagan una tarea en grupo, siendo imposible hacerla individualmente. No hay ningún robot que mande sobre el resto y deben distribuirse las tareas entre ellos de una manera óptima. Derivado del CEBOT se encontró una arquitectura descentralizada que usualmente utiliza robots homogéneos conocida como SWARM, [10] y [11], que busca generar una “inteligencia colectiva” o *swarm intelligence* que les permite solucionar problemas a partir de comportamientos

elementales en cada uno de los robots y que reaccionan a estímulos de aquellos que los rodean utilizando comunicación implícita, a diferencia del CEBOT no se acoplan. Otro trabajo relacionado es sobre ALLIANCE [12] para estudiar la cooperación en un grupo de robots heterogéneos débilmente acoplados. Cada robot implementa un mecanismo que permite que cada uno sienta en mayor o menor grado los efectos de sus propias acciones y la de los otros robots a través de percepción y comunicación explícita. El sistema consta de varios robots muy independientes y con una IA variable según el caso (desde una IA simple a una bastante potente). De este modo, su comportamiento variará según reconozca sus acciones y las del resto. En la arquitectura L-ALLIANCE se les permite incluso aprender de las acciones y modificar los parámetros mediante los cuales se activarán sus distintos comportamientos.

También en [13] se plantea la arquitectura AuRA la cual combina la planeación de una arquitectura deliberativa con la rápida respuesta de una arquitectura reactiva. Finalmente encontramos en [14] y [15] a GOFER, que estudia la solución de problemas distribuidos usando técnicas de Inteligencia Artificial (IA) tradicional, el sistema consta de varios robots (3 más o menos) con una IA básica donde un planeador y agendador central (CTPS, *Central Task Planning Scheduling*) tiene una vista global de la ejecución de tareas de los robots y su disponibilidad para ejecutarlas. De esta manera una vez la unidad central ha distribuido las tareas, cada robot cuenta con una IA para resolverlas.

Las características de las arquitecturas serán abordadas con detalle en el capítulo II.

Particularmente, los robots empleados para el manejo de materiales incluyen diversos tipos de AGV (*Autonomous Guided Vehicles*) [16] diseñados para que se desplacen autónomamente por un entorno, típicamente industrial, siguiendo un determinado circuito.

El circuito se establece empleando distintos medios. El más habitual es el de un cable enterrado que crea un campo magnético que debe ser detectado por un sensor situado en la parte inferior del vehículo. Son los denominados *vehículos filoguiados*.

También se utilizan técnicas basadas en el establecimiento de marcas en la superficie tales como las bandas magnéticas, ópticas, o de partículas con determinadas propiedades químicas, que requieren sensores específicos [17].

Algunas de las técnicas mencionadas anteriormente tienen sus limitaciones ya que han sido enfocadas para ambientes estáticos que asumen que el robot es el único objeto en movimiento. Por ejemplo, para entornos cambiantes y complejos, donde la actualización de un área sea primordial, representa una dificultad enorme ya que gran parte se manifiesta en un considerable consumo de recursos de cómputo y que de alguna forma afecta en los tiempos de respuesta del robot.

Adicionalmente, los robots autónomos necesitan saber su posición respecto al ambiente y a los demás robots con el fin de completar sus tareas individualmente o de manera cooperativa. No obstante, estimar con precisión la posición propia del robot así como el

estado de los objetos en movimiento no es un trabajo sencillo. En la mayoría de los casos debido a que la información que los robots reciben a través de sus sensores es intrínsecamente incierta y el control sobre sus actuadores es también impreciso. Es necesario aclarar que los robots autónomos o *inteligentes* son los más evolucionados desde el punto de vista del procesamiento de información. Son máquinas capaces de percibir, modelar el entorno, planificar y actuar para alcanzar objetivos sin la intervención, o con una intervención muy pequeña, de supervisores humanos. Pueden trabajar en entornos poco estructurados y dinámicos, realizando acciones en respuesta a contingencias variadas en dicho entorno.

Por estas razones, esta tesis propone el diseño e implementación de un sistema semiautónomo (el ambiente estructurado en el que se trabaja es conocido para los cobots y a la vez, dependen de una unidad central que les distribuye las tareas), descentralizado y jerárquico de multi-robots competitivos capaces de trabajar de manera colaborativa basado en la arquitectura GOFER enfocado al problema del manejo de materiales. Éste se apoya en un esquema de localización que teniendo en cuenta la complejidad en el procesamiento de cada cobot (*collaborative robot*), determine su posición y orientación en tiempo real. La división de trabajo se realizará mediante algoritmos de planeación y control. Más importante aún, el sistema evita posibles colisiones gracias a campos potenciales artificiales de tal manera que el transporte de objetos se realice de manera efectiva permitiendo así la ejecución concurrente de tareas, su control descentralizado y además garantizar la escalabilidad.

La tesis se encuentra dividida en 7 capítulos.

I. INTRODUCCIÓN. El capítulo tiene como objetivo dar un comunicado en general del campo de investigación para tener una idea del problema a tratar, enfatizar aspectos ya estudiados por otros investigadores, indicar la mayor necesidad de investigación y mencionar los propósitos de estudio.

II. ROBOTS MÓVILES COLABORATIVOS. En este capítulo se explican conceptos que se utilizan a lo largo de la tesis, se conoce la clasificación de los robots y sus arquitecturas para finalmente describir a nuestro robot desde el material empleado para su construcción y su diseño hasta explicar sus controladores, actuadores y efectores.

III. VISIÓN, COMUNICACIÓN E INTERFAZ. En este capítulo se describe el *software* de visión llamado ReaTIVision para la localización de los cobots, el protocolo de comunicación inalámbrica existente entre éstos basado en Zigbee y la interfaz desarrollada para la interacción de los distintos elementos del sistema en lenguaje de programación C#.

IV. PLANIFICACIÓN DE TRAYECTORIAS Y TAREAS. Este capítulo trata del modelo reactivo para la detección de colisiones basado en campos potenciales artificiales así como del comportamiento realizado en cada cobot

V. MODELO DE CONTROL PARA EL ROBOT MÓVIL. Este capítulo habla del modelo cinemático de robots móviles y se aborda en particular el modelo diferencial. También de la aplicación del modelo en los campos potenciales y en la generación de las trayectorias.

VI. RESULTADOS. Este capítulo habla de los resultados experimentales finales donde se aprecian gráficas de trayectorias seguidas por los cobots en distintas pruebas propuestas.

VII. CONCLUSIONES Y TRABAJO FUTURO. Este capítulo destaca los aspectos más relevantes de la tesis así como el enriquecimiento al campo de los SMR para continuar con esta línea de investigación en la Universidad Nacional Autónoma de México.

CAPÍTULO II. ROBOTS MÓVILES COLABORATIVOS

Es necesario definir los elementos del sistema que conforman el trabajo, en este capítulo abordamos aquellos que poseen los robots móviles así como sus subsistemas.

2.1 Introducción a los robots móviles colaborativos

El desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la Robótica, restringido inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos. Se trata también de incrementar la autonomía limitando todo lo posible la intervención humana.

En los años ochenta el incremento espectacular de la capacidad computacional y el desarrollo de nuevos sensores, mecanismos y sistemas de control, permite aumentar la autonomía. Es decir, que el robot tenga la suficiente inteligencia como para reaccionar y tomar decisiones basándose en observaciones de su entorno, sin suponer que este entorno es perfectamente conocido.

La autonomía de un robot móvil se basa en el sistema de navegación automática. En estos sistemas se incluyen tareas de planificación, percepción y control. Una vez realizada la planificación de la trayectoria (abordada en capítulos posteriores), es necesario planificar movimientos concretos y controlar dichos movimientos para mantener al vehículo en la trayectoria planificada. De esta forma, se plantea el problema del seguimiento de caminos, que para vehículos con ruedas se concreta en determinar el ángulo de dirección teniendo en cuenta la posición y orientación actual del vehículo con respecto a la trayectoria que debe seguir.

2.2 Conceptos básicos

2.2.1 Definiciones

Para comprender más a fondo la nomenclatura de este proyecto y a la vez distinguir a los robots cuya actividad consiste en fabricar piezas o bienes mediante procesos de manufactura de aquéllos cuya tarea no es la de producir estos bienes, se utiliza en la actualidad de manera genérica el término de *robots de servicio*. La Federación Internacional de Robótica, define de “manera provisional” a estos como:

Robots de servicio (IFR): Un robot que opera de manera semi o totalmente autónoma para realizar servicios útiles a los humanos y equipos, excluidas las operaciones de manufactura [...] Los robots de servicio pueden estar equipados o no con un brazo manipulador como los industriales. A menudo, pero no siempre, son robots móviles [...]

De acuerdo a la norma ISO 8373 se define robot móvil como:

Robot móvil (ISO): Robot que contiene todo lo necesario para su pilotaje y movimiento (potencia, control y sistema de navegación).

Por otra parte, en ocasiones se hace referencia a los robots móviles como AGV, siglas que proceden del término *Autonomous Guided Vehicle* o vehículo guiado autónomamente. La definición es insuficiente para abarcar la realidad, pues dentro de los robots móviles se pueden considerar los robots móviles terrestres (denominados UGV *Unmanned Ground Vehicles*) los robots móviles aéreos (conocidos como UAV *Unmanned Aerial Vehicles*) y los robots móviles submarinos (identificados como AUV *Autonomous Underwater Vehicles*), siendo todos ellos en ocasiones, a pesar del nombre, semi-autónomos.

Las dificultades derivadas de la falta de estructuración del entorno y de la imposibilidad de predecir las situaciones en las que los robots de servicio, sean móviles o estacionarios, deben operar, hace que en muchas ocasiones se deba recurrir a modos de funcionamiento semi-autónomos o incluso totalmente dependientes de las órdenes de operador.

Un último concepto derivado de la telerobótica son los denominados Cobots (*Collaborative Robots*). Los Cobots, ideados por Peshkin y Colgate en el año 1996 se definen como:

Cobot: Dispositivo robótico que manipula objetos en colaboración física directa con él.

Los Cobots, ideados para la colaboración estrecha entre humano y robot, se utilizan en tareas donde es preciso guiar al operador por caminos sometidos a restricciones (superficies o restricciones virtuales) o donde éste debe mover cargas notables. El cobot, limita las trayectorias que puede hacer el humano o soporta la carga, mientras éste dirige el movimiento.

2.2.2 Agentes Inteligentes.

Un agente es todo aquello que puede considerarse que percibe su ambiente mediante sensores y que responde o actúa en tal ambiente por medio de efectores. No todos los agentes que tenemos en nuestro alrededor son agentes inteligentes. El problema de determinar si un agente es inteligente o no, puede verse como determinar qué es la inteligencia, lo cual es complicado. En este trabajo se considera que un agente es inteligente si es capaz de tener un comportamiento autónomo flexible para cumplir objetivos que tenga propuestos. Las características del concepto de flexibilidad que buscan son:

Reactivo. Son agentes que perciben su ambiente y reaccionan ante estímulos en el momento adecuado para cumplir sus objetivos.

Pro activos. Son agentes que toman la iniciativa, es decir, no esperan a un estímulo externo para realizar acciones que le lleven a cumplir con sus objetivos.

Sociabilidad. Son agentes capaces de comunicarse con otros agentes para satisfacer sus objetivos, que para este trabajo es de suma importancia.

Características que podemos considerar opcionales y no tocamos en este trabajo son encontradas en varios tipos de agentes, como la movilidad y la necesidad de interacción con usuarios y el consiguiente aprendizaje de su comportamiento.

2.2.2.1 Coordinación entre agentes

Los agentes se comunican para lograr mejores resultados propios o mejores resultados en común. La comunicación permite a los agentes coordinar sus acciones y su comportamiento, generando comportamientos coherentes en sociedad.

La coordinación es una propiedad de un sistema de agentes realizando una actividad en un ambiente compartido. Llamaremos *cooperación* a la coordinación de agentes que no son antagonistas; la *negociación* es la coordinación entre agentes que compiten entre sí o agentes individualistas. Para lograr que dos agentes puedan cooperar correctamente, cada uno de ellos debe tener el modelo del otro agente, para así poder anticipar la acción del otro. La coherencia se refiere a qué tan bien un sistema se comporta como una sola unidad.

Se dice que un grupo de agentes forman un equipo cooperativo cuando:

- Todos los agentes comparten una meta común.
- Cada agente requiere realizar sus propias tareas para lograr la meta común al grupo o a un subgrupo.
- Cada agente acepta una petición para lograr su cometido.

2.3 Arquitecturas de sistemas multi-robots

Como se mencionó en la introducción, la arquitectura de un conjunto de robots se refiere a la infraestructura que permite el comportamiento colaborativo. Incluye aspectos como:

- Centralizado o descentralizado.
- Diferenciación.
- Comunicación.
- Modelado de agentes (capítulo V).

2.3.1 Centralizadas o Descentralizadas

Las arquitecturas centralizadas están caracterizadas por un único agente de control. Las descentralizadas carecen de dicho agente. Existen dos tipos de arquitecturas descentralizadas: arquitecturas distribuidas, en las cuales todos los agentes son iguales en

términos de control poniéndose de acuerdo en la meta global mediante un diálogo, y arquitecturas jerárquicas, que son localmente centralizadas apareciendo la figura de robots prioritarios.

Un SMR puede ser categorizado respecto a la arquitectura en centralizado o descentralizado dependiendo de la estrategia de coordinación aunque esto no implica que un sistema sea totalmente centralizado o descentralizado, existen formas intermedias como lo señala Cao *et. al.* En [18].

2.3.2 Diferenciación

En cuanto a la diferenciación, se divide en homogéneo y heterogéneo. En el primero todos los robots tienen las mismas características y capacidades como por ejemplo, capacidad de procesamiento y cómputo, mecanismos de comunicación, dispositivos sensoriales y de actuación etc. mientras que en el segundo hay robots con diferentes capacidades. Las arquitecturas heterogéneas son más complejas ya que los robots tienen que conocer las capacidades de los demás y se complica la coordinación.

2.3.3 Comunicación

El otro punto es la comunicación, refiriéndose a la manera en que los robots pueden interactuar. Existen tres formas principales:

- Interacción mediante el ambiente.
- Interacción mediante el sensado.
- Interacción mediante comunicación.

En la interacción mediante el ambiente no hay una forma explícita de comunicación, los robots interactúan a través del medio ambiente que actúa como una especie de “memoria compartida”. Mediante el sensado, los robots interactúan a través de sus sensores, identificando a los demás y distinguiéndolos de otros objetos en el ambiente. Para ello se usan diversos sensores como infrarrojos, de ultrasonido, de tacto, etc. Finalmente en la interacción mediante comunicación, existe una comunicación directa entre los robots ya sea mediante mensajes uno a uno (*unicasting*) o a todos (*broadcasting*). Un robot puede usar *broadcasting* para dar a conocer su localización a todo el sistema o bien utilizar *unicasting* para comunicarse exclusivamente con otro robot [19].

2.4 Nuestros robots móviles.

2.4.1. Prototipo

Para este trabajo se desarrollaron dos robots móviles de configuración diferencial, se les acopló, en la parte frontal, un mecanismo para el transporte de cajas tipo montacargas. El

cual se basa en un tornillo sin fin que levanta y desciende el material, así mismo los efectores tipo pinzas pueden ser ajustados a las dimensiones deseadas. El material utilizado en la estructura principal de los prototipos está hecho de aluminio, **Figura 2. 1.**

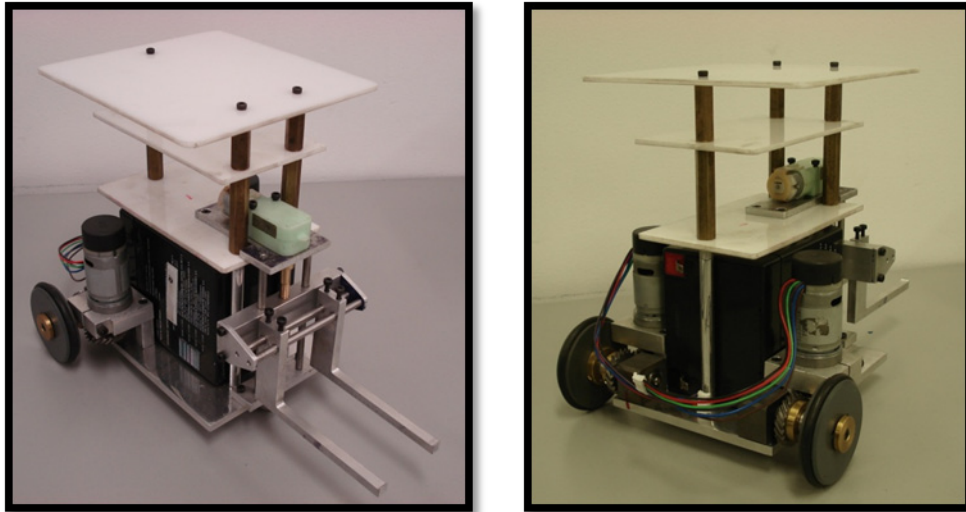


Figura 2. 1: Robot móvil elaborado

Asimismo se diseñó un sistema modular de dos pisos (entrepasos) hechos de plástico para colocar el circuito y fijar la ameba, como se aprecia en la **Figura 2. 2**

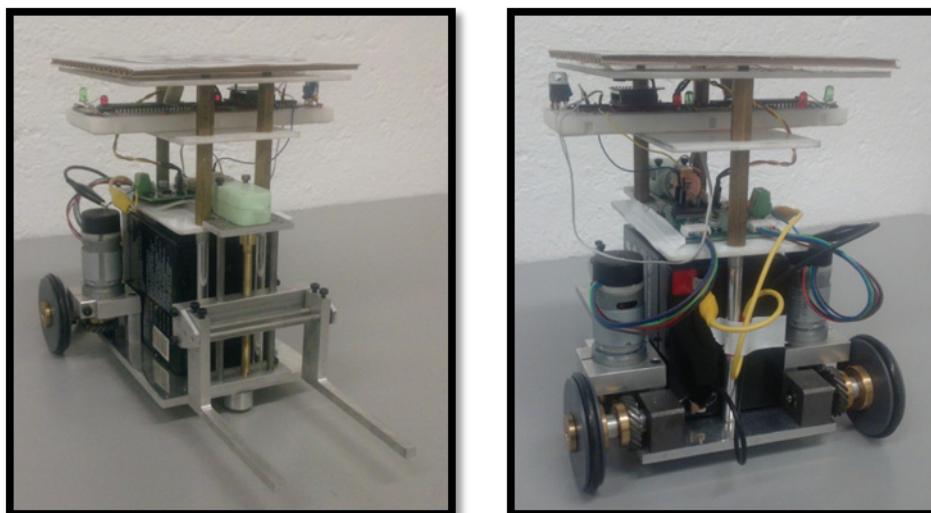


Figura 2. 2: Circuito montado en el cobot

2.4.2. Tarjeta MD-25

Los robots móviles cuentan con dos motores, modelo EMG-30, de corriente directa y que contienen un encoder de cuadratura interno que genera 360 pulsos por vuelta del eje. Para una mejor respuesta a los cambios de velocidades de los motores se utiliza la tarjeta MD-25 del mismo fabricante **Figura 2. 3**. Las características principales de la tarjeta es que con ella se pueden controlar dos motores ya sea de forma independiente, es decir, los motores pueden girar a diferente velocidad y sentido o bien, para que los dos motores funcionen igualmente sólo es necesario indicar el sentido de giro y la velocidad. Otra característica importante es que la tarjeta guarda, en cuatro registros de 8 bits, la suma de todos los pulsos generados por el encoder del motor.

La tarjeta requiere de una fuente de 12 [V] para su funcionamiento y cuenta con una etapa de regulación para que el voltaje en todo el circuito interno de la tarjeta sea de 5 [V]. La tarjeta tiene un microcontrolador PIC-877A y dos etapas de potencia para los motores. Para poder controlar la tarjeta es necesario hacer uso de otro microcontrolador, para ello se eligió el PIC-16F887. La comunicación entre el PIC y la tarjeta es por medio del protocolo I²C.

La tarjeta cuenta con 17 registros de 8 bits cada uno, de los cuales 5 pueden ser leídos y/o escritos y el resto sólo leídos. De los 5 registros, dos son para la velocidad de cada motor en donde el valor 0 corresponde a reversa total, el valor 128 corresponde a paro total y el valor 255 corresponde a la máxima velocidad, un registro es para indicar la aceleración que queremos al cambio de velocidad, otro registro para el modo de operación de la tarjeta y el último registro para la comunicación I²C.

De los registros de solo lectura, 8 corresponden a los encoders de los motores, cuatro para cada motor, en estos cuatro se guarda la suma total de los pulsos generados por el encoder, en otros dos registros nos indican la cantidad de corriente que pasa por cada motor, un registro nos indica el voltaje de alimentación de la tarjeta y el último registro nos indica características de la tarjeta.

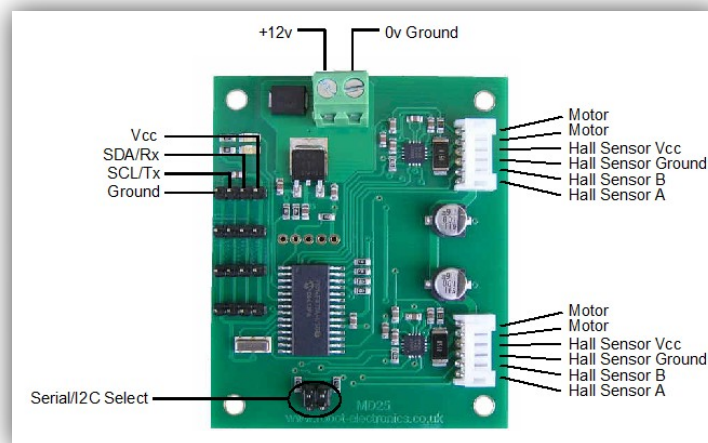


Figura 2. 3: Tarjeta MD25

CAPÍTULO III. VISIÓN, COMUNICACIÓN E INTERFAZ

3.1 Visión.

Se llama visión a la capacidad de interpretar nuestro entorno gracias a los rayos de luz que alcanzan el ojo. La visión o sentido de la vista es una de las principales capacidades sensoriales del hombre, de muchos animales y organismos. Algunos de estos últimos tienen receptores luminosos capaces de reaccionar ante determinados movimientos y sombras, pero la verdadera visión supone la formación de imágenes en el cerebro que permiten la abstracción del entorno y la posibilidad de interacción con el mismo. En el presente trabajo se incorpora visión al sistema por medio de una cámara de video, que es el elemento tecnológico más cercano a la visión que tenemos los seres humanos.

3.1.1 ReactIVision.

Tradicionalmente el reconocimiento y procesamiento de imágenes por computadora se asocia con un proceso difícil que consume una buena cantidad de los recursos del sistema, además de requerir equipos especializados de alto costo. Sin embargo, en la actualidad existen herramientas que facilitan dicha tarea, reduciendo el gasto de recursos computacionales y el costo en equipo de video, una de estas herramientas es la plataforma reactIVision descrita a continuación.

ReactIVision es una plataforma de visión por computadora para la interacción tangible basada en el plano. Su código es abierto y principalmente está diseñada para la construcción de interfaces tangibles bidimensionales con el usuario.

La plataforma fue desarrollada como elemento sensor principal para la reactTable expuesto por [20] como un instrumento musical electroacústico tangible. Utiliza marcadores visuales espacialmente diseñados (símbolos *fiducial*) que pueden ser asociados a objetos físicos. Los marcadores son reconocidos y localizados por un algoritmo de visión por computadora optimizado para el diseño particular de los marcadores, mejorando la velocidad promedio y la robustez del proceso de reconocimiento. Estos símbolos marcadores *fiducial* permiten distinguir miles de identidades únicas, añadiendo la posibilidad del preciso cálculo del ángulo de rotación del marcador en un plano de dos dimensiones.

ReactIVision y sus componentes están estructurados por la combinación de distintas licencias de software libre como son GPL, LGPL, BSD y puede ser obtenido como ejecutable o como código fuente abierto del sitio *SourceForge*.

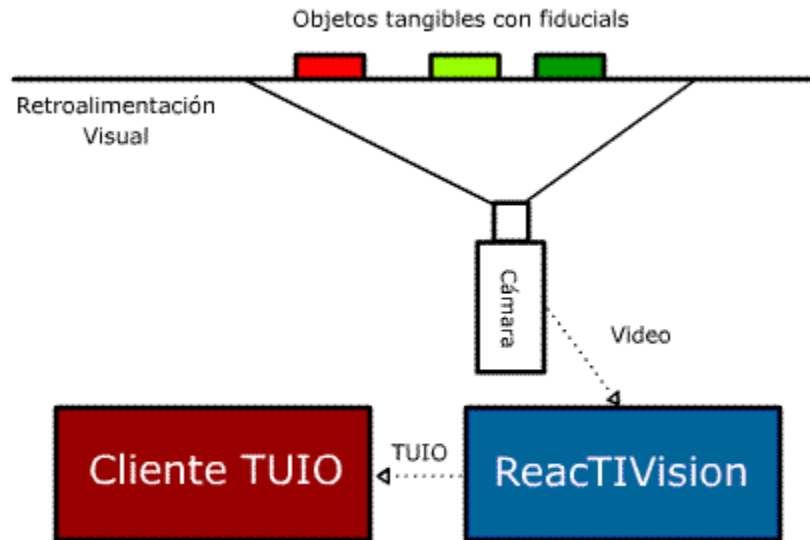


Figura 3. 1: Diagrama de ReactIVision.

3.1.1.1 Arquitectura

ReactIVision está diseñado como una aplicación de construcción fragmentada y no como un objeto de una librería de código. Cada componente del sistema está implementado como un proceso ejecutable independiente. La comunicación entre los componentes se logra utilizando un protocolo publicado (TUIO). Este diseño simplifica el uso del mismo facilitando la integración con distintos ambientes de programación. La arquitectura también permite la ejecución de los componentes de la plataforma en distintas máquinas, lo cual puede ser bastante útil en ciertos contextos de instalación o aplicaciones de red.

3.1.1.2 Componente de Reconocimiento

La aplicación ReactIVision adquiere la imagen de una cámara, busca símbolos *fiducial* en el flujo de video -cuadro por cuadro- y manda los datos sobre los símbolos encontrados a través de un *socket* de red hacia una aplicación receptora. La aplicación está diseñada de forma modular, haciendo fácil la inclusión de nuevas imágenes para el reconocimiento y componentes para el procesamiento.

El código está programado en C++ Portable combinado con otros componentes para la adquisición. Este software también se encuentra de manera independiente como código abierto con el nombre PortVideo.

3.1.1.3 Componente de Comunicación

ReactIVision define su propio protocolo de comunicación TUIO descrito por Kaltenbrunner et al. [21] que fue específicamente diseñado para las necesidades de las interfaces tangibles basadas en una mesa: codificando y transmitiendo los atributos de los artefactos tangibles que se encuentran sobre la superficie de la misma.

Para poder ofrecer una mejor y más confiable comunicación con aplicaciones de clientes locales y remotos las capas del protocolo redundan los mensajes sobre una estructura de transporte UDP. TUIO define un set abierto de instrucciones para el control. Estos mensajes constantemente transmiten la presencia, posición y ángulo de los símbolos reconocidos, así como sus parámetros derivados. En el cliente, estos mensajes redundantes son decodificados a funciones genéricas como agregar símbolo, actualizar símbolo y remover símbolo, cada una correspondiente al evento físico aplicado sobre cada uno de los objetos.

3.1.1.4 Componente Cliente

Para facilitar el desarrollo de interfaces, el cliente de reactIVision puede ser programado en distintos lenguajes como son C++, C#, Java, SuperColloder, Max/MSP y Flash. El cliente se encarga de decodificar los mensajes para trasladarlos a alguna aplicación gráfica o simplemente imprimirlos en la consola.

3.1.1.5 Símbolos Fiducial

La geometría de los símbolos *fiducial* ha sido rediseñada, hasta llegar a la amiba, para ser reconocida por medio de algoritmos genéticos, reduciendo el tamaño de los marcadores y haciendo más eficientes los algoritmos de seguimiento.

Para el reconocimiento de los símbolos *fiducial* la imagen de origen es convertida a blanco y negro con un algoritmo adaptativo de mapeo. Esta imagen es segmentada en regiones gráficas adyacentes de blanco y negro alternante, dentro de estas se buscan tres estructuras únicas contenidas en los símbolos *fiducial*. Finalmente las tres estructuras identificadas son comparadas con un diccionario que le asigna un ID numérico en particular de acuerdo con la semejanza obtenida.

3.1.1.6 Fiducial Amiba

La compacta geometría del *fiducial* amiba (**Figura 3. 2**) fue obtenida a través de algoritmos genéticos. Al utilizar también un algoritmo genético para el reconocimiento se optimiza la búsqueda del *fiducial* utilizando un conjunto de funciones para el reconocimiento de figuras, formas, tamaños, localización de centroides y orientación. ReactIVision funciona con un conjunto de 90 símbolos distintos escogidos de 128 conjuntos de tres patrones. Los patrones se conforman de nodos hoja, la limitación en la cantidad de los mismos es

para favorecer la exclusión en el reconocimiento, favoreciendo la robustez del algoritmo y evitando la detección de falsos positivos debido al ruido.

La posición del símbolo es calculada como el centroide de todos los nodos hoja, (pequeñas manchas circulares **Figura 3. 3(b)**), los cuales proveen mayor precisión en el reconocimiento. La orientación del símbolo (**Figura 3. 3(d)**) es calculada como el vector que va del centroide del marcador (**Figura 3. 3(b)**) hacia el centroide de los nodos hoja oscuros (**Figura 3. 3(c)**) distribuidos en la parte superior del símbolo.

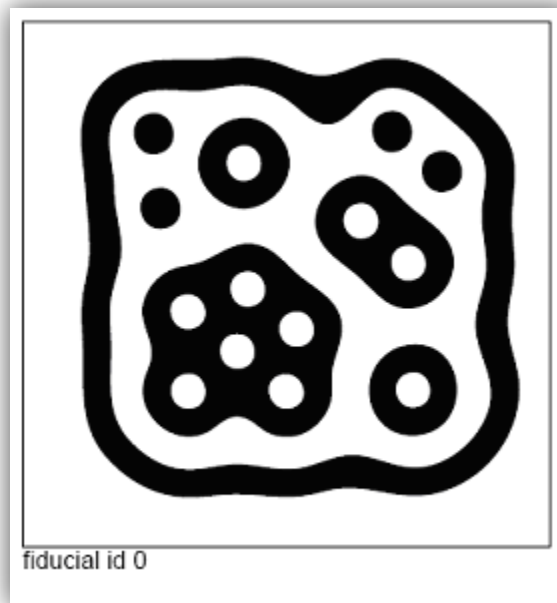


Figura 3. 2: Fiducial amiba.

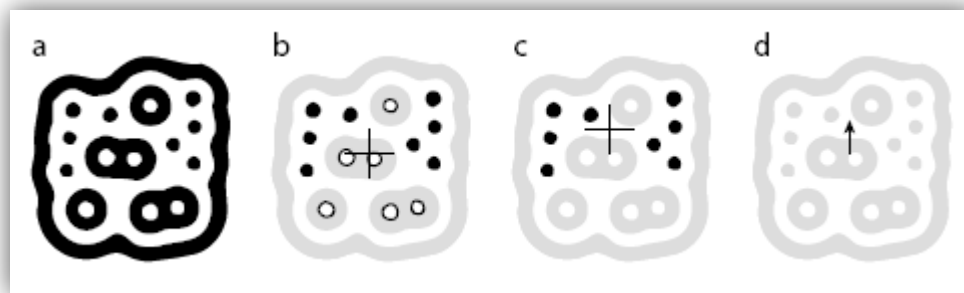


Figura 3. 3: (a)fiducial de reactIVision, (b) nodos hoja blancos y negros y su centroide promedio, (c) nodos hoja negros y su centroide promedio, (d) vector usado para calcular la orientación del fiducial.

3.1.2 Cámaras y lentes

Gran parte de la tarea de reconocimiento está obviamente ligada a la calidad de la imagen adquirida, para un correcto funcionamiento de la plataforma reactIVision es necesario que la cámara se encuentre bien enfocada sobre la superficie de trabajo, además de tener un tiempo de exposición adecuado a fin de evitar una imagen de captura borrosa o desvanecida, de igual forma la superficie de trabajo debe encontrarse bien iluminada.

ReactIVision en general trabaja con cualquier tipo de cámara y lente. Para una buena calidad es conveniente el uso de cámara digital o webcam, USB o FireWire, con resolución mínima de 640x480 a 30 cuadros por segundo. Para una mejor calidad cámaras USB2 proveen mejor resolución y cuadros por segundo. Los lentes de ángulo amplio u ojo de pescado también pueden ser utilizados, sin embargo, es conveniente una calibración por software de la plataforma para obtener un resultado adecuado.

3.1.3 Interpretación de la Información

Es importante la definición de elementos que puedan ser interpretados y analizados por la interfaz, con la finalidad de dar un significado a los elementos visualizados por la cámara. En este caso se ocuparán distintos patrones para cada uno de los elementos del medio, siendo el patrón *fiducial* 0 el correspondiente al primer robot, el patrón *fiducial* 3 correspondiente al segundo robot, los *fiducials* 1, 2, 4 y 5 correspondientes a los objetos a mover, los patrones 6, 7, 8 y 9 correspondientes a los lugares asignados de las cajas y cualquier otro patrón *fiducial* será reconocido como un obstáculo. Se explicará a detalle en el capítulo VI.

Para este trabajo se utilizó la HandyCam Sony modelo DCR-HC28 conectada a la computadora con cable FireWire y con resolución NTSC de 720x480. La pérdida de linealidad asociada con el lente de la cámara se supone despreciable para fines prácticos.

Los valores de posición en X y Y son relativos a la posición en la pantalla de captura y se encuentran entre los valores (0,0) y (1,1). Se aprovecha esta característica para que los valores adquiridos por reactIVision puedan ser independientes de la lejanía o cercanía de la cámara con el campo de trabajo, en futuros desarrollos con una cámara fija se puede asignar un factor de escala a los valores de posición.

El sistema de referencia utilizado en este trabajo en función de reactIVision localiza la coordenada (0, 0) en la esquina inferior izquierda de la pantalla de captura y la coordenada (1, 1*relación proporcional de la pantalla de captura) en la esquina superior derecha de la pantalla de captura (**Figura 3. 4**), los ángulos son medidos de la manera convencional, positivos en el sentido contrario a las manecillas de reloj.

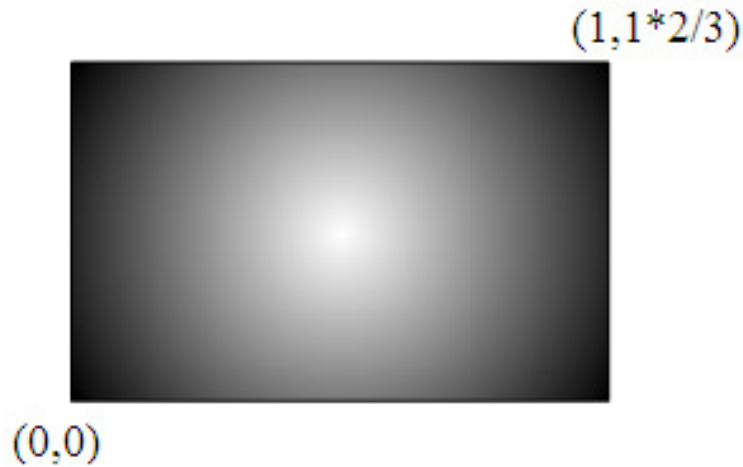


Figura 3. 4: Pantalla de captura de 720x480.

3.2 Comunicación inalámbrica

3.2.1 Introducción

La comunicación es un proceso imprescindible para cualquier robot, por muy poca autonomía (entendiendo por autonomía a la capacidad de no depender más que de sí mismo para realizar cualquier tarea) de la que éste goce. Según la RAE [22], comunicar es transmitir señales mediante un código común al emisor y receptor. El robot puede ser emisor y receptor indistintamente ya que puede enviar y recibir información a/de otro receptor/emisor, respectivamente.

La comunicación inalámbrica es aquella en la que extremos de la comunicación no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. De acuerdo al órgano regulador de la IEEE existen distintos tipos de protocolos de comunicación, los cuales están estandarizados según la velocidad de transmisión y su alcance, los más importantes se muestran en la **Figura 3. 5.**

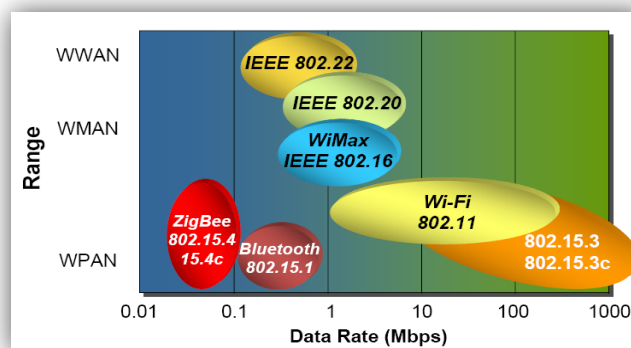


Figura 3. 5: Estándares de Familias inalámbricas

3.2.2 Protocolo Zigbee.

Zigbee es una plataforma global abierta estandarizada bajo la norma 802.15.4 de la IEEE diseñada para soportar capacidades avanzadas de acoplamiento de enroutados [23]. Las especificaciones de Zigbee son desarrolladas por un creciente consorcio de compañías que conforman la “Alianza Zigbee”. Esta Alianza está formada por más de 300 miembros, incluyendo semiconductor, módulos, *stacks* y desarrolladores de software.

3.2.2.1 Arquitectura

Zigbee define tres diferentes dispositivos: coordinador, router y dispositivo final (*coordinator, router y end devices*).

El **coordinador** tiene las siguientes características:

- Selecciona el canal y el PAN ID (64-bit y 16-bit) para comenzar la red.
- Puede permitir a routers y dispositivos finales a unirse a la red.
- Puede asistir en el enroutado de datos.
- No puede *dormir*. Debe de estar activo siempre.

El **router** tiene las siguientes características:

- Debe unirse a un PAN-Zigbee antes de poder transmitir, recibir o enroutar datos.
- Después de unirse, puede permitir a routers y dispositivos finales a unirse a la red.
- Después de unirse, puede asistir en el enroutado de datos.
- No puede *dormir*. Debe de estar activo siempre.

El **dispositivo final** tiene las siguientes características:

- Debe unirse a un PAN-Zigbee antes de poder transmitir o recibir datos.
- No puede permitir a dispositivos a unirse a la red.
- Siempre debe transmitir y recibir datos de RF a través de su *pariente*. No puede enroutar datos.
- Puede entrar a niveles bajos de potencia para conservarla y también puede ser energizado con baterías.

En una red Zigbee, el coordinador debe seleccionar un PAN ID (64-bit y 16-bit) así como el canal para empezar la red. Después de eso, se comporta prácticamente como un router. El coordinador y el router pueden permitir a otros dispositivos a unirse a la red y enroutar datos.

Después que un dispositivo final se une a un router o a un coordinador, éste debe de ser capaz de transmitir o recibir datos de RF a través de dicho router o coordinador. El router o coordinador que permita a un dispositivo final unirse se convierte en el *pariente* de dicho dispositivo final. Debido a que el dispositivo final puede dormir, el pariente debe de

ser capaz de retener paquetes de datos entrantes para el dispositivo final hasta que sea capaz de despertar y recibir la información.

La arquitectura se muestra en la **Figura 3. 6(a)** y su respectiva aplicación en la **Figura 3. 6(b)**.

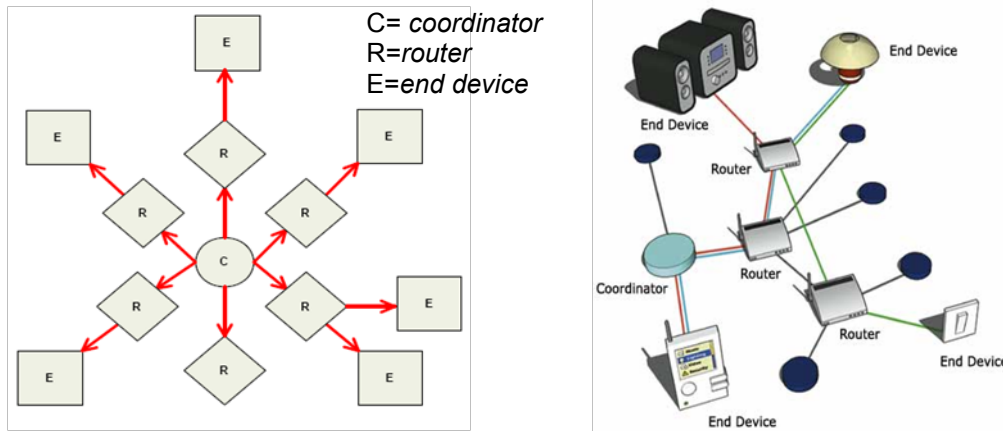


Figura 3. 6: (a)Arquitectura Zigbee.

(b)Aplicación.

A continuación se muestra una tabla con las diferencias de operación entre el modo *Transparente* y el modo *API*.

Características de Operación Transparente	
Interfaz simple.	Todos los datos recibidos serialmente son transmitidos excepto si el modulo está en modo comando.
Fácil soporte.	Es más fácil para una aplicación de soportar la operación transparente y el modo comando.
Características de Operación API	
Fácil manejo de transmisión de datos a múltiples destinos.	Transmitir datos de RF a múltiples remotos sólo requiere cambiar la dirección en el esquema. Este proceso es mucho más rápido que en el modo transparente donde se tiene que entrar a modo comando AT, cambiar de dirección, salir del modo comando y después transmitir datos.
La trama de datos recibidos indica la dirección del remitente.	Todos los datos de RF API indican la fuente de dirección.

3.2.2.2 Modos de operación

Existen dos modos de operación según la aplicación que se requiera: modo transparente y modo API, la siguiente tabla compara las ventajas entre ambos.

En este trabajo se utilizó el modo API ya que se tienen dos robots que reciben información particular de la unidad central (PC), la cual consta de dos módulos Xbee conectados vía

USB a distintos puertos, dichos módulos fueron configurados como coordinadores gracias al software basado en Windows X-CTU provisto por Digi [24]. Por otra parte los módulos que fueron asignados a cada cobot fueron configurados como *routers*. Cabe aclarar que para el correcto funcionamiento de la red es necesario que todos los módulos se encuentren en el mismo canal de radio y pertenezcan a la misma PAN ID (**Figura 3. 7 (a) y (b)**).

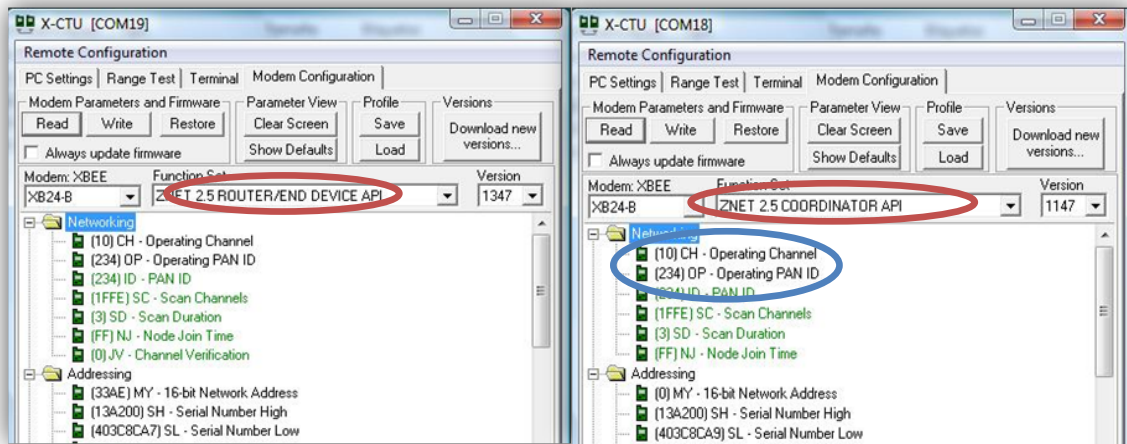


Figura 3. 7: (a)Módulo Coordinador.

(b)Módulo Router.

3.3 Interfaz

Se diseñó una interfaz en C# para el monitoreo de los cobots. Para su explicación, la dividiremos en 4 partes:

En la **Figura 3.8** visualizamos los valores obtenidos por reactIVision de posición y ángulo de cada cobot. Como recordamos, al primero se le asignó el *fiducial 0* y al segundo el *fiducial 3*. Estos valores fueron utilizados para la obtención de los vectores resultantes atractivos y repulsivos respecto al plano X-Y así como su ángulo resultante, dicho modelo se explicará en el siguiente capítulo. Los valores de posición y campos potenciales son unitarios, el ángulo obtenido por reactIVision se encuentra en un rango entre 0° y 360°.

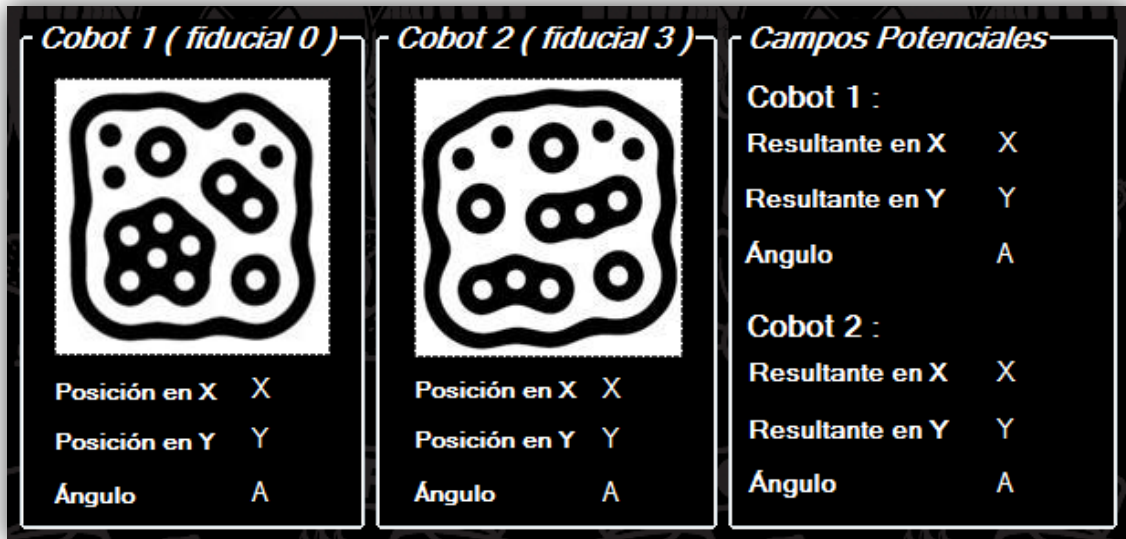


Figura 3. 8: Posiciones y campos antes del inicio del programa.

En la **Figura 3.9** se observa el control de velocidad el cual muestra las velocidades angulares de la llanta izquierda y derecha W_i y W_d obtenidas de la tarjeta MD-25, éstas velocidades se encuentran en un rango entre $0 < 128$, cuando el cobot va hacia atrás y entre $128 > 255$ cuando el cobot va hacia adelante. Dichos valores fueron explicados anteriormente, siendo 128 valor equivalente a 0 [rad/s] (velocidad mínima) y 0 o 255 equivalente a ± 0.7 [rad/s] (velocidad máxima para esta tesis). También observamos la velocidad lineal y angular a lo largo de la trayectoria que lleva cada cobot, cuya obtención se basa en el modelo explicado en el siguiente capítulo.



Figura 3. 9: Control de velocidad y Distancias.

Por otro lado, se agregaron las distancias que tienen los cobots respecto a sus cajas, a sus metas y entre ellos. Cabe señalar que las unidades están normalizadas y van de [0-1].

Posteriormente en la **Figura 3.10** se tiene la forma en que se inicializa la comunicación entre la PC y los cobots. Primero se selecciona el puerto de un coordinador que será asignado al router del cobot, después la velocidad en la que trabajan cuidando que hayan sido configurados en el mismo valor. Finalmente se abre el puerto para esperar la transferencia de datos.



Figura 3. 10: Comunicación PC - Cobot.

Complementando la interfaz, en la **Figura 3.11** se tienen tres botones. El primero inicializa o detiene el programa según sea el caso y cambia hasta que se vuelve a presionar, el segundo exporta los datos a Excel sólo cuando la tarea concluye ya que mientras tanto se encuentra inhabilitado y el tercero cierra el programa.

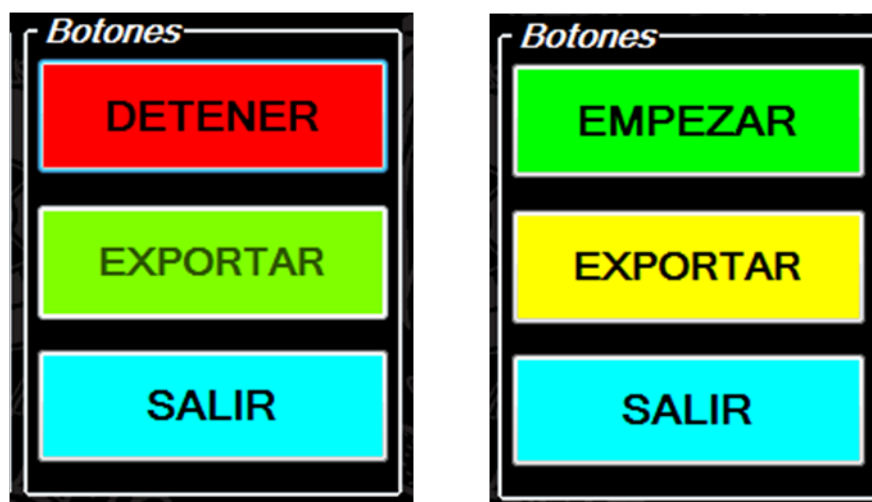




Figura 3. 11: (a)Comienzo del programa.

(b)Programa en curso.




UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Ingeniería. Departamento de Mecatrónica




**SISTEMA SEMI-AUTÓNOMO DE ROBOTS MÓVILES COLABORATIVOS
PARA EL MANEJO DE MATERIALES**

Cobot 1 (fiducial 0)



Posición en X X
Posición en Y Y
Ángulo A

Cobot 2 (fiducial 3)



Posición en X X
Posición en Y Y
Ángulo A

Campos Potenciales

Cobot 1 :
Resultante en X X
Resultante en Y Y
Ángulo A

Cobot 2 :
Resultante en X X
Resultante en Y Y
Ángulo A

Control de velocidad

Cobot 1 :
Llanta derecha Wd
Llanta izquierda Wi
Vel lineal Vlin
Vel. Angular Vang



Cobot 2 :
Llanta derecha Wd
Llanta izquierda Wi
Vel lineal Vlin
Vel. Angular Vang

Distancias

Cobot 1 :
Distancia a su caja Dc
Distancia a su meta Dm
Distancia al C2 Do

Cobot 2 :
Distancia a su caja Dc
Distancia a su meta Dm
Distancia a meta C1 C1
Distancia a meta C2 Dist2

Comunicación

Cobot 1: Puerto	Seleccione un puerto	Abrir	Cerrador	 40649DAA
Velocidad (baudios)	115200	Abrir	Cerrar	 403C8CB1
Cobot 2: Puerto	Seleccione un puerto	Abrir	Cerrar	
Velocidad (baudios)	115200			

Botones

EMPEZAR

EXPORTAR

SALIR

ALUMNOS : Alfredo Miguel Ángeles García.
Daniel Lima Robleda.

TUTOR : Dr. Víctor Javier González Villela.

2010




Figura 3. 12: Interfaz para el manejo de los cobots.

CAPÍTULO IV. PLANIFICACIÓN DE TRAYECTORIAS Y TAREAS

4.1 Localización de un robot en el plano.

Es la estimación de la posición de un robot dentro de un área que determina un sistema de coordenadas. Tanto en ambientes internos como externos los robots necesitan implementar alguna técnica que les permita conocer su posición con el fin de completar satisfactoriamente su objetivo.

La mayoría de las técnicas usadas al respecto, utilizan la información obtenida del ambiente o entorno a través de los sensores del robot y/o las medidas de odometría que permiten registrar los incrementos de posición del robot. Otras técnicas utilizan elementos incorporados al ambiente denominados “marcas” que permiten triangular la posición del robot. En [25] se muestran estas técnicas con más detalle.

El problema de localización suele clasificarse en:

- **Localización local:** Consiste en determinar la posición del robot a través del conocimiento a priori de su posición inicial, haciendo seguimiento del movimiento del robot (traslacional y rotacional) usando odometría. Sin embargo ésta presenta errores acumulativos clasificados como sistemáticos y no sistemáticos. Los primeros dependen de las propiedades del robot (resolución de encoders, diámetro de las llantas, etc.) mientras que los segundos dependen de las condiciones del ambiente como por ejemplo el coeficiente de fricción de la superficie de desplazamiento del robot. Para corregir dichos errores, ésta técnica se suele combinar con el uso de marcas para hacer triangulación y corregir la posición [26].
- **Localización Global:** Permite determinar la posición del robot sin tener como referencia una posición inicial. Para lo cual se manejan hipótesis de la posición utilizando enfoques probabilísticos entre los que se destacan el análisis bayesiano, las cadenas de Markov o el filtro de Kalman, en [27] se hace una referencia de estas técnicas. En [27] se presenta una combinación del filtro de Kalman con métodos de Lógica Difusa.

Las técnicas citadas tienen sus limitaciones, por ejemplo, para entornos demasiado complejos, mantener un modelo actualizado y consistente del entorno representa una dificultad enorme que se manifiesta en un considerable consumo de recursos de cómputo y que de una u otra forma afectan los tiempos de respuesta del robot. Por otra parte muchas de las técnicas mencionadas han sido enfocadas para ambientes estáticos que asumen que el robot es el único objeto en movimiento.

Por estas razones, uno de los planteamientos de este trabajo es un esquema de localización por medio de un sistema de visión manejado por un software que reconoce patrones establecidos colocados sobre los robots para hacer fácil la identificación de dichos elementos.

4.2 Tipos de Arquitecturas.

Desde el punto de vista de la planificación, existen diferentes arquitecturas diseñadas teniendo en cuenta especificaciones sobre eficiencia, tiempo de respuesta ante acontecimientos no previstos, y disponibilidad de información del entorno.

En un extremo se sitúan las arquitecturas orientadas esencialmente a soportar una planificación *puramente estratégica*, utilizando modelos del entorno suficientemente precisos, suponiendo que la situación en la que va a ejecutarse la tarea puede ser conocida a priori. Son arquitecturas funcionales basadas en planificación con modelo centralizado para verificación de la información suministrada por los sensores y generación de acciones.

En el otro extremo se sitúan las arquitecturas orientadas esencialmente a suministrar flexibilidad, adaptabilidad y capacidad de reacción ante condiciones no previstas. Las arquitecturas *puramente reactivas* implementan una estrategia de control como una colección de pares condiciones-acciones. El sistema consiste en una colección de reglas reactivas, conteniendo una descripción interna mínima del estado. Se basa en un acoplamiento directo entre sensores y actuaciones mediante bucles rápidos de realimentación.

Las arquitecturas basadas en *comportamientos (behaviors)* [28] pueden considerarse como una evolución de las puramente reactivas. Los comportamientos describen la forma en que se reacciona ante un determinado estado de los sensores, y las actuaciones se determinan de forma más elaborada que la simple consulta a una tabla de correspondencias, utilizándose distintas formas de representaciones internas y realizándose con ellas cálculos para decidir qué acción hay que realizar. De esta forma, cabe mencionar comportamientos tales como “evitar obstáculos”, o “levantar objeto”. En las arquitecturas basadas en comportamientos, se trata de emplear controladores sencillos fáciles de depurar y utilizar.

Las denominadas *arquitecturas híbridas* ofrecen un compromiso entre las puramente reactivas y las orientadas a planes. Se emplea un sistema reactivo para control de bajo nivel, y un planificador para tomas de decisiones de un nivel superior. Se separa el sistema de control en dos o más partes comunicadas, pero básicamente independientes. En la mayor parte de los casos, el proceso reactivo de bajo nivel se encarga de funciones de seguridad, mientras que el de alto nivel utiliza un planificador para seleccionar la acción. Esta arquitectura es en la cual está basado nuestro trabajo.

4.2.1 Diseño funcional de la arquitectura

El enfoque típico consiste en partir de las tareas que es necesario realizar y descomponerlas en funciones, definiendo las interacciones entre ellas. Los problemas típicos que hay que abordar en el diseño de arquitecturas de control inteligente son los siguientes:

1. Definición de niveles de jerarquización.

Se suele seguir un enfoque descendente con refinamientos jerárquicos y planes descendentes hasta llegar a los bucles de control de los actuadores. El ciclo de tiempo para realizar una determinada función puede variar entre minutos, para razonamiento simbólico, y milisegundos, en el control de los actuadores.

Por consiguiente parece necesario especificar de la forma más precisa y cuantitativa posible los requerimientos necesarios y evaluar la complejidad asociada a cada tarea. Es claro que una definición que no contemple funciones que después pudieran ser relevantes dificultaría la implantación posterior de dichas funciones, mientras que una especificación con niveles carentes de suficiente contenido puede llevar a una complejidad artificial de la arquitectura que origine rendimientos inferiores.

2. Compromiso planes/reacciones.

Este compromiso es hoy día típico en el diseño de arquitecturas de sistemas de control inteligente de robots. Se trata de establecer un compromiso entre la ejecución de planes mediante estrategias de búsqueda, de la forma más aproximada posible a la óptima, para alcanzar unos objetivos previamente definidos, y la capacidad de reacción a contingencias no previstas a partir de información sensorial.

3. Conocimiento del estado.

Se trata de definir la información que debe mantener el sistema de control sobre el estado del robot y su entorno. Se representa el entorno mediante mapas y/o conocimiento simbólico, así como la definición de estructuras de datos que representen de forma adecuada dicho estado en tiempo real. Nótese como en la representación del entorno a partir de sistemas de visión pueden involucrarse datos tan diversos como imágenes, objetos geométricos y símbolos.

4. Otras características.

En la definición funcional de la arquitectura es necesario involucrar también la definición de prioridades, la asignación de tareas, típicamente continua en algunas funciones, tales como la evasión de obstáculos, y bajo demanda de otras, tales como el reconocimiento de un determinado objeto.

4.3 Campos Potenciales.

El comportamiento basado en campos potenciales artificiales [29] es un método que genera una respuesta rápida a la información obtenida por los sensores. Constituye también un método de arquitectura híbrida para la planificación en el que se prima la

reactividad, integrándose las funciones de planificación con las de control de movimientos.

También se conoce como *Navegación mediante evasión de obstáculos en línea*; éste método genera un campo vectorial que representa un espacio navegacional sujeto a una función potencial arbitraria diferenciable. El valor de una función potencial puede ser visto como energía y el gradiente del potencial como una fuerza. La teoría de campos potenciales considera al robot como *una partícula positivamente cargada*, influenciada por un campo de potencial (campo de fuerzas). Las fuerzas pueden ser de dos tipos:

- **Fuerzas de atracción:** Una meta o posición a alcanzar, o comportamiento *SeekGoal* (buscar meta), **Figura 4. 1**, tiene como objetivo llevar al cobot hacia una meta identificada. El conjunto de vectores crea un campo que representa potenciales sintéticos de energía para el cobot.
- **Fuerzas de repulsión:** Obstáculos, o comportamiento *AvoidObstacle* (evitar obstáculo), tiene como objetivo generar alejamiento de un objeto, un campo repulsivo, ligado a un radio de influencia, fuera del cual el potencial repulsivo es cero, **Figura 4. 2**.

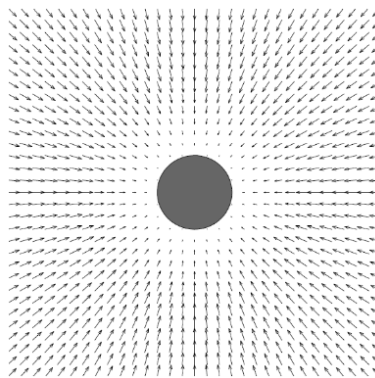


Figura 4. 1: Campo atractivo correspondiente al comportamiento "SeekGoal"

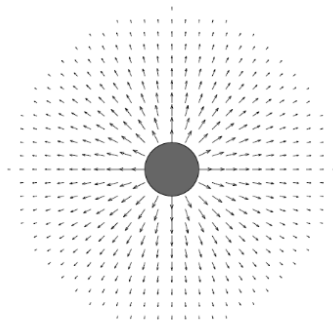


Figura 4. 2: Campo repulsivo correspondiente a comportamiento "AvoidObstacle".

La combinación de fuerzas atractivas y repulsivas generar un campo potencial artificial que cumpla con los dos comportamientos de forma simultánea y eficiente, como se muestra en la **Figura 4. 3**, dirigiendo cuidadosamente al cobot de una localidad inicial hacia una localidad meta, al mismo tiempo que permiten al robot ir evitando obstáculos.

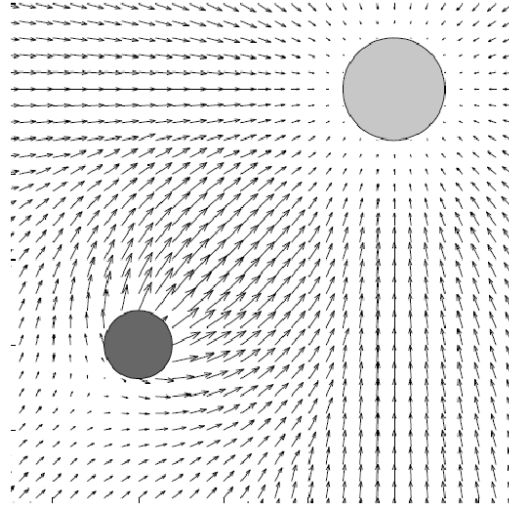


Figura 4. 3: Campos potenciales artificiales.

4.3.1 Modelo matemático de campos potenciales.

En [30] se plantea el modelo matemático del campo para un espacio determinado y llevar a cabo su aplicación.

De acuerdo con [30] la fuerza de atracción queda definida por:

$$F_{atr}(q) = \begin{cases} -\varepsilon_1(q - q_{des}) & \forall |q - q_{des}| \leq d_1 \\ -\varepsilon_1 \frac{(q - q_{des})}{|q - q_{des}|} & \forall |q - q_{des}| > d_1 \end{cases}$$

y la fuerzas de repulsión por:

$$F_{rep}(q) = \begin{cases} \mu \left(\frac{1}{|q - q_{obs}|} - \frac{1}{d_0} \right) * \frac{1}{|q - q_{obs}|^2} * \frac{(q - q_{obs})}{|q - q_{obs}|} & \forall |q - q_{obs}| \leq d_0 \\ \emptyset & \forall |q - q_{obs}| > d_0 \end{cases}$$

donde:

$q \triangleq$ posición inicial del cobot.

$q_{obs} \triangleq$ posición del obstáculo.

$q_{des} \triangleq$ posición del destino.

$\varepsilon_1 \triangleq$ constante de atracción.

$\mu \triangleq$ constante de repulsión.

$d_1, d_0 \triangleq$ radio de influencia al robot y al obstáculo respectivamente.

Los valores obtenidos por medio de los campos potenciales artificiales nos brindan la trayectoria óptima para llegar a nuestro objetivo por la ruta más eficiente de acuerdo con los parámetros establecidos en el modelo, quedando este determinado por los valores de las constantes atractivas y repulsivas.

4.3.2 Limitaciones de la navegación mediante campos potenciales.

El problema en este tipo de método recae en la presencia de mínimos locales, es decir, lugares que no indican el verdadero mínimo global, y el potencial resulta nulo. Una situación así lleva al cobot a una posición que no es el destino que se busca alcanzar, o pudiera quedar atrapado en una zona cerrada, por ejemplo, avanzando de un lado a otro pero sin salir de dicha área.

Una forma de solucionar es por medio de un planificador global que podría entonces trazar desde la posición actual del cobot, un camino diferente hacia el objetivo, sin embargo el coste computacional resulta excesivamente alto para aplicarlo en cada ciclo en un sistema de tiempo real.

Otras de las alternativas es emplear comportamientos que dirijan al cobot hacia cualquier lugar, es decir, den un movimiento aleatorio para lograr sacar al cobot del mínimo local.

4.4 Planificación de Caminos.

Si el robot está en una posición y debe desplazarse a otra posición distinta, es necesario encontrar un camino a seguir, y sea de preferencia el camino más corto, evitando chocar con los obstáculos presentes en el entorno.

Existen dos casos:

- Conocemos el entorno de navegación.
- Desconocemos el entorno de navegación y los obstáculos pueden cambiar de posición.

La planeación de caminos y navegación de un robot móvil puede entonces llevarse a cabo mediante un sistema de visión que facilite identificar lugares mediante el reconocimiento de marcas artificiales colocadas estratégicamente en el entorno de navegación.

4.5 Planificación de la tarea.

Para que los robots realicen la tarea de forma efectiva se debe contar con un planeador para que le asigne a cada robot una tarea en específico y que no se repita.

Para ello jerarquizamos los robots y desarrollamos un algoritmo de planificación de tareas el cual consiste básicamente en determinar las distancias de los robots a cada una de las metas y la de menor valor se le asigna a los robots para que se dirijan hacia ese punto.

Para evitar que el planeador le asigne la misma meta a los robots, el algoritmo calcula primero las distancias de las metas al robot de mayor jerarquía, le asigna la meta que se encuentra a la menor distancia y deshabilita dicha meta; posteriormente calcula las distancias de las metas restantes al robot de menor jerarquía, se le asigna de igual forma la que se encuentra más cerca y se deshabilita dicha meta.

Una vez asignadas las metas a cada robot, estos se dirigen hacia dicho punto, en el momento en que los robots alcancen dicho punto el algoritmo le asigna a cada robot la nueva posición a dónde dirigirse la cual selecciona de una tabla de correspondencia, ya que cada meta se le ha asignado previamente una posición final.

CAPÍTULO V. MODELO DE CONTROL PARA EL ROBOT MÓVIL

El control automático es una herramienta necesaria para que los robots puedan realizar de manera eficaz y eficiente las tareas asignadas a ellos. En general ha constituido una herramienta vital para la industria como por ejemplo en las máquinas de control numérico para la manufactura de herramientas, en líneas de producción en diferentes ramas como la automotriz, farmacéutica, de alimentos, etc.

El control se encarga de mantener un punto de operación en particular en los sistemas y procesos, éste se realiza a través de variables. Una de ellas es la variable controlada, que es la cantidad o condición que se mide y controla, otra es la variable manipulada, que es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada. Controlar significa medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar la desviación del valor medio respecto del valor deseado.

El objetivo del control automático en robótica móvil es controlar el robot a partir de la información procedente de sensores con objeto de ejecutar un plan establecido o realizar una determinada tarea reaccionando ante cambios en el entorno.

5.1 Modelo de robots móviles.

Para poder realizar el control de un robot móvil es necesario conocer el modelo geométrico, cinemático y dinámico del robot. El modelo geométrico del robot es la relación entre los valores de las variables de acción y la situación (posición y orientación) de un sistema de referencia que se elige normalmente al punto de guía deseado, solidario al robot, que se define teniendo en cuenta la tarea que se pretende desarrollar con el robot.

La cinemática estudia el movimiento sin considerar las fuerzas que lo producen, por lo tanto, se trata de estudiar tanto las propiedades geométricas como las temporales del movimiento. En términos generales, se considera, además del problema puramente geométrico involucrado en el posicionamiento estático, las variaciones en el tiempo de las posiciones y orientaciones; es decir, las velocidades u aceleraciones.

En el modelo geométrico y cinemático se involucra esencialmente el estudio de las relaciones existentes entre el espacio de las variables de acción y el espacio de trabajo que suele ser un espacio cartesiano.

Asimismo, el modelar implica tener una representación de los robots que permita hacer inferencias sobre otros agentes. Esto nos permite hacer más eficiente la comunicación

entre ellos, facilita la comunicación implícita mediante el ambiente y sensores y también permite la colaboración de robots heterogéneos.

5.1.1 Modelo cinemático de robots móviles.

Para facilitar el modelo del robot se consideran las siguientes hipótesis simplificadoras:

- El robot se mueve sobre una superficie plana.
- Los ejes de guiado son perpendiculares al suelo.
- Se supone que las ruedas se mueven con rodadura pura; es decir el deslizamiento es despreciable.
- El robot no tiene partes flexibles.
- Durante un periodo de tiempo suficientemente pequeño en el que se mantiene constante la consigna de dirección, el vehículo se moverá de un punto al siguiente a lo largo de un arco de circunferencia.
- El robot se comporta como un sólido rígido.

Considérese un sistema de referencia {G} y un sistema {L} con centro en el punto de guiado del vehículo y eje \hat{Y}_L en la dirección del eje longitudinal del vehículo.

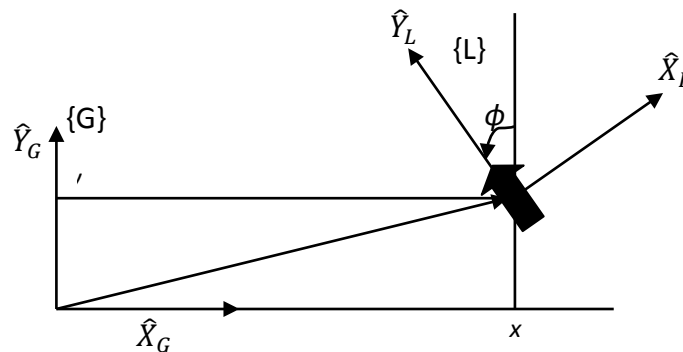


Figura 5. 1: Cambio del sistema de referencia en navegación de robots móviles.

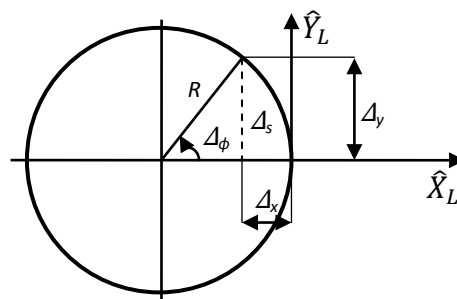


Figura 5. 2: Círculo osculador.

Supóngase que el vehículo se desplaza en un intervalo de control según un arco de circunferencia. Esta suposición es válida para intervalos de control suficientemente pequeños.

Entonces la velocidad lineal del vehículo viene dada por:

$$v = \frac{\Delta s}{\Delta t} \quad (5.1)$$

y la velocidad angular por:

$$\omega = \frac{\Delta \varphi}{\Delta t} \quad (5.2)$$

La longitud del arco recorrido por el robot en el mismo intervalo de tiempo viene dada por:

$$\Delta s = R \Delta \varphi \quad (5.3)$$

siendo R el radio de giro o radio de la circunferencia que describe el punto de guiado.

La curvatura se define como la inversa del radio de giro:

$$\gamma = \frac{1}{R} = \frac{\Delta \varphi}{\Delta s} \quad (5.5)$$

Las ecuaciones de movimiento en el sistema $\{L\}$ de la **Figura 5. 2** en la posición inicial son:

$$\Delta x_L = -(R - R \cos(\varphi)) \quad (5.5)$$

$$\Delta y_L = R \operatorname{sen}(\varphi) \quad (5.6)$$

Si la orientación inicial del vehículo con respecto al sistema {G} es de ϕ , el movimiento en el sistema {G} se determina rotando ϕ :

$$\Delta x = R[\cos(\Delta\phi) - 1] \cos(\phi) - R \operatorname{sen}(\Delta\phi) \operatorname{sen}(\phi) \quad (5.7)$$

$$\Delta y = R[\cos(\Delta\phi) - 1] \operatorname{sen}(\phi) + R \operatorname{sen}(\Delta\phi) \cos(\phi)$$

Suponiendo que el intervalo de control es suficientemente pequeño, también lo será el cambio de orientación $\Delta\phi$ con lo cual se tendrá que

$$\cos(\Delta\phi) \cong 1 \quad (5.8)$$

$$\operatorname{sen}(\Delta\phi) \cong \Delta\phi$$

Sustituyendo en las anteriores ecuaciones se tiene que

$$\Delta x = -R\Delta\phi \operatorname{sen}(\phi) \quad (5.9)$$

$$\Delta y = R\Delta\phi \cos(\phi)$$

y teniendo en cuenta (5.3)

$$\Delta x = -\Delta s \operatorname{sen}(\phi) \quad (5.10)$$

$$\Delta y = \Delta s \cos(\phi)$$

Dividiendo ambas ecuaciones por Δt , teniendo en cuenta (5.1) y haciendo tender Δt a cero se llega a:

$$x' = -v \operatorname{sen}(\varphi) \quad (5.11)$$

$$y' = v \cos(\varphi) \quad (5.12)$$

ecuaciones a las que se le puede añadir:

$$\varphi' = \omega \quad (5.13)$$

la cual proporciona la variación de la orientación.

5.1.2 Modelo cinemático para el control del robot móvil diferencial.

En la **Figura 5. 3** se muestra la locomoción con guiado diferencial. En este caso, las variables de control son las velocidades de las ruedas laterales.

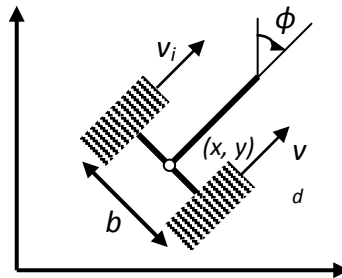


Figura 5. 3: Locomoción con guiado diferencial.

El vector de coordenadas globales del punto de guía de un robot móvil, de acuerdo con 5.11, 5.12 y 5.13, puede expresarse como un sistema de la forma:

$$\begin{bmatrix} x' \\ y' \\ \varphi' \end{bmatrix} = \begin{bmatrix} -\operatorname{sen}(\varphi) & 0 \\ \cos(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Sean ω_i y ω_d , las velocidades de giro de las ruedas izquierda y derecha, respectivamente. Si el radio de la rueda es r , las velocidades lineales correspondientes a las ruedas son $v_i = \omega_i r$ y $v_d = \omega_d r$. En este caso, la velocidad lineal y la velocidad angular vienen dadas por:

$$v = \frac{v_d + v_i}{2} = \frac{(\omega_d + \omega_i)r}{2} \quad (5.15)$$

$$\omega = \frac{v_d - v_i}{b} = \frac{(\omega_d - \omega_i)r}{b} \quad (5.16)$$

Siendo b la distancia que separa las ruedas del vehículo.

Por consiguiente, si se especifican la velocidad v y angular ω del vehículo, las velocidades de giro que hay que aplicar a las ruedas izquierda y derecha son:

$$\omega_i = \frac{v - (b/2)\omega}{r} \quad (5.17)$$

$$\omega_d = \frac{v + (b/2)\omega}{r} \quad (5.18)$$

Y por último aplicando el modelo cinemático inverso correspondiente como lo plantea Ollero en [31] el modelo en términos de las variables de las ruedas, queda:

$$\begin{bmatrix} \omega_i \\ \omega_d \end{bmatrix} = \begin{bmatrix} -(r \cdot \text{sen}(\varphi))/2 & (r \cdot \text{cos}(\varphi))/2 & -r/b \\ -(r \cdot \text{sen}(\varphi))/2 & (r \cdot \text{cos}(\varphi))/2 & r/b \end{bmatrix} \begin{bmatrix} x' \\ y' \\ \varphi' \end{bmatrix}$$

Modelo que controla la velocidad angular de las ruedas del robot en función de la velocidad lineal proyectada sobre los ejes y la velocidad angular del robot móvil.

5.2 Campos Potenciales en el Modelo de Control.

Para obtener una relación entre los valores obtenidos a través de campos potenciales y el modelo de control [32] propone que el problema de seguir la trayectoria puede ser resuelto con las siguientes reglas:

$$v = \begin{cases} v_{max} & \text{si } d_g > k_r \\ \frac{v_{max}}{k_r} \cdot d_g & \text{si } d_g \leq k_r \end{cases} \quad (5.19)$$

donde d_g es la distancia a la meta y k_r es la constante previamente definida para campos atractivos.

Por otra parte la velocidad angular se plantea como función del ángulo de error resultado de la resta entre el ángulo de la fuerza resultante del comportamiento de campos potenciales y el ángulo actual del robot.

$$\theta_e = \theta_{potencialfields} - \theta_{robot} \quad (5.20)$$

La velocidad angular queda determinada por:

$$\omega = \omega_{max} \text{sen}(\theta_e) \quad (5.21)$$

Con 5.20 y 5.21 el comportamiento esperado es que el robot se dirija a la posición final a máxima velocidad lineal cuando se encuentre lejos de la meta y desacelere al aproximarse, mientras su velocidad angular varía en función del ángulo de error dictaminado por el método de campos potenciales.

CAPÍTULO VI. RESULTADOS

En este capítulo se plantea una aplicación práctica para el sistema desarrollado, los datos reportados son resultados de las pruebas experimentales.

6.1 Caso de estudio.

El sistema consiste en dos robots y una unidad central con IA que conoce el problema a resolver y las capacidades de cada robot, de esta manera una vez que la PC ha distribuido las tareas, cada robot cuenta con una IA básica para ejecutar dicha tarea.

Los cobots deberán ser capaces de transportar materiales de un punto a otro sin colisionar entre sí, es decir, deberá recoger el material de una zona de carga y transportarlo hasta el lugar de descarga previamente asignado a dicho material. El cobot deberá recoger el material más cercano considerando su jerarquía.

6.2 Metodología.

El sistema completo consta de tres sub-sistemas principales: el primero es el software de visión "ReactIVision" el cual determina en tiempo real la posición y ángulo de cada elemento fiducial que se encuentra dentro del área de visión, al software se le indica que fiducial corresponde a los cobots, las cajas y los stacks.

El segundo elemento es la interfaz que procesa las posiciones dadas por "ReactIVision" y después los algoritmos de campos potenciales y asignación de tareas determinan que material debe recoger cada cobot y las velocidades de cada una de sus llantas.

Por último se mandan los valores de las velocidades vía inalámbrica con el protocolo de comunicación ZigBee y se procesa la información recibida por medio del microcontrolador colocado en cada cobot para controlar la velocidad de los motores.

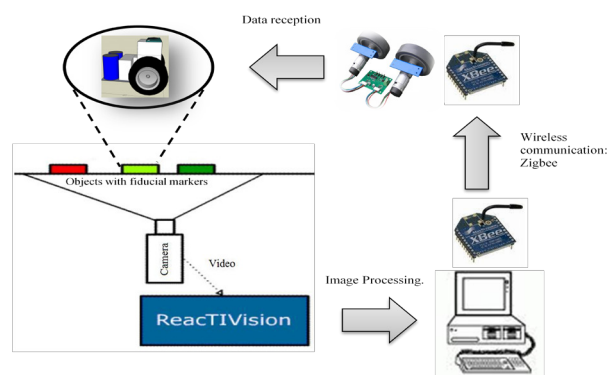


Figura 6. 1: Ciclo del sistema empleado.

6.3 Entorno de las pruebas.

Para poder reconocer los cobots y los demás elementos del medio tales como las metas (cajas que representan materiales) y los *stacks* (lugar de descarga del material), fue necesario adherir el símbolo *fiducial* correspondiente en la parte superior, haciendo coincidir el centroide del símbolo con el centro de giro del robot. La medida del símbolo *fiducial* para el cobot fue de 15x10[cm], para las cajas y los *stacks* fue de 7x7[cm].



Figura 6. 2: Fiducials para los cobots y las cajas.

La distancia entre ruedas de los cobots $b = 0.17$ [m] y el radio de la rueda $r = 0.015$ [m]. La velocidad lineal máxima de los cobots es de 5 [m/s] debido a la velocidad de procesamiento de la imagen.

Para las pruebas se utilizó una cámara de video HandyCam Sony conectada a la computadora con cable FireWire 1394, montada a 2.3[m] de altura.] El área de trabajo enfocada por la cámara fue de aproximadamente 1.50X2.00 [m].



Figura 6. 3: Área de trabajo.

6.4 Pruebas

La notación que se usará en cada prueba es la siguiente:

- R1: Posición (X,Y) del Cobot 1.
- R2: Posición (X,Y) del Cobot 2.
- C1: Posición (X,Y) de la Meta 1.
- C2: Posición (X,Y) de la Meta 2.
- C3: Posición (X,Y) de la Meta 3.
- C4: Posición (X,Y) de la Meta 4.
- S1: Posición (X,Y) del *Stack* 1.
- S2: Posición (X,Y) del *Stack* 2.
- S3: Posición (X,Y) del *Stack* 3.
- S4: Posición (X,Y) del *Stack* 4.

Cabe aclarar que las imágenes de fondo en todas las gráficas siguientes representan el área de trabajo utilizada, sin embargo en algunas ocasiones los puntos en las graficas para representar las posiciones de los cobots, las cajas o los “stacks” no coinciden con la imagen, esto es debido a que la imagen tomada por la cámara de video se ve cóncava y no plana para tener una mejor representación. Recordando también que todas las posiciones se adquieren con el centroide de cada ameba y su posición en el plano se da de forma normalizada, es decir, en el eje X se tienen valores de 0 – 1.0 (cero a uno punto cero) y en el eje Y se obtienen valores de 0 – 0.66 (cero a cero punto sesenta y seis) esto es debido a la relación de la cámara 4:6 donde el eje Y es 2/3 (dos tercios) el eje X.

Se realizaron tres pruebas, cada una presenta características diferentes de nuestro sistema. Es decir, en la primera se muestra la ejecución eficaz del manejo de materiales, en la segunda contempla la teoría de campos potenciales y por último en la tercer prueba observamos el comportamiento de la arquitectura GOFER.

6.4.1 Prueba 1. Transporte de cajas.

El objetivo de esta prueba es comprobar que la recolección, el traslado y la descarga de material se efectúe de la manera más simple y directa, es decir, que cada cobot recoja la caja y la lleve a su lugar previamente asignado; para hacer más eficiente el manejo de materiales, las trayectorias de cada cobot no se cruzan entre sí y tampoco entran al campo de influencia del otro.

Estos parten de reposo en la posición R1(0.27665,0.51066) y R2(0.26514,0.22129), las posiciones fijas de las cajas son C1(0.72010,0.23821), C2(0.86604,0.11330), C3(0.74204,0.59158) y C4(0.86492,0.42648) y las posiciones de los *stacks* son S1(0.15320,0.12819), S2(0.14399,0.26965), S3(0.14553,0.41173) y S4(0.15080,0.56154).

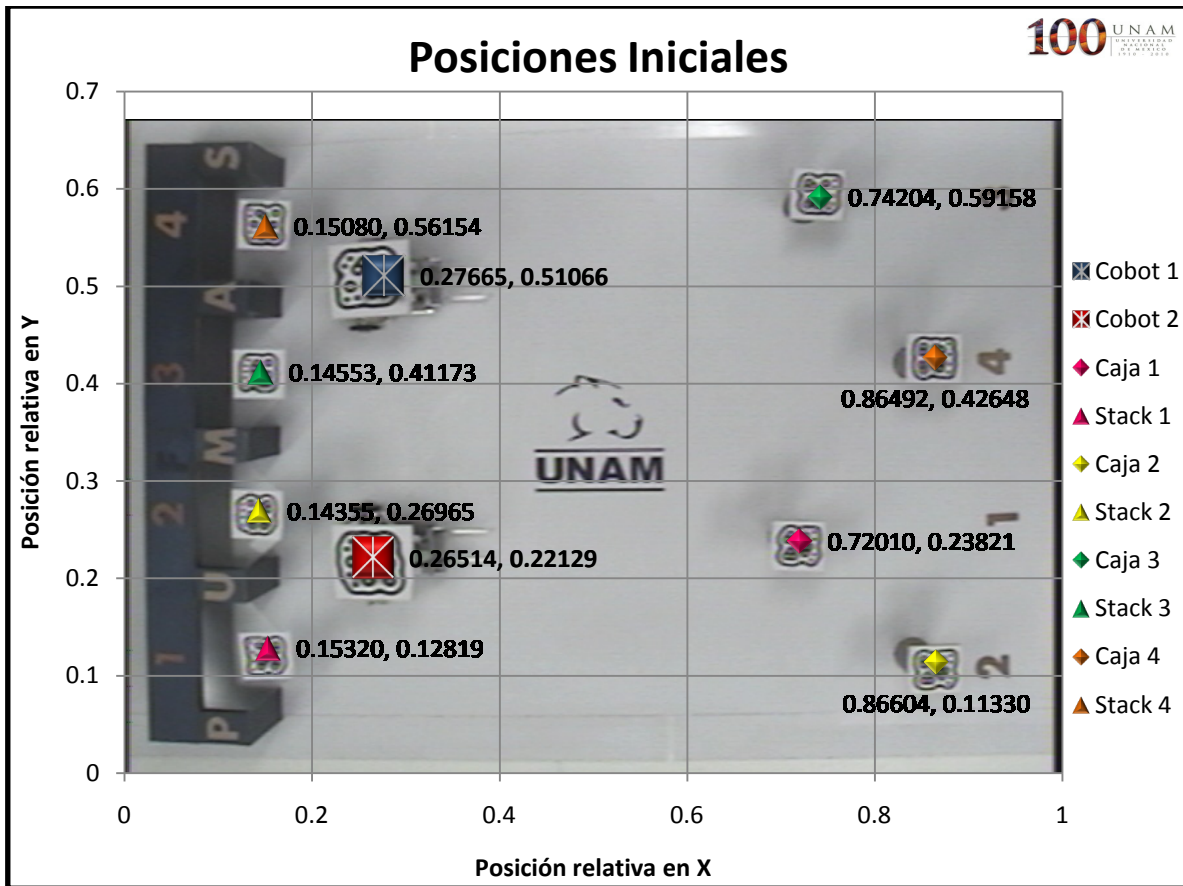


Figura 6. 4: Posiciones iniciales de los elementos que conforman la prueba.

A continuación se presentan las graficas de la trayectoria seguida por el Cobot 1 así como su velocidad lineal y angular.

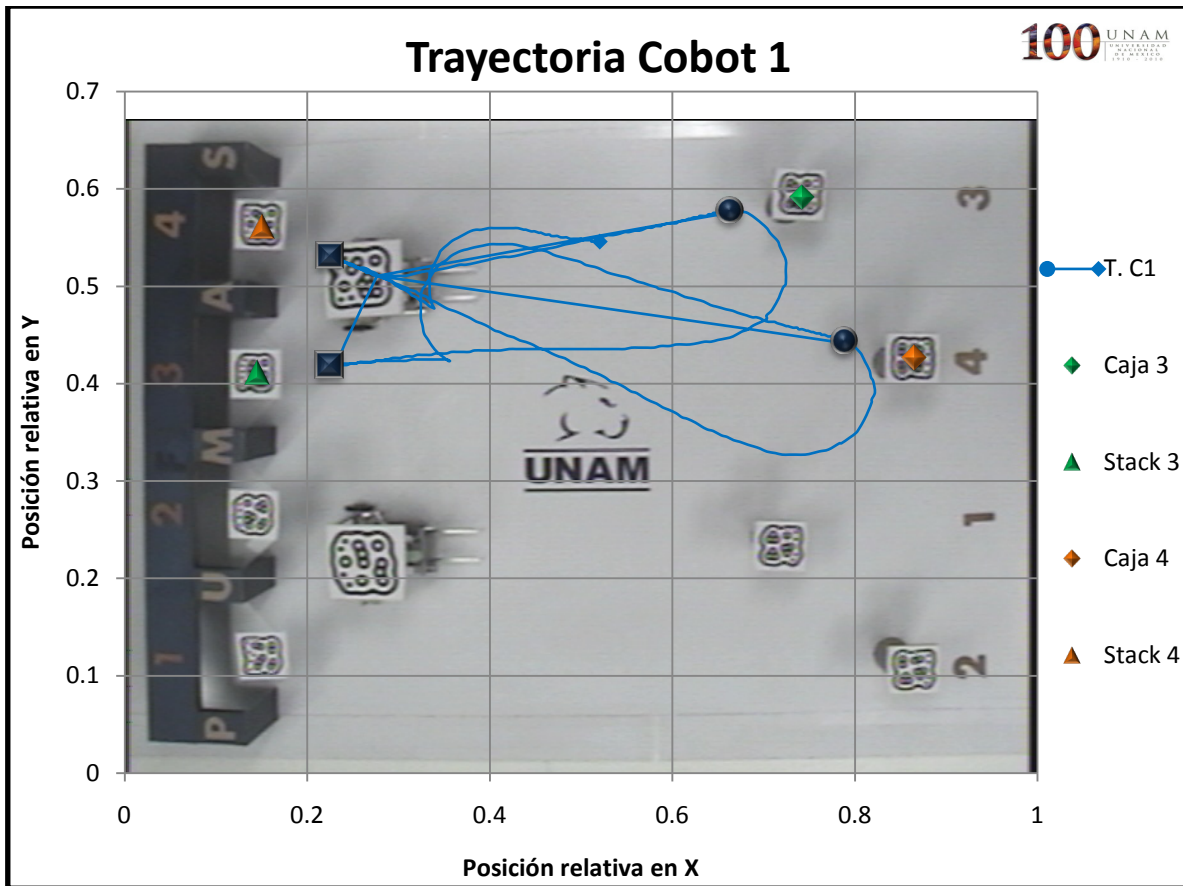


Figura 6. 5: Trayectoria completa del Cobot 1.

Como se puede observar en la **Figura 6. 5** la trayectoria seguida por el Cobot 1 es la más corta y suave para llegar de un punto a otro.

En la **Figura 6. 6** observamos la primera parte de la trayectoria, donde se distingue su inicio y final por la simbología del lado derecho. Asimismo dentro de la trayectoria resaltan dos puntos, un círculo y un cuadrado, que representan la posición final del Cobot 1 cuando se detiene para levantar o dejar su caja. Es necesario que el cobot se encuentre a una distancia adecuada para que el montacargas funcione correctamente y evite colisionar con alguna de sus metas locales, sean cajas o stacks representados por rombos y triángulos respectivamente. Por lo cual se le determinaron a las metas radios de influencia, estos provocan que el Cobot 1 disminuya su velocidad conforme se les acerca y pare completamente cuando la posición del montacargas sea la mejor.

En la **Figura 6. 7** se tiene la segunda parte que completa la trayectoria, y al igual que en la primera la tarea se realizó de la manera esperada.

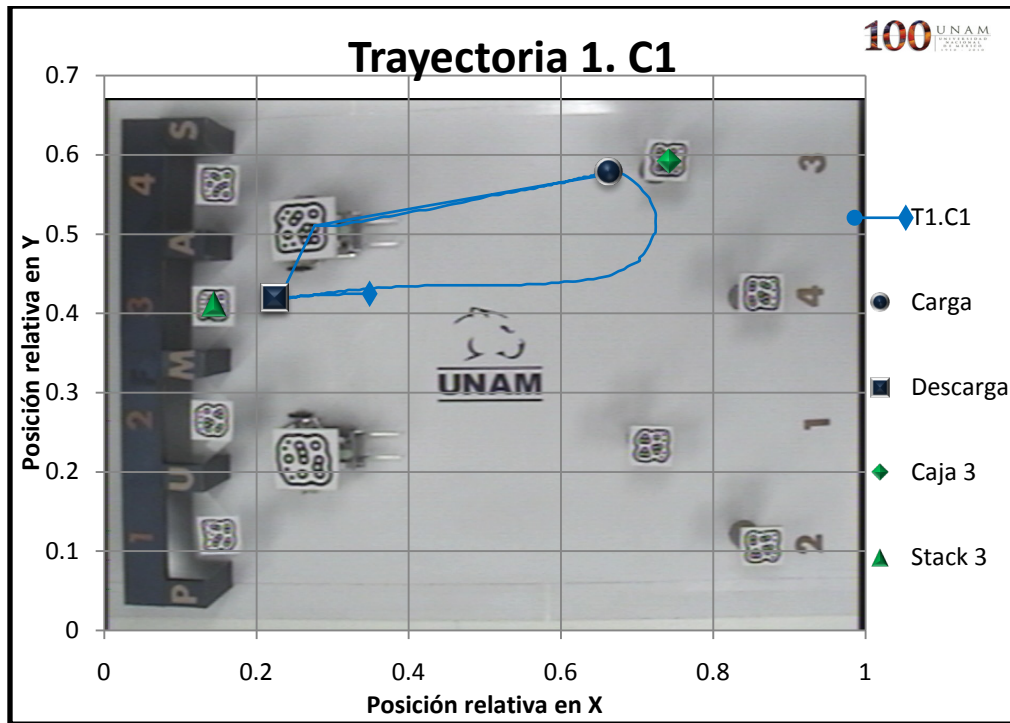


Figura 6. 6: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 3.

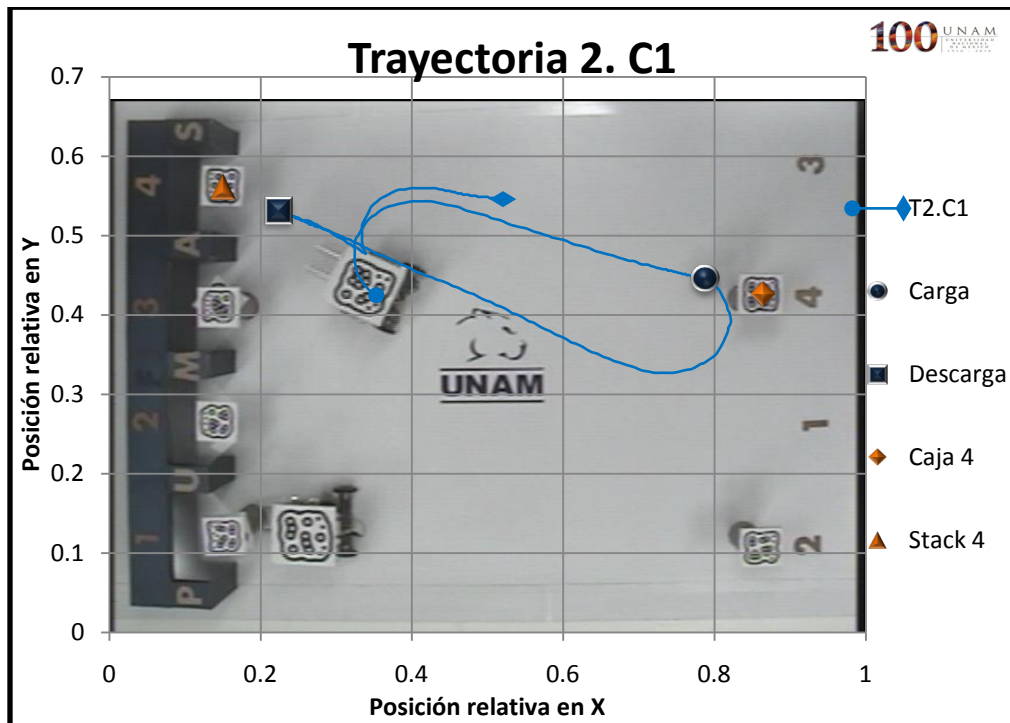


Figura 6. 7: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 4.

En la **Figura 6. 8** se puede observar que se cumple con lo planteado en capítulos anteriores, esto es, el Cobot 1 irá a su máxima velocidad cuando se encuentre lejos de la meta y disminuirá su velocidad hasta detenerse conforme se acerca a ésta. Cabe resaltar que no siempre alcanza su velocidad máxima, ya que puede entrar a un radio de influencia antes de lograrlo, esto lo apreciamos en el último pico. En la **Figura 6. 9** se puede observar que la trayectoria fue suave, es decir, lo más recta posible cuando la velocidad angular tiende a cero y el Cobot solamente gira a su máxima velocidad angular $-6[\text{rad/s}]$ cuando requiere girar casi 180° para dirigirse a su siguiente meta.

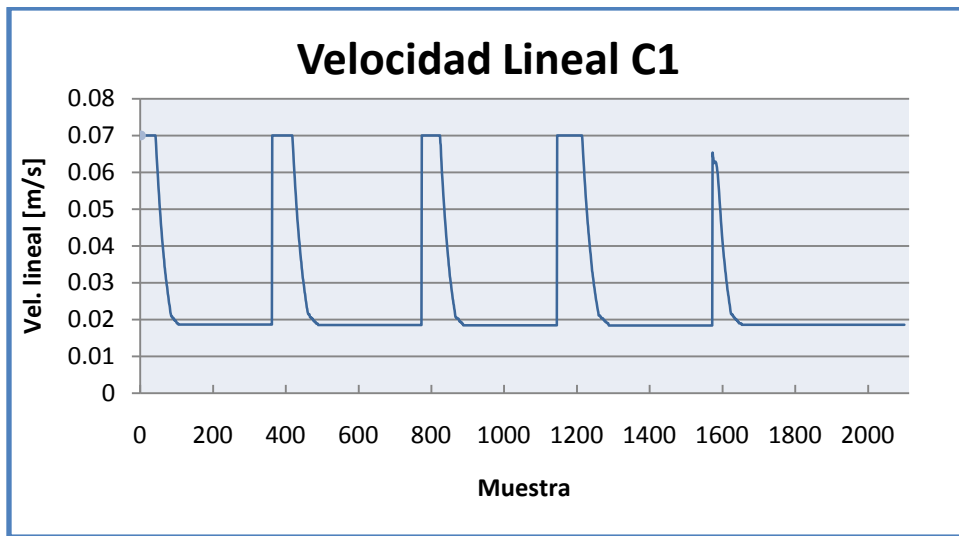


Figura 6. 8: Gráfica de la velocidad lineal del Cobot 1.

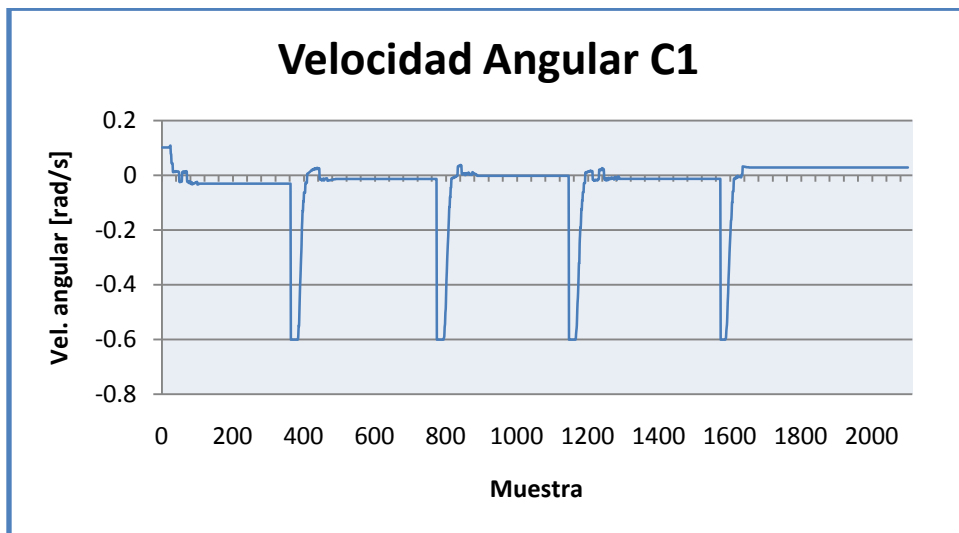


Figura 6. 9: Gráfica de la velocidad angular del Cobot 1.

A continuación se presentan las graficas de la trayectoria seguida por el Cobot 2 así como su velocidad lineal y angular.

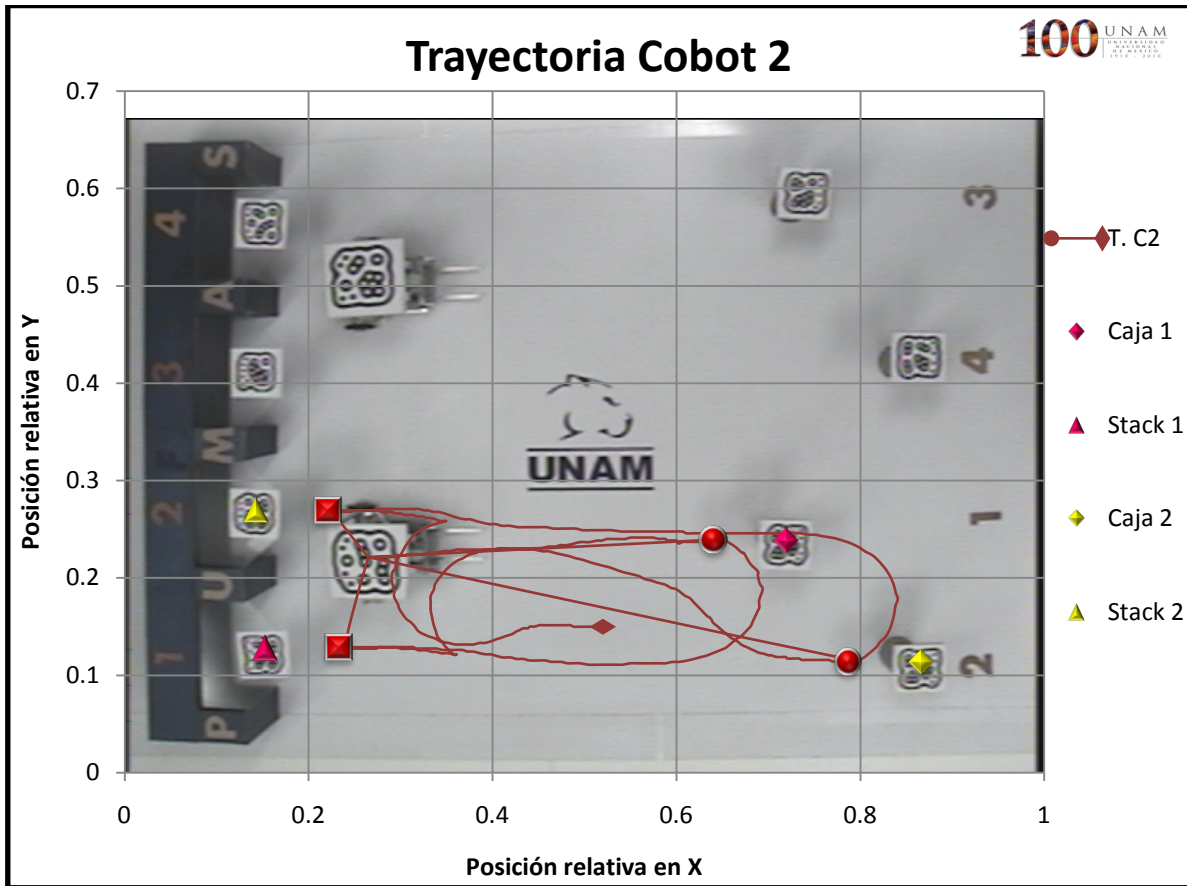


Figura 6. 10: Trayectoria completa del Cobot 2.

En la **Figura 6. 10** se puede observar la trayectoria seguida por el Cobot 2.

En la **Figura 6. 11** observamos la primera parte de la trayectoria la cual es la más corta y suave para ir de un punto a otro y realizar la tarea satisfactoriamente. Sin embargo en la **Figura 6. 12** observamos que la segunda parte de la trayectoria no es la más corta ni tan suave como se esperaba, el Cobot 2 se orientó para llegar en un ángulo de 180° tanto para recoger como para depositar la caja, esto se observa cuando el Cobot 2 llega al punto representado por un círculo rojo, el Cobot 2 giró un poco más de lo necesario a la derecha y posteriormente rectificó girando hacia la izquierda, esto pudo deberse a diversos factores como pueden ser, la visión, la retroalimentación para conocer la posición del Cobot 2 no fue lo suficientemente rápida o se perdieron algunos datos, o simplemente ese fue el resultado obtenido del modelo de control de campos potenciales.

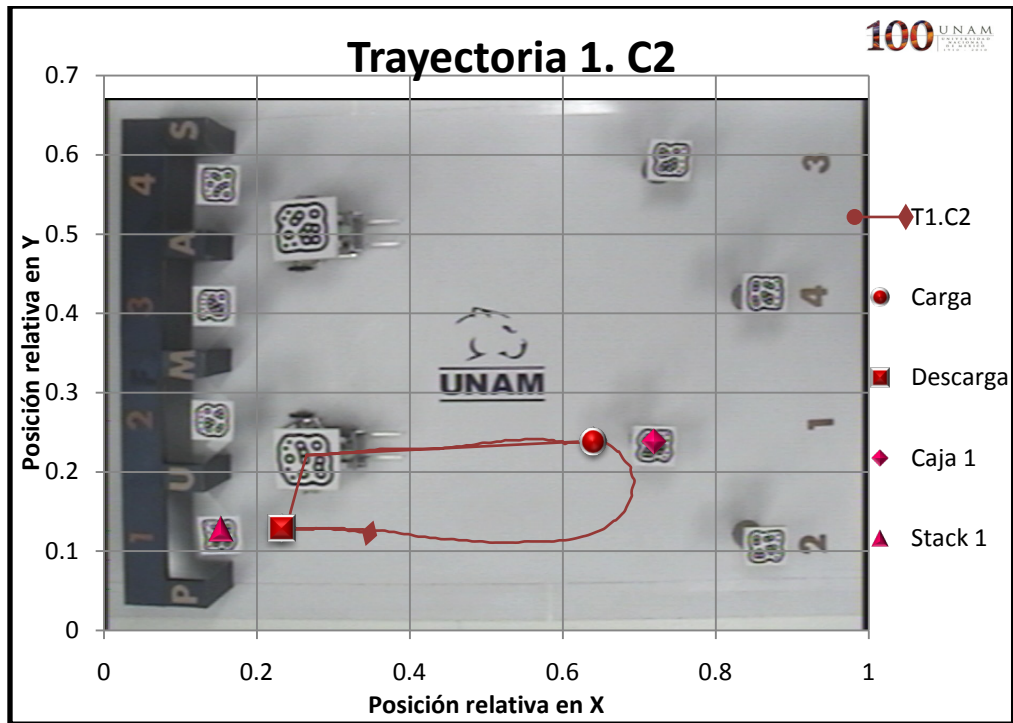


Figura 6. 11: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 1.

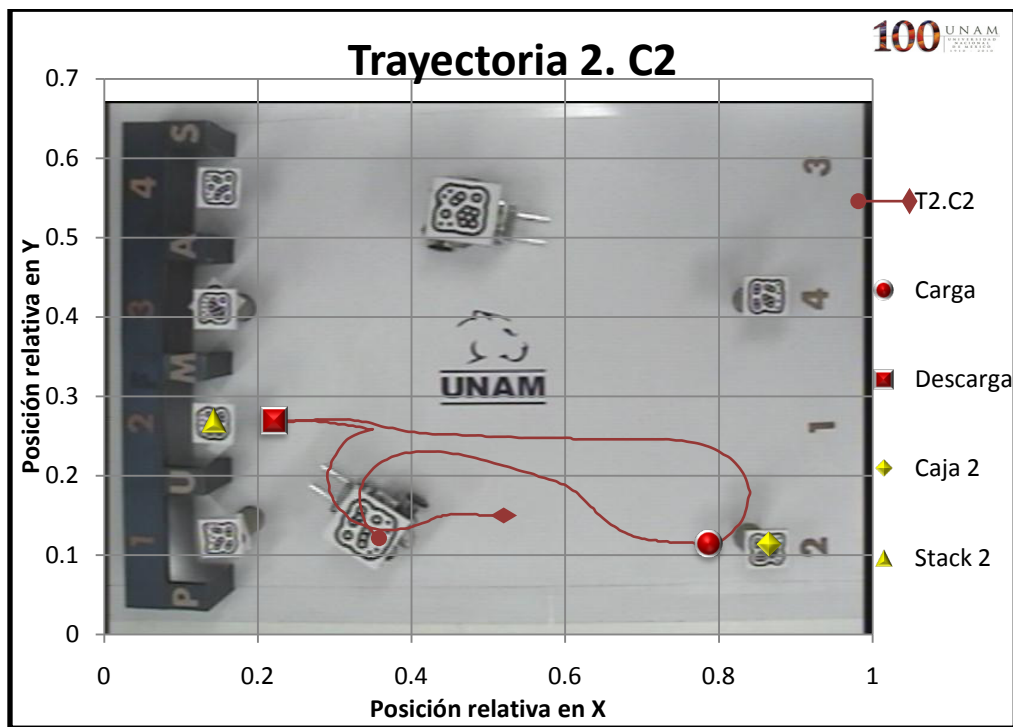


Figura 6. 12: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 2.

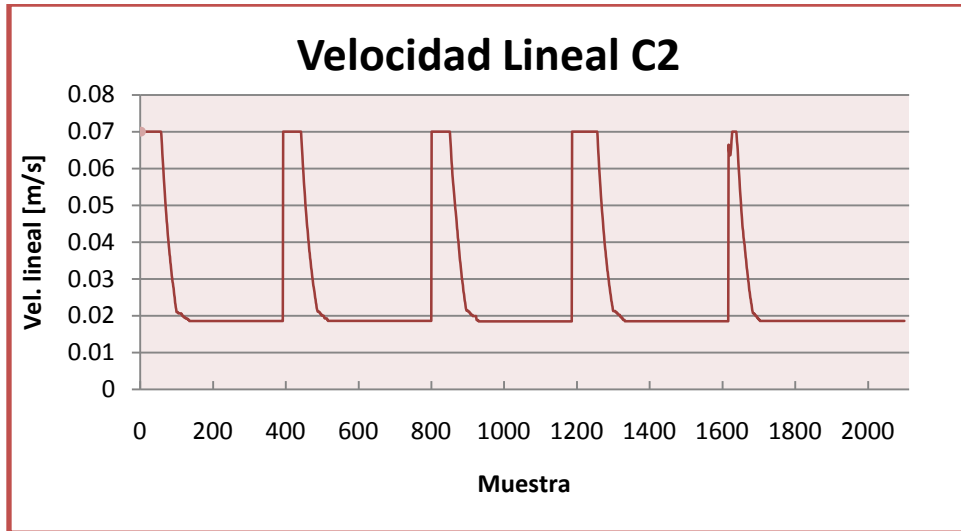


Figura 6. 13: Gráfica de la velocidad lineal del Cobot 2.

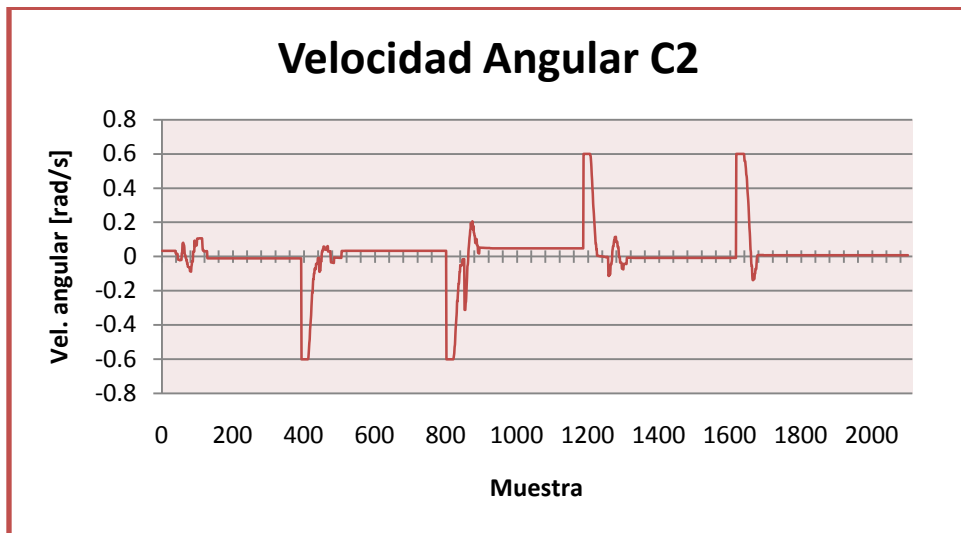


Figura 6. 14: Gráfica de la velocidad angular del Cobot 2.

En la **Figura 6. 13** se observa que el Cobot 2 va a su máxima velocidad lineal cuando se encuentra lejos de la meta y disminuye conforme se acerca a ésta. En la **Figura 6. 14** observamos que la trayectoria es lo más suave posible pero también se observa que, dentro de los puntos muestra 800-1000, hay un giro hacia la derecha y luego rectifica hacia la izquierda que representa cuando el Cobot 2 trató de orientarse para recoger la caja.

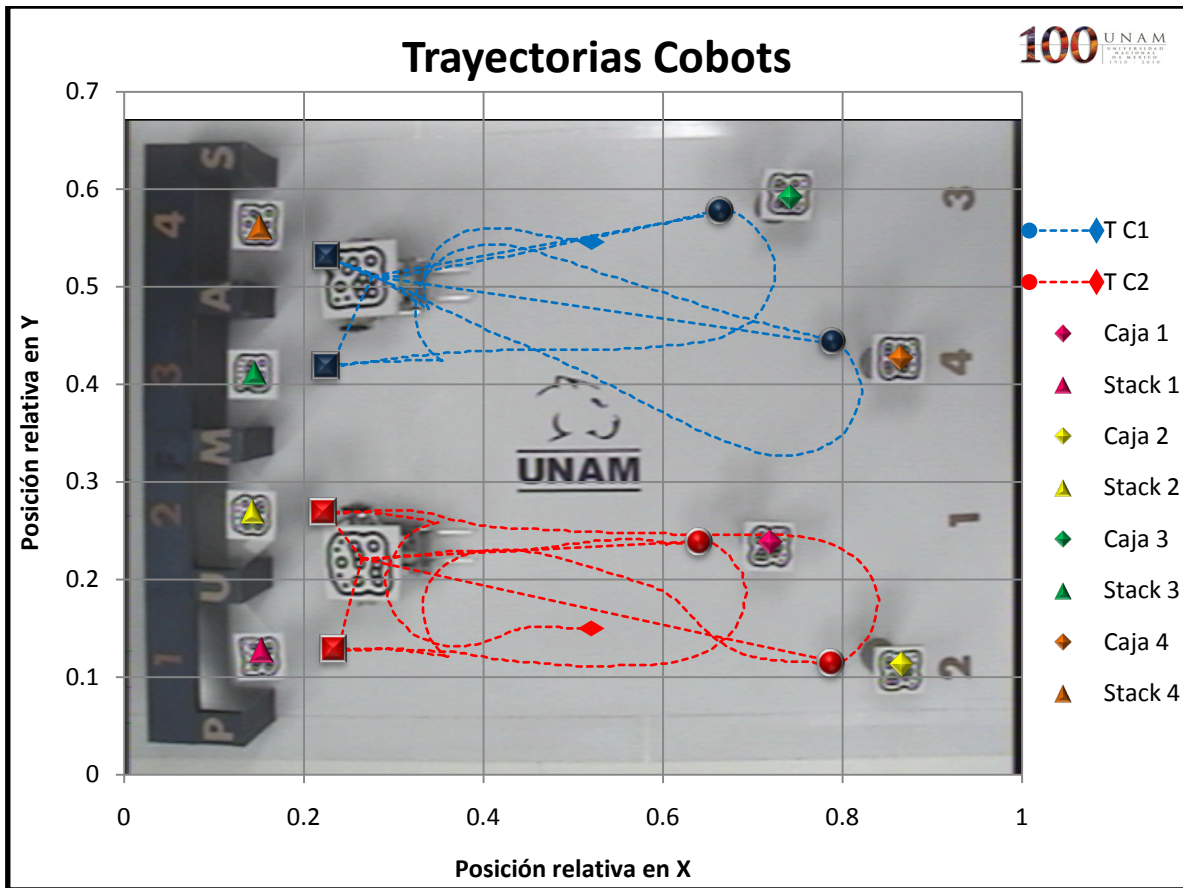


Figura 6. 15: Trayectoria de las Cobots.

En la **Figura 6. 15** se tienen las trayectorias seguidas por los cobots, se puede observar que se cumple con el objetivo de la prueba planteado anteriormente, esto es, los cobots fueron capaces de recoger y dejar las cajas en su lugar correspondiente, de la manera más rápida y simple. Los cobots nunca se cruzaron y por ello nunca se utilizaron campos repulsivos.

6.4.2 Prueba 2. Cruce de trayectorias.

El objetivo de esta prueba es que cada cobot traslade su caja evitando colisionar entre sí. Las cajas se posicionaron para que las trayectorias hacia sus “stacks” correspondientes se cruzaran con las del otro cobot y se apreciara la evasión de obstáculos descrita en capítulos anteriores.

Los cobots parten del reposo en la posición R1(0.24038,0.47288) y R2(0.24807,0.20632), las posiciones de las cajas son C1(0.92715,0.61252), C2(0.72670,0.41728), C3(0.85497,0.09224) y C4(0.78192,0.22294) y las posiciones de los stacks son S1(0.13167,0.11453), S2(0.12966,0.25640), S3(0.13153,0.39583) y S4(0.13691,0.54391), de acuerdo a la **Figura 6. 16**.

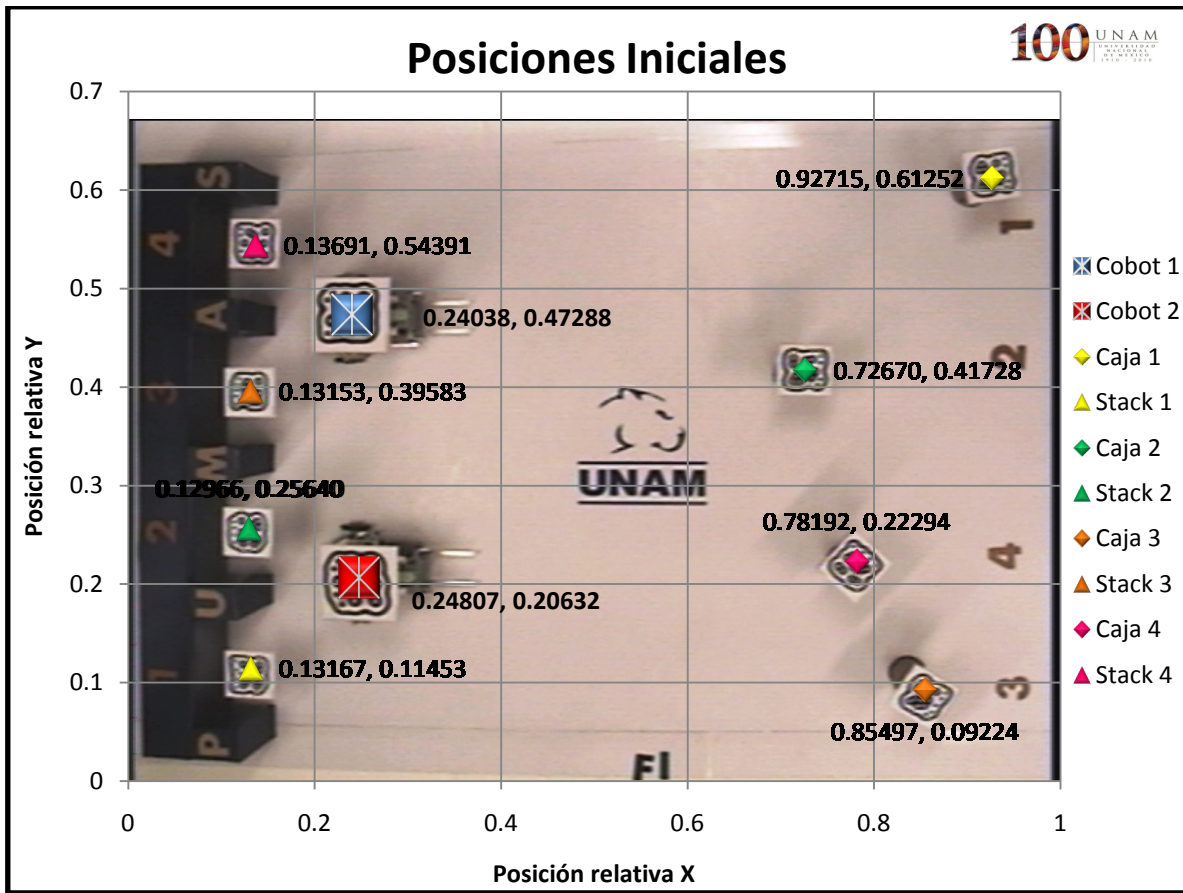


Figura 6. 16: Posiciones iniciales de los elementos que conforman la prueba.

A continuación se muestran las graficas de la trayectoria del Cobot 1 así como las graficas de su velocidad lineal y angular.

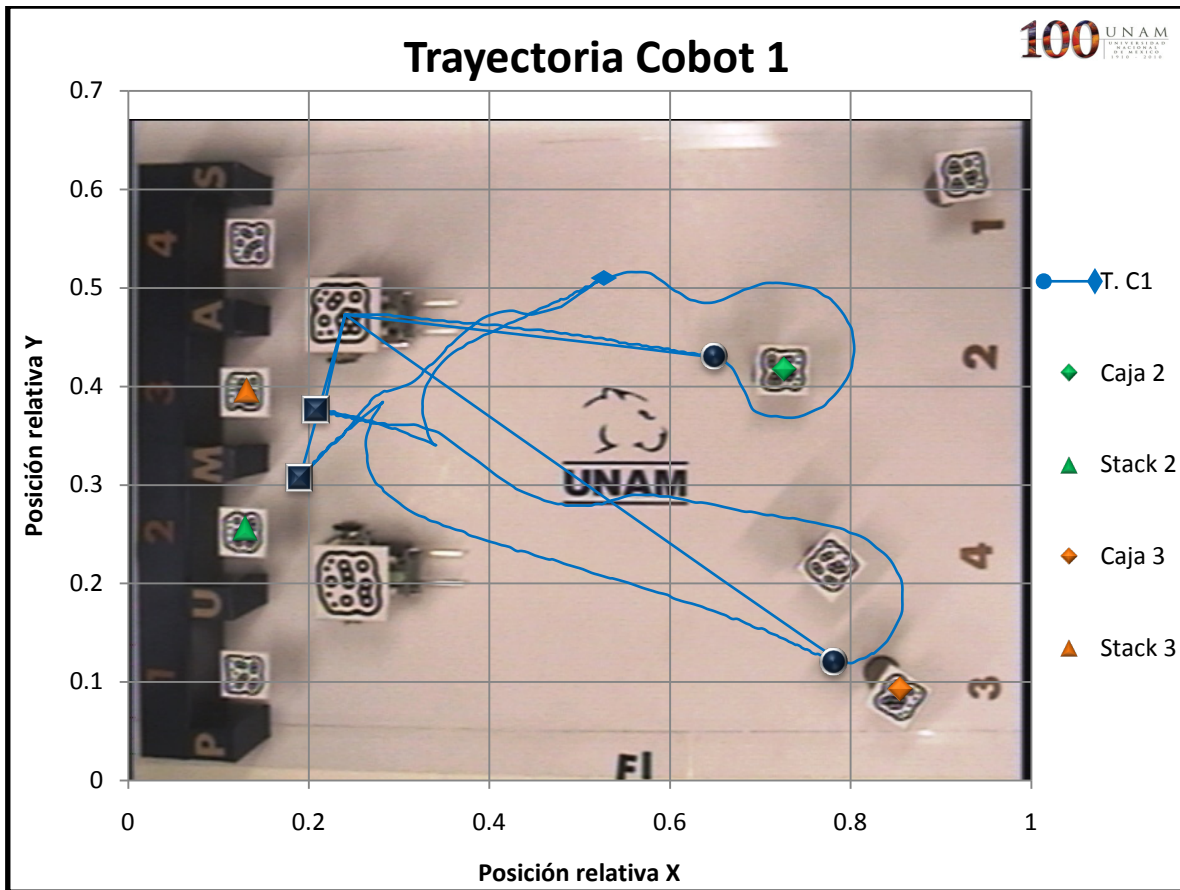


Figura 6. 17: Trayectoria completa del Cobot 1.

En la **Figura 6. 17** se observa la trayectoria completa del Cobot 1, al igual que en la prueba anterior se puede observar que la tarea fue realizada de la forma esperada de acuerdo a la teoría de campos potenciales artificiales y al algoritmo de asignación de tareas.

En la **Figura 6. 18** se representa la trayectoria seguida por el Cobot 1 para recoger y dejar en su correspondiente *stack* la caja 2. Podemos observar que la trayectoria seguida del inicio a que recoge la caja es lo mas recta posible como se planteó anteriormente; la trayectoria para dejar la caja, contrario a la primera parte, oscila, esto es debido a que el Cobot 2 interfiere en el camino del Cobot 1, es decir este último entra al campo de influencia del Cobot 2 que le genera una fuerza de repulsión causando que el Cobot 1 busque una nueva trayectoria para evitar colisionar entre sí. Una vez que el Cobot 1 sale del campo de influencia del Cobot 2, retoma el camino más corto a la meta.

En la **Figura 6. 19** se tiene la trayectoria seguida por el Cobot 1 para recoger y dejar en su *stack* la caja 3. Se puede observar que la trayectoria es la más corta y suave entre los puntos dados como meta y sólo existe una pequeña desviación en la trayectoria debido a que el Cobot 1 reaccionó porque se encontró dentro del campo de influencia del Cobot 2.

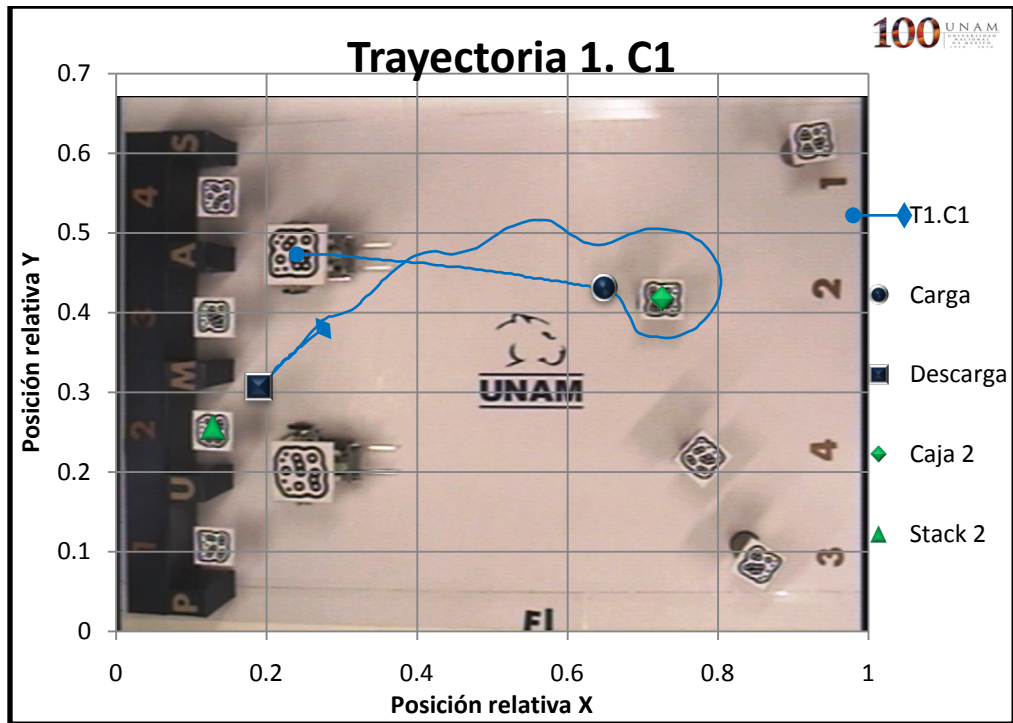


Figura 6. 18: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 2.

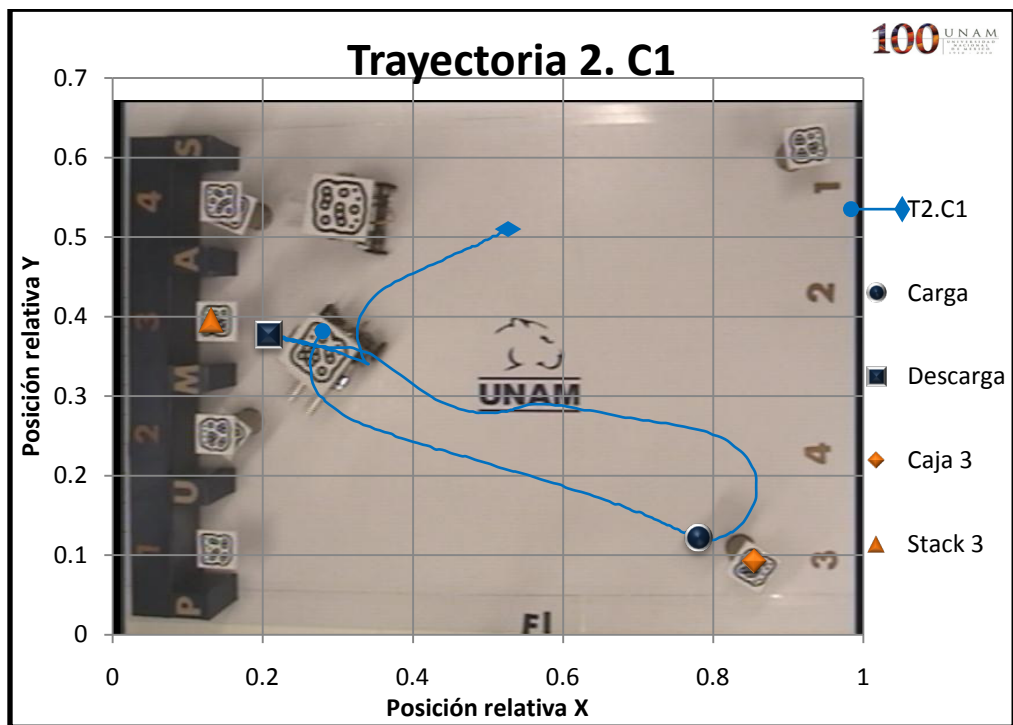


Figura 6. 19: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 3.

En la **Figura 6. 20** se observa que el Cobot 1 va a máxima velocidad cuando está lejos de la meta y disminuye su velocidad conforme se acerca. En la **Figura 6. 21** se observa, entre los puntos muestra 400-600, que el Cobot 1 oscila para evitar colisionar con el Cobot 2, los campos potenciales artificiales entran en acción; después su trayectoria es suave y vuelve a oscilar cuando se encuentra nuevamente dentro del campo de influencia del Cobot 2.

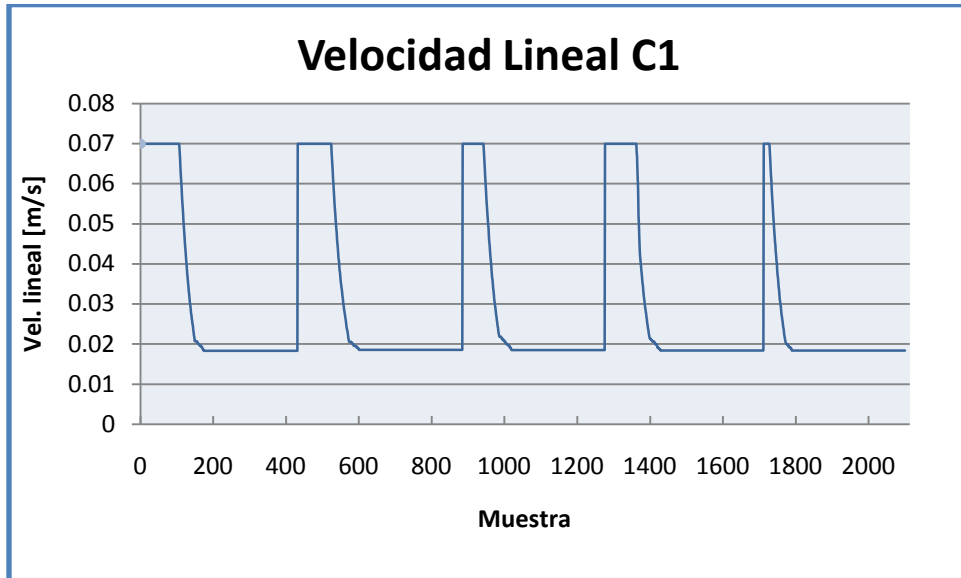


Figura 6. 20: Gráfica de la velocidad lineal del Cobot 1.

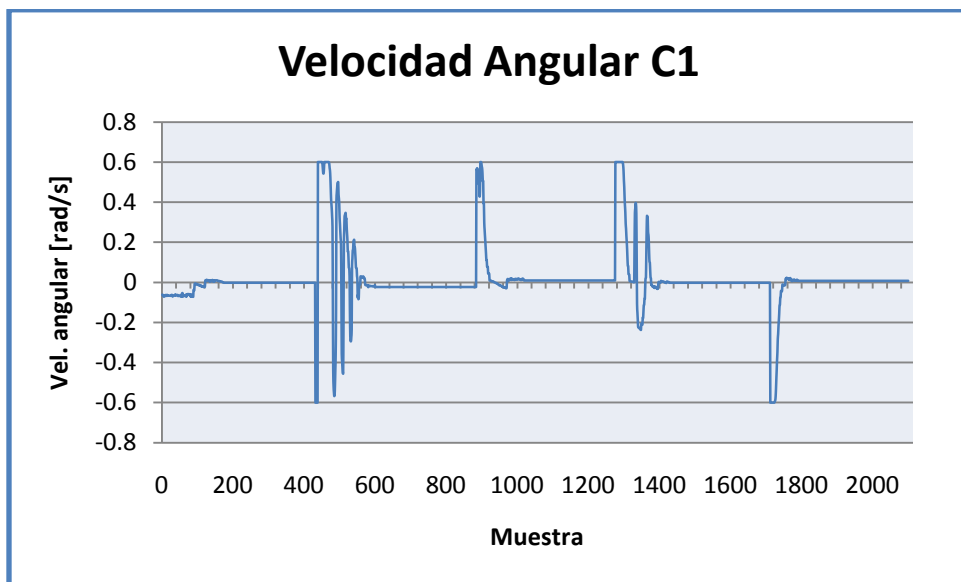


Figura 6. 21: Gráfica de la velocidad angular del Cobot 1.

A continuación se presentan las graficas de la trayectoria seguida por el Cobot 2 así como su velocidad lineal y angular.

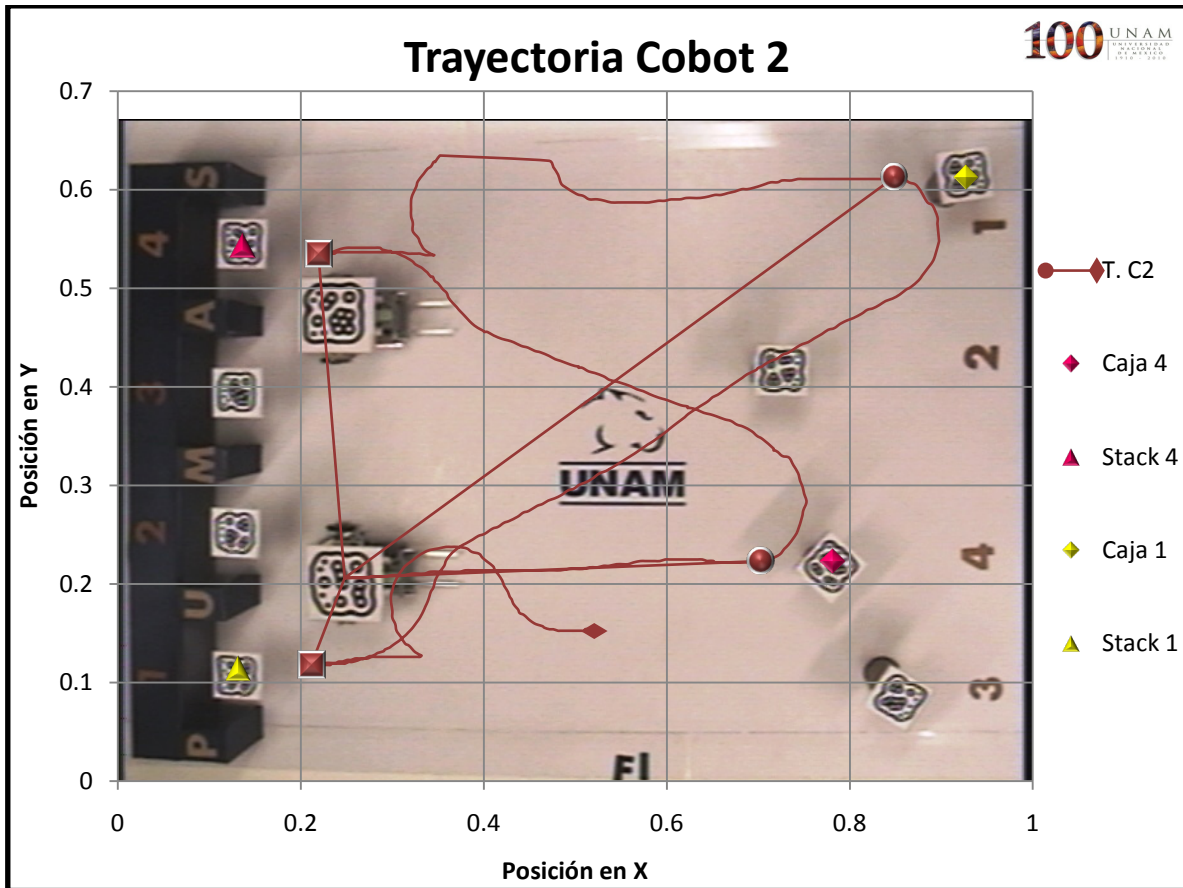


Figura 6. 22: Trayectoria completa Cobot 2.

En la **Figura 6. 22** se puede observar la trayectoria completa seguida por el Cobot 2 para realizar la tarea. La trayectoria tiende a ser la más corta y suave posible pero en algunos puntos el Cobot 2 cambia de dirección.

En la **Figura 6. 23** se observa la primera parte de la trayectoria la cual es la más corta y suave, solamente al momento de que el Cobot 2 se dirige a dejar la caja 4 a su lugar, éste gira para orientarse y colocarse lo más centrado posible al *stack*, En la **Figura 6. 24** se observa la segunda parte de la trayectoria, al principio se observa que el Cobot 2 gira en sentido de las manecillas, después avanza un lapso corto recto hacia la caja 1, gira nuevamente hacia la derecha y rectifica su curso para dirigirse a la caja 1, pero no fue así. Lo que sucedió fue que el Cobot 2 en el primer giro salió del área de visión y el software no obtenía la posición actual del Cobot 2, conservando la última y grificándola en lo que vemos como la línea recta. Éste continuó girando hacia la derecha sin recibir nuevas instrucciones hasta que finalmente regresó al área de visión y fue cuando el programa rectificó la trayectoria al conocer nuevamente su posición. Después de esto la trayectoria fue la más corta y suave posible y al igual que en la primera parte el Cobot 2 se orientó para colocarse lo más centrado posible para dejar la caja 1 en su *stack* correspondiente.

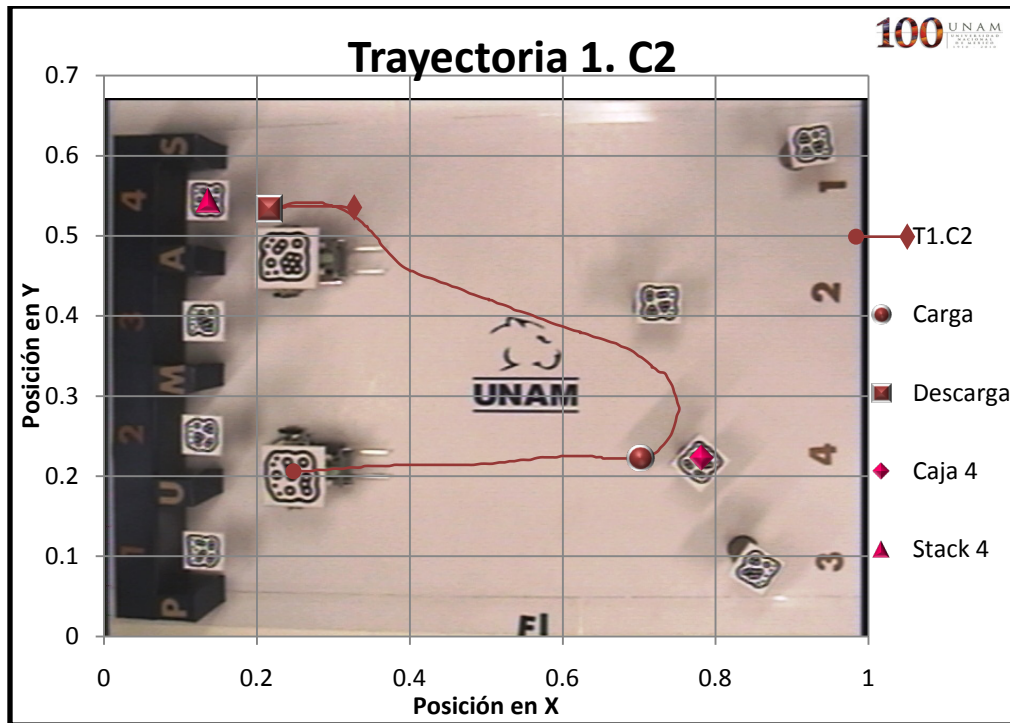


Figura 6. 23: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 4.

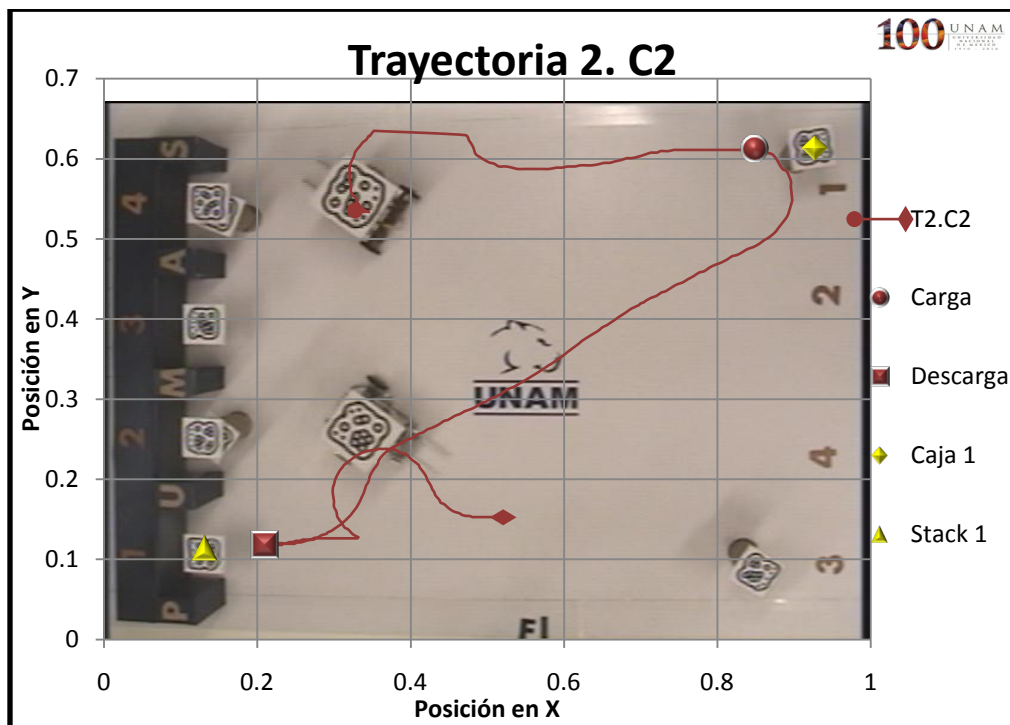


Figura 6. 24: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 1.

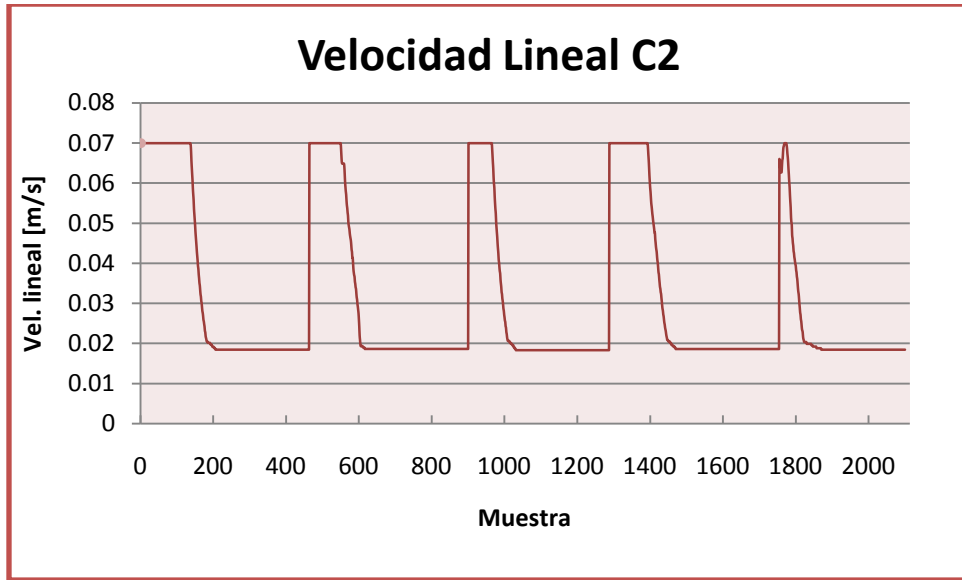


Figura 6. 25: Gráfica de la velocidad lineal del Cobot 2.

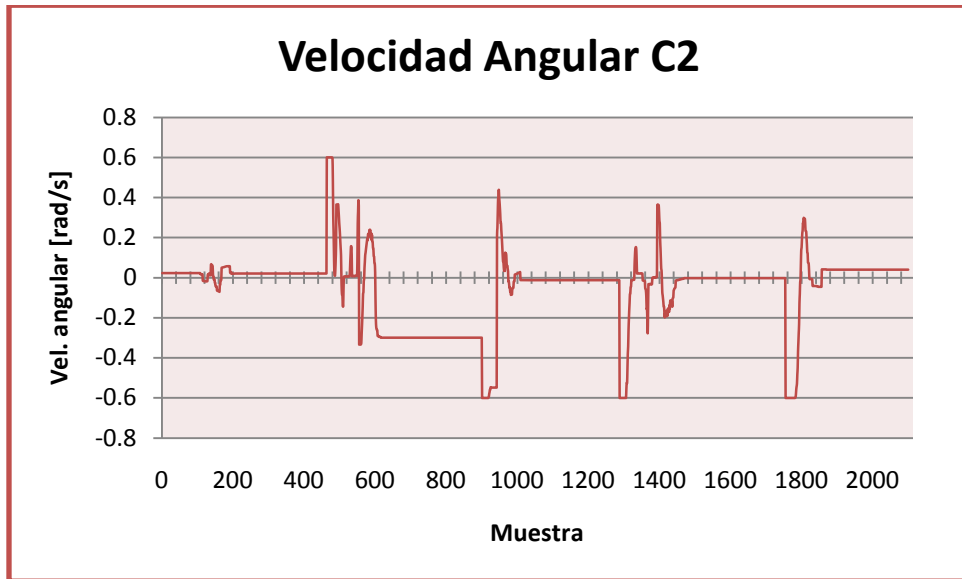


Figura 6. 26: Gráfica de la velocidad angular del Cobot 2.

En la **Figura 6. 25** se observa que la velocidad del Cobot 2 es máxima cuando se encuentra lejos de la meta y disminuye conforme se va acercando a ésta. En la **Figura 6. 26** se observa que el Cobot 2 tuvo que rectificar varias veces su trayectoria y la línea recta que se observa entre los puntos muestra 600-1000 es la parte en la que el Cobot 2 sale del área de visión de la cámara.

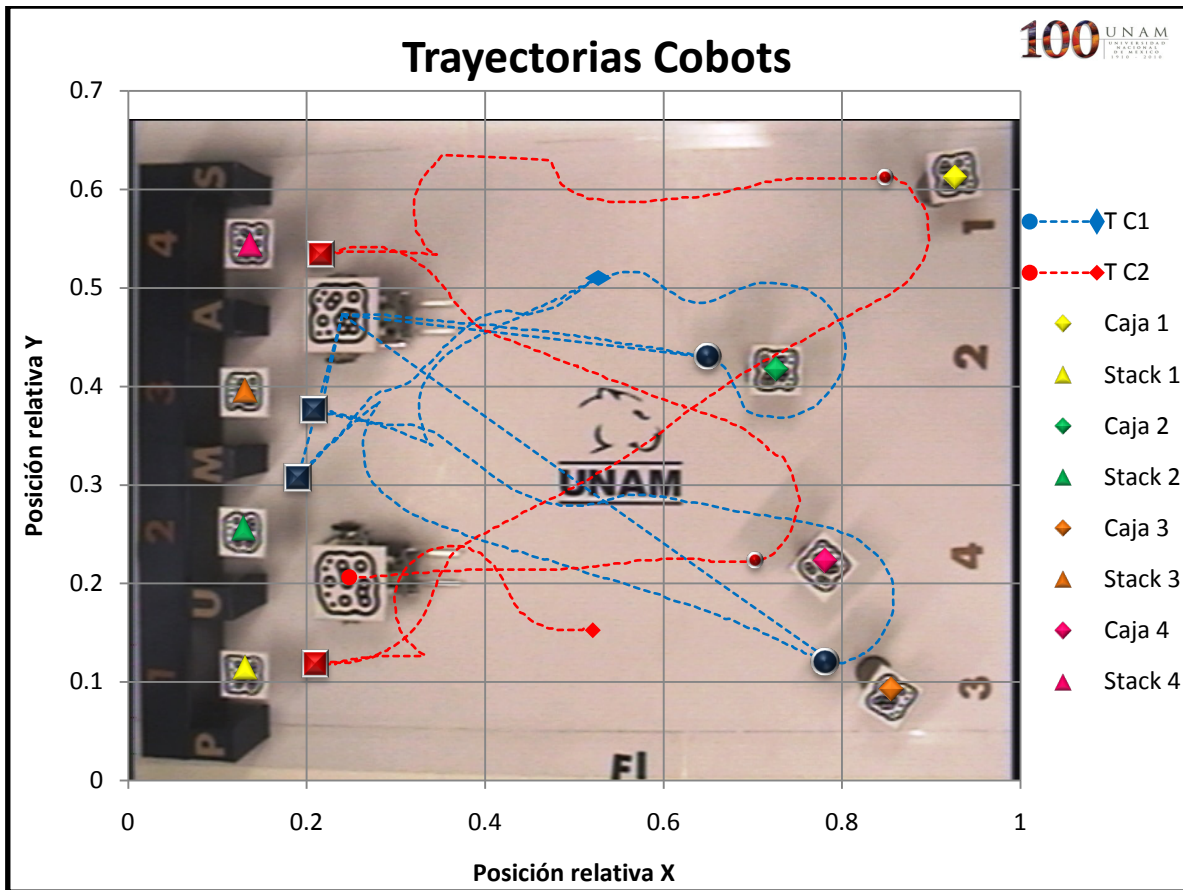


Figura 6. 27: Trayectorias de los Cobots.

En la **Figura 6. 27** se muestran juntas las trayectorias seguidas por cada uno de los Cobots para realizar la tarea. Se puede observar que la tarea se realizó de forma exitosa de acuerdo a lo planteado al inicio de la prueba, los cobots fueron capaces de evadirse a sí mismos para evitar colisionar, al igual que las trayectorias tendieron a ser las más cortas y suaves posibles cuando los campos repulsivos no tenían efecto sobre los Cobots.

6.4.3 Prueba 3. Jerarquía.

El objetivo de esta prueba es apreciar la jerarquía que tiene el cobot 1 sobre el cobot 2. Esto es, si la distancia a una caja es la menor para ambos o bien es la misma, el cobot 1 recogerá y trasladará dicha caja debido al algoritmo planteado.

Los cobots parten del reposo en la posición R1(0.25202,0.53118) y R2(0.28091,0.09897), las posiciones de las cajas son C1(0.87826,0.34282), C2(0.85483,0.09345), C3(0.83414,0.56780) y C4(0.68919,0.33190) y las posiciones de los stacks son S1(0.14354,0.11309), S2(0.14199,0.25480), S3(0.14388,0.39566) y S4(0.14852,0.54420).

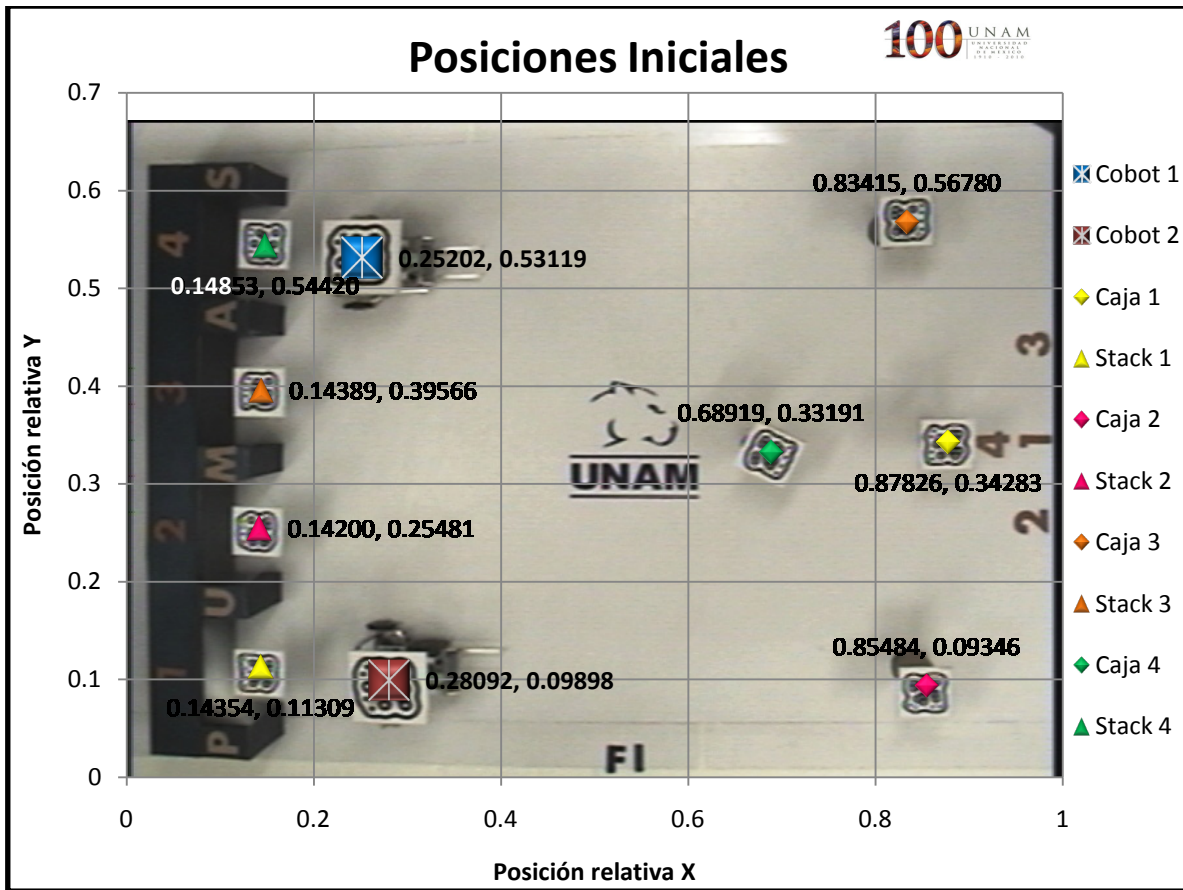


Figura 6. 28: Posiciones iniciales de los componentes que conforman la prueba.

A continuación se presentan las graficas de la trayectoria seguida por el Cobot 1 así como su velocidad lineal y angular.

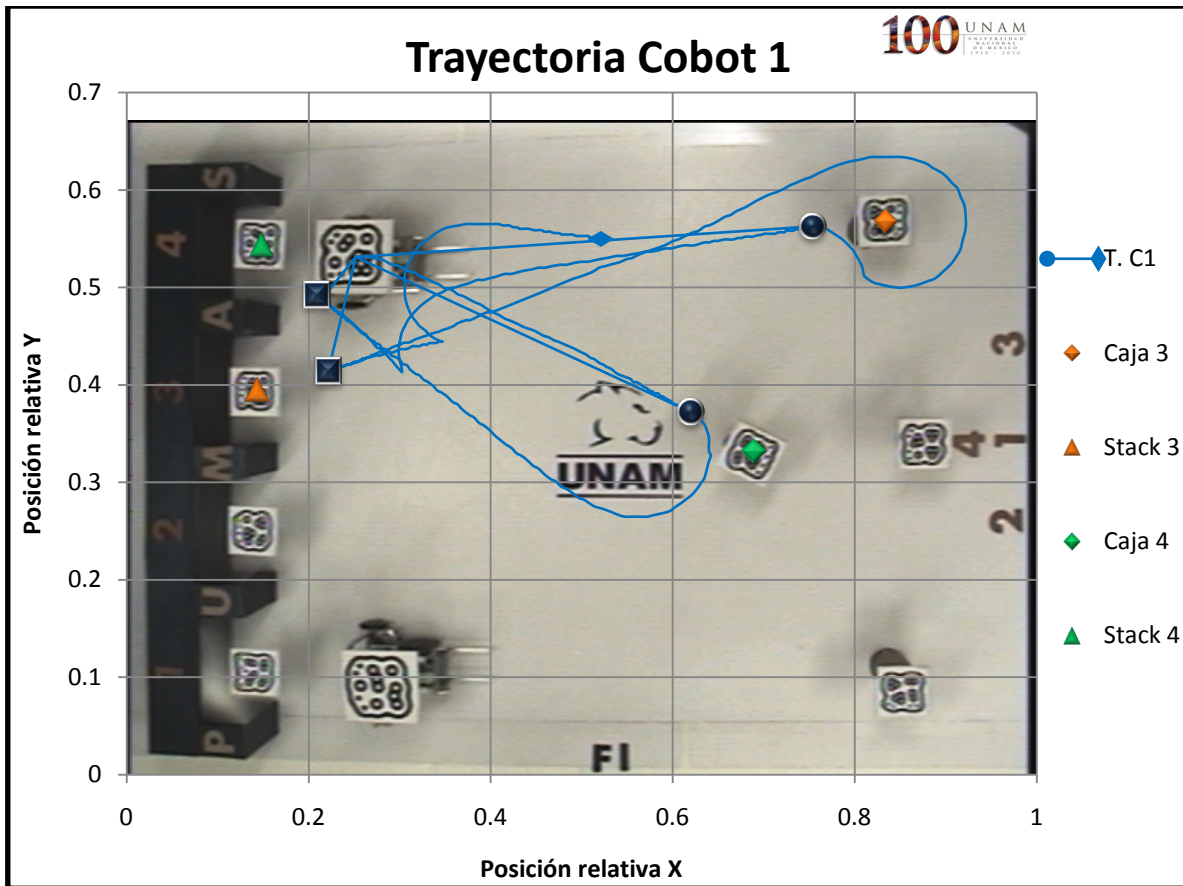


Figura 6. 29: Trayectoria completa seguida por el Cobot 1.

En la **Figura 6. 29** se tiene la trayectoria completa seguida por el Cobot 1, como se puede observar es la más corta entre punto y punto y también es muy suave.

En la **Figura 6. 30** se tiene la primera parte de la trayectoria y se puede observar que es la más corta y suave para realizar la tarea. En la **Figura 6. 31** se tiene la segunda parte de la trayectoria y al igual que en la primera, es muy suave. Sin embargo, después de recoger la caja 3 el Cobot 1 gira hacia la derecha y abruptamente rectifica y gira totalmente hacia la izquierda, esto fue debido a que detectó el campo de influencia del Cobot 2 y se creó una fuerza repulsiva. Lo apreciamos de mejor manera en la **Figura 6. 39**.

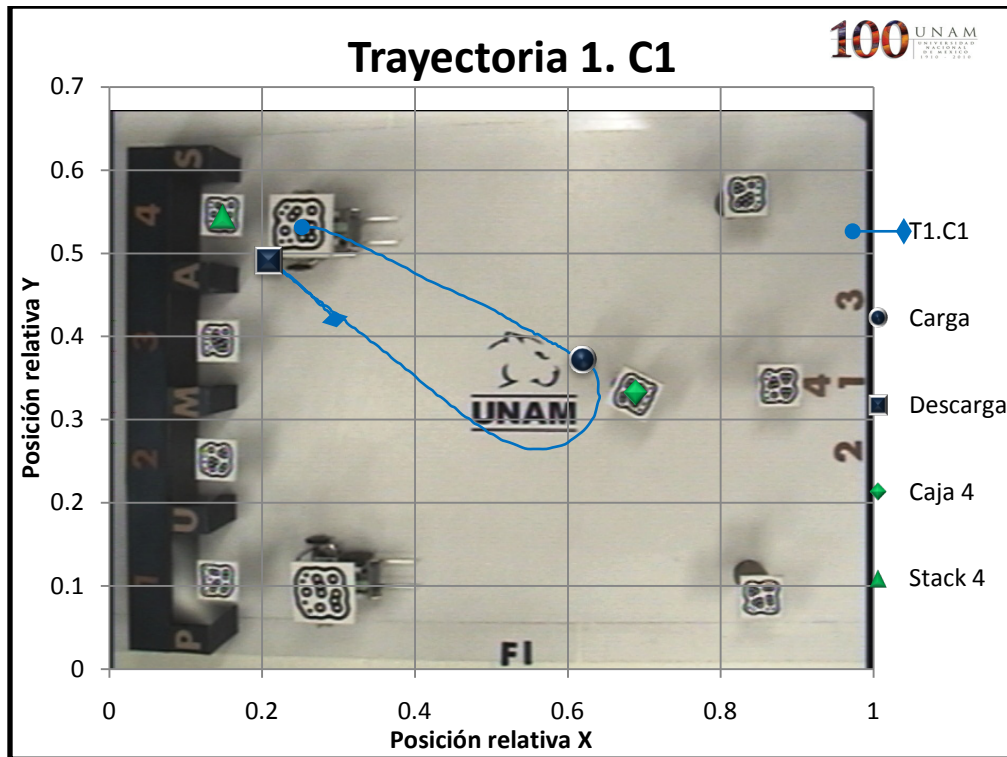


Figura 6. 30: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 4.

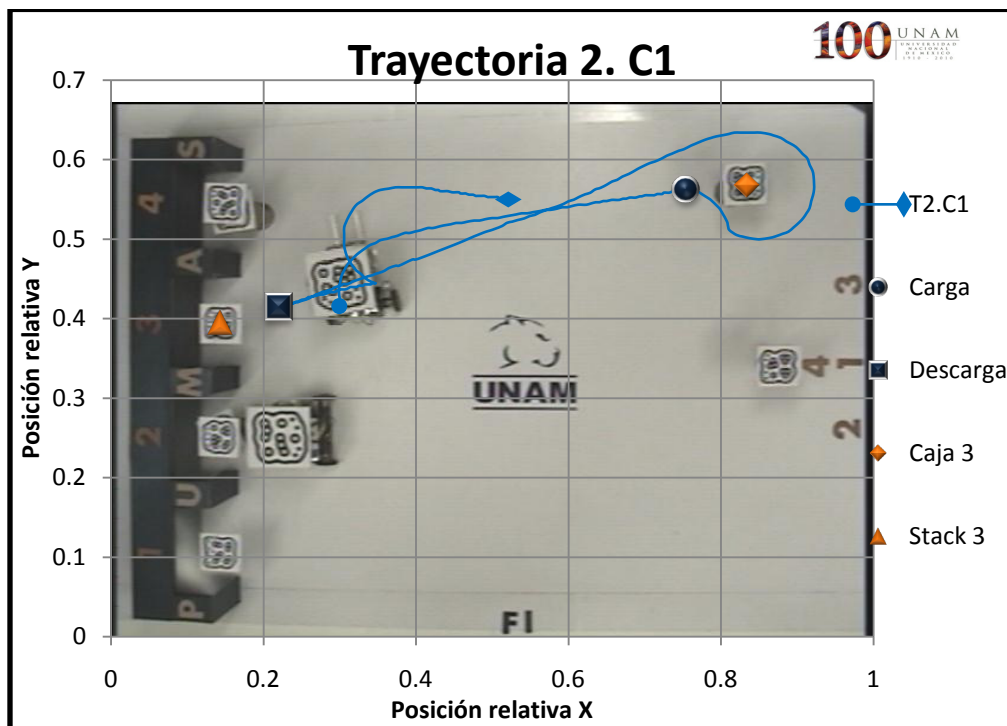


Figura 6. 31: Trayectoria seguida por el Cobot 1 para recoger y dejar la caja 3.

En la **Figura 6. 32** se observa que la velocidad lineal del Cobot 1 fue máxima cuando se encontraba lejos de su meta y fue disminuyendo conforme se acercaba a ésta. En la **Figura 6. 33** se tiene que la trayectoria seguida por el Cobot 1 fue muy suave, entre los puntos muestra 1200-1400 se observa un doble pico, como se explico anteriormente, fue cuando el Cobot 1 giro en sentido de las manecillas siguiendo el camino más corto a su meta local pero se encontró con el Cobot 2 el cual lo obligó a girar hacia el otro lado para evitar una colisión.

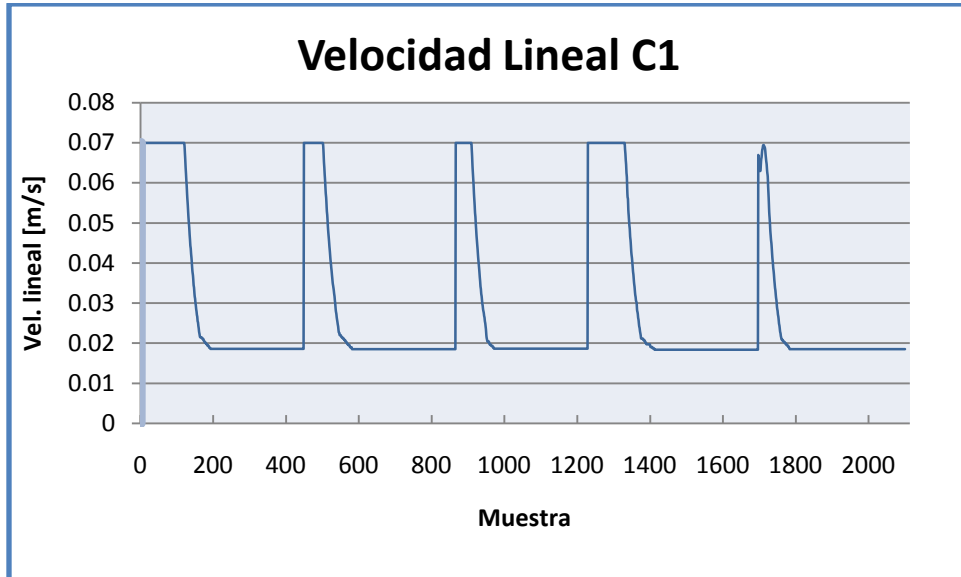


Figura 6. 32: Gráfica de la velocidad lineal del Cobot 1.

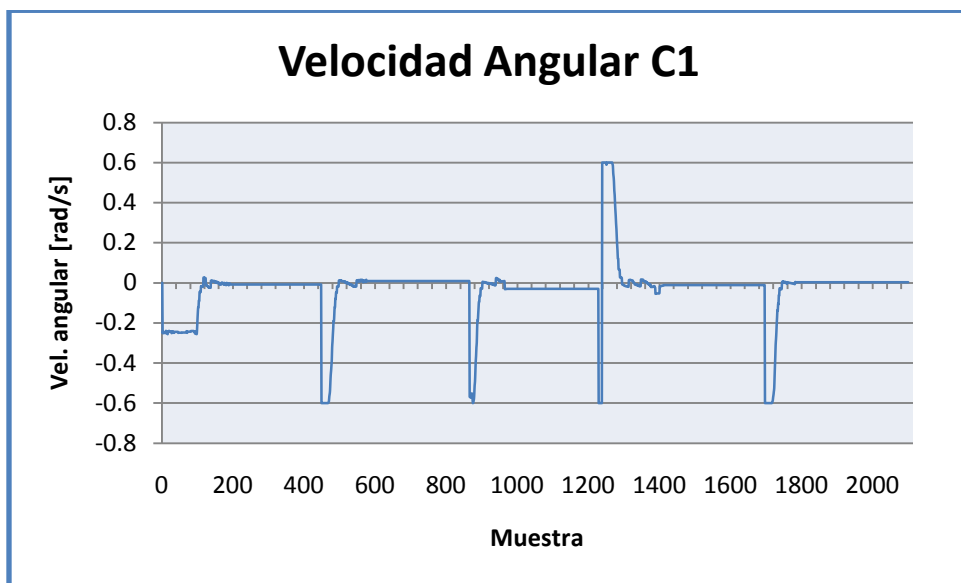


Figura 6. 33: Gráfica de la velocidad angular del Cobot 1.

A continuación se presentan las graficas de la trayectoria seguida por el Cobot 2 así como su velocidad lineal y angular.

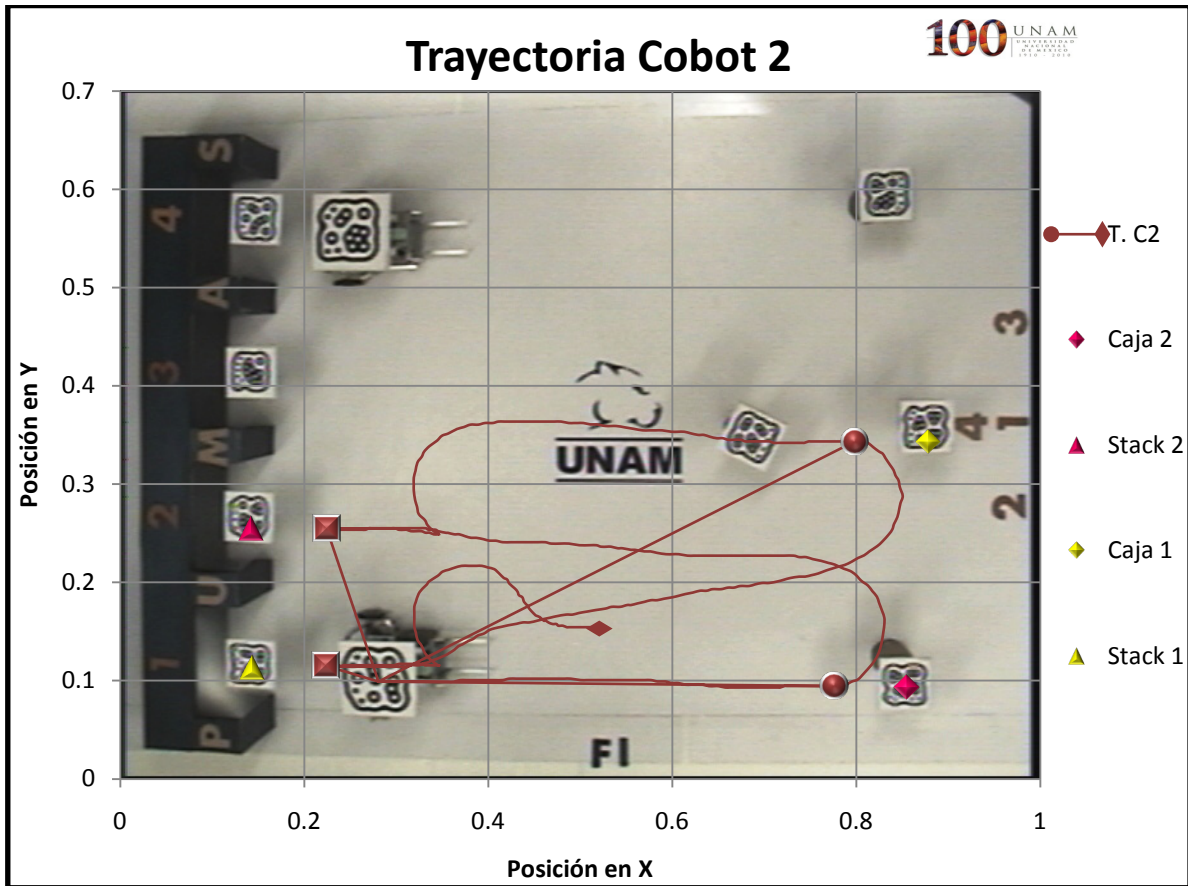


Figura 6. 34: Trayectoria completa del Cobot 2.

En la **Figura 6. 34** se tiene la trayectoria completa del Cobot 2, se puede observar que es muy suave. En la **Figura 6. 35** se tiene la primera parte de la trayectoria y en la **Figura 6. 36** la segunda parte, se puede observar que el Cobot 2 realizó la tarea de la forma más simple y siguiendo la trayectoria más corta entre punto y punto.

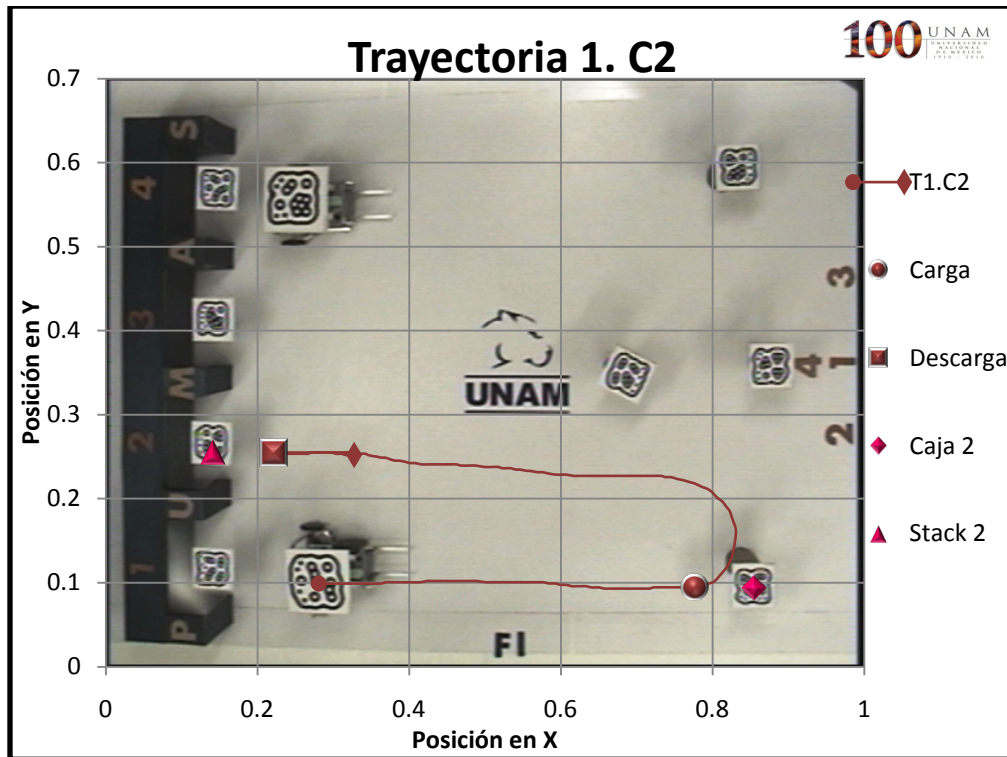


Figura 6. 35: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 2.

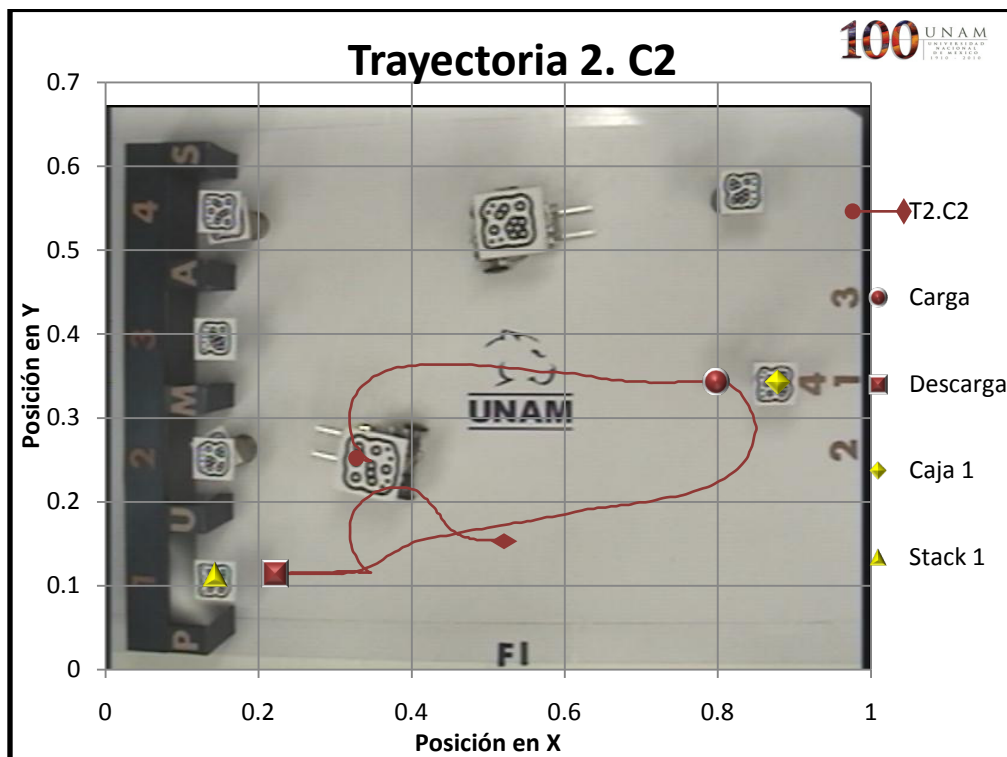


Figura 6. 36: Trayectoria seguida por el Cobot 2 para recoger y dejar la caja 1.

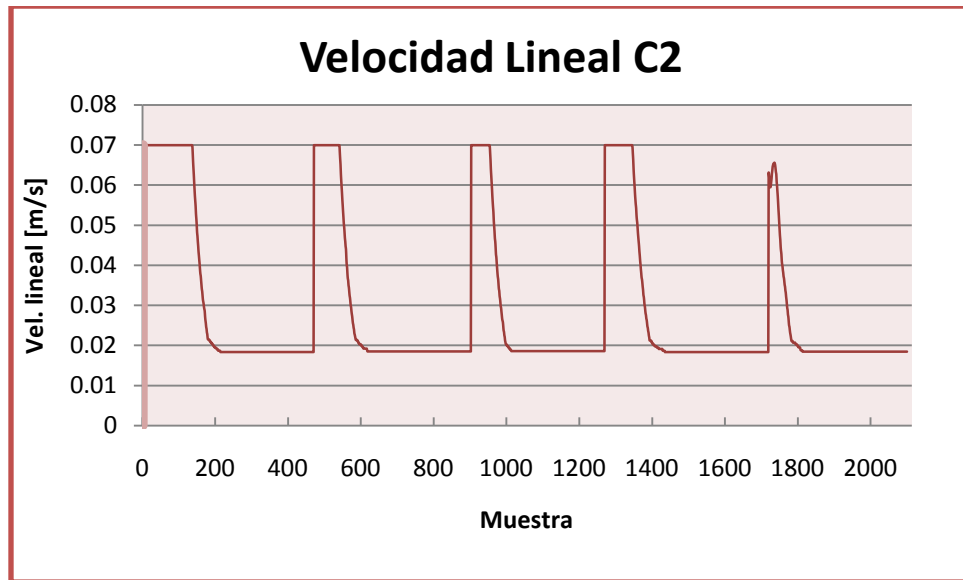


Figura 6. 37: Gráfica de la velocidad lineal del Cobot 2.

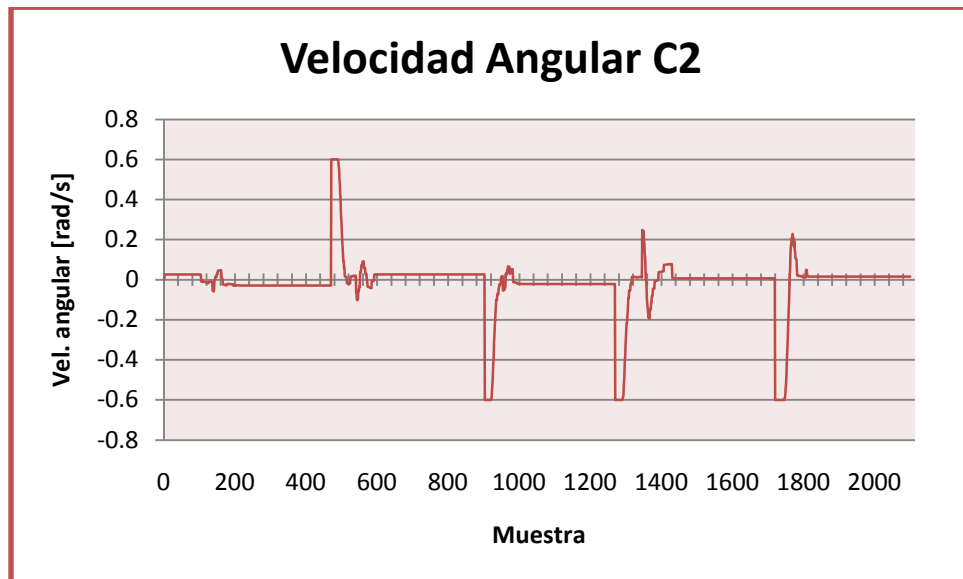


Figura 6. 38: Gráfica de la velocidad angular del Cobot 2.

En la **Figura 6. 37** se muestra la velocidad lineal la cual fue máxima cuando el Cobot 2 se encontraba alejado de la meta y fue disminuyendo conforme se acercaba a ésta. En la **Figura 6. 38** se tiene la velocidad angular y como se puede observa la trayectoria fue muy suave.

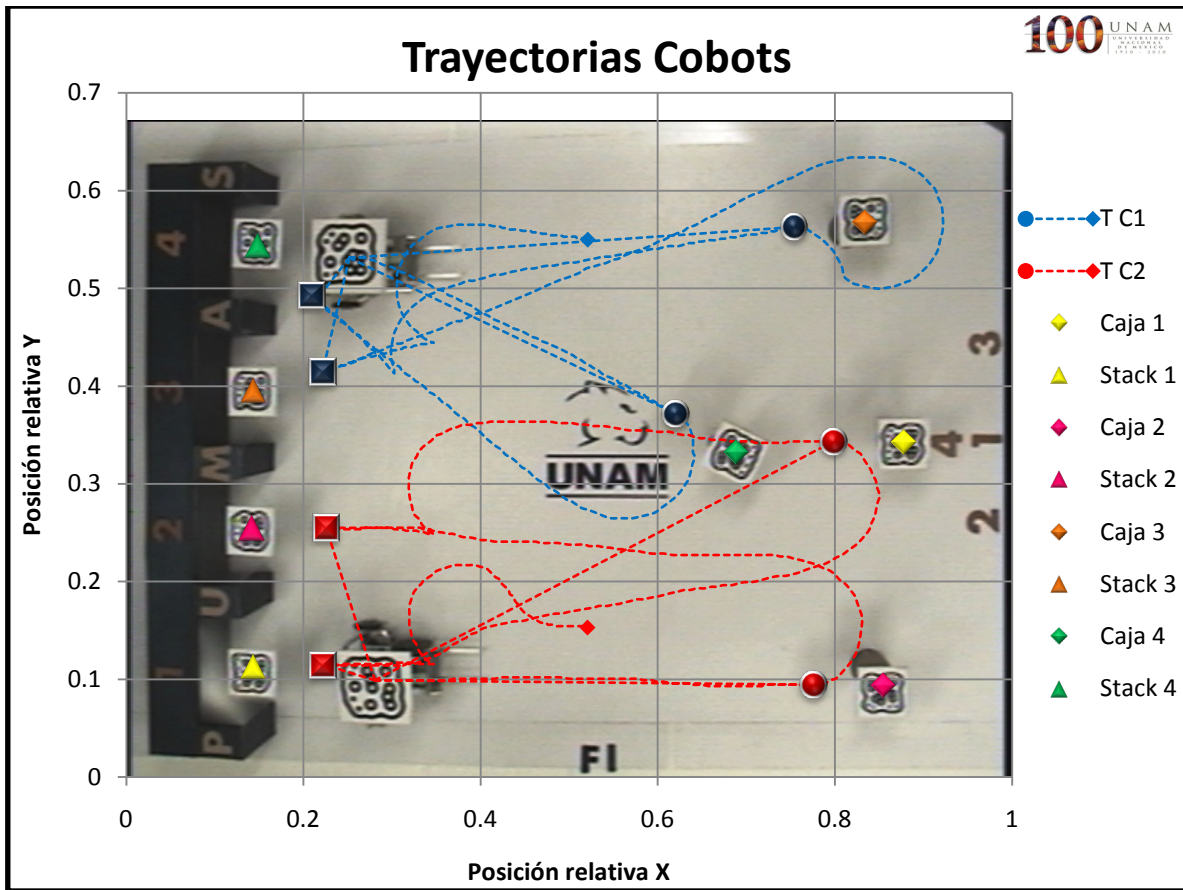


Figura 6. 39: Trayectoria completa de los Cobots.

La **Figura 6. 39** muestra las trayectorias seguidas por los Cobots para realizar la tarea. Como se planteó anteriormente, la distancia de la caja 4 a cada uno de los Cobots es la más cercana, por lo tanto y de acuerdo a la jerarquización dada a los Cobots, el algoritmo le asigna esa caja al Cobot 1 porque es el de mayor jerarquía y le asigna otra caja al Cobot 2, en este caso la 2. Se determinó de esta manera para evitar que los dos Cobots se dirijan hacia la misma caja y realicen la misma tarea.

También se observa como el Cobot 1 al recoger la caja 3 gira a la derecha y se encuentra dentro del campo de influencia del Cobot 2 por lo que cambia de sentido para dirigirse hacia el *stack* 3.

6.5 Análisis de Resultados

Las pruebas presentadas anteriormente fueron únicas e irrepetibles ya que la posición de cobots, cajas y cámara siempre variarían al preparar el entorno de las pruebas.

Se observa que los cobots fueron capaces de realizar la tarea de manejo de materiales sin alguna complicación, evitando colisionar entre sí, siguiendo la trayectoria más corta para recoger y dejar las cajas y respetando la jerarquía dada a los cobots.

Un punto importante es cuando los campos potenciales dejan de actuar sobre los cobots, esto sucede en el momento en que van de reversa después de dejar la caja en su correspondiente *stack*. Se determinó de esta manera porque si el modelo de control actuara en ese instante el cobot arrastraría la caja y colisionaría con su *stack*, para evitar esto deshabilitamos los campos potenciales y le indicamos al cobot que camine de reversa por 3 [s] y después de este tiempo el modelo toma de nueva cuenta el control sobre estos.

Por otra parte el Cobot 2 sólo está sujeto a fuerzas atractivas tanto de las cajas como de los *stacks* correspondientes, la fuerza repulsiva que le ejerce el Cobot 1 esta deshabilitada. Sin embargo, si el Cobot 2 entra al radio de influencia del Cobot 1 disminuirá su velocidad y se detendrá por completo hasta que la distancia sea mínima para evitar colisionar entre sí.

CAPÍTULO VII. CONCLUSIONES Y TRABAJO FUTURO

7.1 Conclusiones

En esta tesis se presentó un sistema semi-autónomo de robots móviles colaborativos para el manejo de materiales. Nuestro sistema se implementó en base a la teoría de campos potenciales artificiales para la navegación de los cobots, un sistema de visión para la localización de los cobots en el plano y un algoritmo de planificación de tareas los cuales trabajan conjuntamente de manera efectiva. Cabe mencionar que el análisis de la problemática en conjunto con las soluciones propuestas, dan la pauta para poder diseñar e implementar sistemas más sofisticados que respondan con mayor provecho a los requerimientos que pudieran plantearse en el futuro, esto se debe a que los algoritmos se diseñaron con la idea de migrar a otra plataforma robótica más flexible y completa.

Se observa de los resultados experimentales, que el funcionamiento en conjunto de los componentes del sistema fue altamente satisfactorio, esto se deriva del hecho de que todos los elementos diseñados e implementados funcionaron de la manera esperada, generando una respuesta adecuada para los parámetros establecidos, todos los elementos aquí planteados fueron implementados y probados con éxito.

Los cobots fueron capaces de completar la tarea favorablemente, la cual fue recolectar un material asignado a cada uno y transportarla a un lugar determinado, evitando colisionar entre sí. Los robots se colocaron en un ambiente desconocido y dinámico y su comportamiento fue el esperado de acuerdo a lo programado.

Se corroboró que la navegación de los cobots controlada por campos potenciales artificiales es un método eficiente para entornos dinámicos, en este caso el objetivo es estático pero siempre hay un obstáculo en movimiento. Se enriqueció el modelo con una modificación de los campos potenciales artificiales del Cobot 2, con el cual se mejoró el comportamiento de los cobots ya que se evitaron mínimos locales y a la vez se instauró una jerarquización reflejándose en una arquitectura preestablecida (GOFER).

Debido a que las actuales técnicas típicamente no son escalables a un gran número de robots (*e.g.* miles), nuestro sistema se diseñó para garantizar la escalabilidad de forma que los recursos de cómputo fueran mínimos centralizando la planificación, navegación y comunicación en una única unidad central.

Por otra parte la comunicación inalámbrica entre la PC y los cobots, bajo el protocolo ZigBee, trajo consigo un menor uso de recurso de cómputo, ahorro de energía y sobre todo no se presentaron errores de comunicación, pérdida de datos o interferencia.

Como experiencia fue muy gratificante llevar un proyecto desde su conceptualización hasta la implementación de un prototipo funcional. Nos permitió desarrollar un proyecto que integró los conocimientos adquiridos a lo largo de nuestra carrera, además de permitirnos culminar nuestra formación como Ingeniero Mecatrónico de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, de la cual nos sentimos sumamente orgullosos.

7.2 Trabajo futuro

Para el trabajo a futuro es posible cambiar el software de visión por uno que no implique tener marcadores de identificación, como lo es *Swistrack* [33]. De igual manera, poder trabajar en un área más extensa y con más robots a cargo. Los Cobots podrán ser modificados con una nueva rueda loca para un mejor desplazamiento así como un nuevo efector y actuador para el buen desempeño del montacargas.

De igual forma quedan muchas cuestiones abiertas en problemas de planificación de caminos y coordinación.

Debido a que los sistemas multi-robot operan en estocásticos e impredecibles configuraciones, el estudio de la interacción dinámica en estas configuraciones puede llevar a descubrimientos de alto impacto en amplios rangos de complejos sistemas no-lineales.

APÉNDICES

Apéndice A – Código programa C#

```
////////////////////////////////////  
//CODIGO PARA LA ADQUICION DE DATOS DE LA POSICION DE LOS COBOTS Y //  
// PLANIFICACION DE LA TAREA A REALIZAR POR LOS COBOTS //  
//DESARROLLADO POR: //  
//ALFREDO MIGUEL ANGELES GARCIA //  
//DANIEL LIMA ROBLEDA //  
//TESIS: SISTEMA SEMI-AUTONOMO DE ROBOTS MOVILES COLABORATIVOS //  
//PARA EL MANEJO DE MATERIALES //  
//FACULTAD DE INGENIERIA, UNAM 2010 //  
//DEPARTAMENTO DE INGENIERIA MECATRONICA //  
////////////////////////////////////  
  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Text;  
using System.Windows.Forms;  
using System.Collections;  
using System.IO.Ports;  
using TUIO;  
using Excel = Microsoft.Office.Interop.Excel;  
using System.Reflection;  
  
namespace Cobots  
{  
    public partial class Cobots : Form, TuioListener  
    {  
        SerialPort PuertoSerial;  
        SerialPort PuertoSerial1;  
        bool puertoAbierto = false;  
        Excel.Application AplicacionExcel; //Declaración de aplicación Excel.  
        private TuioClient client; //cliente de Reactivision.  
/*****VARIABLES*****/  
        Int64 nreg; //Contador de registros  
        int contador = 0;  
        int contador2 = 0;  
        int motores = 0xFC;  
        int motores2 = 0xFC;  
        int motora, motorb;  
        int motora2, motorb2;  
        float Xrob; //variables de posición del Cobot 1  
        float Yrob;  
        float Arob;  
        float Xrob2; //variables de posición del Cobot 2  
        float Yrob2;  
        float Arob2;  
        float Xgol; //variables de posición de meta para Cobot 1 y 2
```

```

float Ygol;
float Agol;
float Xgol2;           //variables de posición de meta para Cobot 1 y 2
float Ygol2;
float Agol2;
float Xgol3;           //variables de posición de meta para Cobot 1 y 2
float Ygol3;
float Agol3;
float Xgol4;           //variables de posición de meta para Cobot 1 y 2
float Ygol4;
float Agol4;
float Xgol5;           //variables de posición de meta para Cobot 1 y 2
float Ygol5;
float Agol5;
float Xgol6;           //variables de posición de meta para Cobot 1 y 2
float Ygol6;
float Agol6;
float Xgol7;           //variables de posición de meta para Cobot 1 y 2
float Ygol7;
float Agol7;
float Xgol8;           //variables de posición de meta para Cobot 1 y 2
float Ygol8;
float Agol8;
float XgolC1;          //variables arreglo de posición de la meta final C1
float YgolC1;
float XgolC2;          //variables arreglo de posición de la meta final C2
float YgolC2;
//Banderas para Distancias, Metas y Ciclos
bool registra = false;
bool metaflag = true;   //Bandera para la asignación de cajas del Cobot 1.
bool metaflag2 = true;  //Bandera para la asignación de cajas del Cobot 2.
bool metafflag = false; //Bandera para el descargue de cajas Cobot 1.
bool metafflag2 = false; //Bandera para el descargue de cajas Cobot 2.
bool meta1 = true;
bool meta2 = true;
bool meta3 = true;
bool meta4 = true;
bool metafM1C1= false;
bool metafM2C1= false;
bool metafM3C1= false;
bool metafM4C1= false;
bool metafM1C2= false;
bool metafM2C2= false;
bool metafM3C2= false;
bool metafM4C2= false;
bool ciclo = false;
bool ciclo2 = false;
//Variables para el cálculo de mínimas distancias.
double d1m1;
double d1m2;
double d1m3;
double d1m4;
double d2m1;
double d2m2;
double d2m3;
double d2m4;
//Para campos potenciales y salidas de control COBOT 1.
double atrax, atray, anguloatr; //Variables de atracción.
double repyx, repy;           //Variables de repulsión.

```

```

double errorangulo; //Variable para calcular el error ángulo.
double dg, dobs, dist; //Dist a su meta,dist al obstáculo y dist a meta de C2.
double vellinctrl; //Velocidad lineal de control.
double velangctrl; //Velocidad angular de control.
double wi, wd; //Velocidad angular de las ruedas izquierda y derecha.
//Para campos potenciales y salidas de control COBOT 2.
double atrax2, atray2, anguloatr2; //Variables de atracción.
double repx2, repy2; //Variables de repulsión.
double disx2, disy2; //Variables para calcular dist2.
double errorangulo2; //Variable para calcular el error ángulo.
double dg2, dobs2,dist2; //Dist a su meta,dist al obstáculo y dis a meta de C1.
double vellinctrl2; //Velocidad lineal de control.
double velangctrl2; //Velocidad angular de control.
double wi2, wd2; //Velocidad angular de las ruedas izquierda y derecha.
//Arreglo del registro Xrob, Yrob, Arob, vellinctrl, velangctrl, Xgol, Ygol, Xobs, Yobs.
double[,] registros = new double[2101, 14];
//Para las fuerzas.
double dis = 0.1;
double Kr = 0.3; //Constante de radio de influencia atracción.
double E1 = 0.5, E2 = 1; //Constantes de atracción.
double Ki = 0.20; //Constante del radio de influencia repulsión.
double Nu = .15; //Constante de repulsión.
double Wmax = .6; //Velocidad angular máxima.
double Vmax = 0.07; //Velocidad lineal máxima.
double b = 0.177; //Distancia entre las ruedas.
double c = 0.059; //Radio de la rueda.
double kprop = 21; //Constante de control proporcional.
/*****CONFIGURACION DE PUERTOS*****/
public Cobots(int port)
{
    InitializeComponent();
    //Conectar la forma con Reactivision
    client = new TuioClient(port);
    client.addTuioListener(this);
    client.connect();
    //Carga el combo box con los puertos encontrados
    busca_puertos(cmbPuerto);
    busca_puertos(cmbPuerto2);
    //Inicializamos velocidad
    cmbVel.Items.Add("1200");
    cmbVel.Items.Add("2400");
    cmbVel.Items.Add("4800");
    cmbVel.Items.Add("9600");
    cmbVel.Items.Add("19200");
    cmbVel.Items.Add("57600");
    cmbVel.Items.Add("115200");
    cmbVel.Text = "115200";
    cmbVel2.Items.Add("1200");
    cmbVel2.Items.Add("2400");
    cmbVel2.Items.Add("4800");
    cmbVel2.Items.Add("9600");
    cmbVel2.Items.Add("19200");
    cmbVel2.Items.Add("57600");
    cmbVel2.Items.Add("115200");
    cmbVel2.Text = "115200";
}
private void busca_puertos(ComboBox cb)
{
    // Obtenemos los puertos que se encuentran configurados en el equipo

```



```

string[] ports = SerialPort.GetPortNames();
cb.Items.Clear();
// Los desplegamos en el ComboBox
foreach (string port in ports)
{
    cb.Items.Add(port);
}
cb.Text = "Seleccione un puerto";
}
private void configura_puerto(string npuerto, int baudios, Parity paridad, int
nBits, StopBits bits_parada)
{
    PuertoSerial = new SerialPort(npuerto, baudios, paridad, nBits, bits_parada);
}
private void configura_puerto2(string npuerto, int baudios, Parity paridad, int
nBits, StopBits bits_parada)
{
    PuertoSerial1 = new SerialPort(npuerto, baudios, paridad, nBits,
bits_parada);
}
private void abre_puerto()
{
    if (PuertoSerial != null)
    {
        PuertoSerial.Open();
        puertoAbierto = true;
    }
}
private void cierra_puerto()
{
    if (PuertoSerial != null)
    {
        if (PuertoSerial.IsOpen == true)
        {
            PuertoSerial.Close();
            btnCerrar.Enabled = false;
            btnAbrir.Enabled = true;
            puertoAbierto = false;
        }
    }
}
private void abre_puerto2()
{
    if (PuertoSerial1 != null)
    {
        PuertoSerial1.Open();
        puertoAbierto = true;
    }
}
private void cierra_puerto2()
{
    if (PuertoSerial1 != null)
    {
        if (PuertoSerial1.IsOpen == true)
        {
            PuertoSerial1.Close();
            btnCerrar3.Enabled = false;
            btnAbrir3.Enabled = true;
            puertoAbierto = false;
        }
    }
}

```

```

    }
}
}
//Datos enviados mediante cada módulo.
private void enviar(byte[] dato)
{
    if ((PuertoSerial != null) && (PuertoSerial.IsOpen == true))
        PuertoSerial.Write(dato, 0, 21);
}
private void enviar2(byte[] dato)
{
    if ((PuertoSerial1 != null) && (PuertoSerial1.IsOpen == true))
        PuertoSerial1.Write(dato, 0, 21);
}
private void cmbPuerto_SelectedIndexChanged(object sender, EventArgs e)
{
    configura_puerto(cmbPuerto.Text, Convert.ToInt32(cmbVel.Text), Parity.None,
8, StopBits.One);
    btnAbrir.Enabled = true;
}
private void cmbVel_SelectedIndexChanged_1(object sender, EventArgs e)
{
    configura_puerto(cmbPuerto.Text, Convert.ToInt32(cmbVel.Text), Parity.None,
8, StopBits.One);
    btnAbrir.Enabled = true;
}
private void cmbPuerto2_SelectedIndexChanged_1(object sender, EventArgs e)
{
    configura_puerto2(cmbPuerto2.Text, Convert.ToInt32(cmbVel2.Text),
Parity.None, 8, StopBits.One);
    btnAbrir3.Enabled = true;
}
private void cmbVel2_SelectedIndexChanged_1(object sender, EventArgs e)
{
    configura_puerto2(cmbPuerto2.Text, Convert.ToInt32(cmbVel2.Text),
Parity.None, 8, StopBits.One);
    btnAbrir3.Enabled = true;
}
private void btnAbrir_Click_1(object sender, EventArgs e)
{
    try
    {
        abre_puerto();
        btnCerrar.Enabled = true;
        btnAbrir.Enabled = false;
    }
    catch (Exception)
    {
        cierra_puerto();
        MessageBox.Show("No se puede establecer la conexion en " + cmbPuerto.Text
+ " verifique que el \n dispositivo esta conectado, en caso de estarlo desconectar y
volver a conectar", "Error");
    }
}
private void btnCerrar_Click_1(object sender, EventArgs e)
{
    cierra_puerto();
}
private void btnAbrir3_Click_1(object sender, EventArgs e)

```

```

{
    try
    {
        abre_puerto2();
        btnCerrar3.Enabled = true;
        btnAbrir3.Enabled = false;
    }
    catch (Exception)
    {
        cierra_puerto2();
        MessageBox.Show("No se puede establecer la conexion en " +
cmbPuerto2.Text + " verifique que el \n dispositivo esta conectado, en caso de estarlo
desconectar y volver a conectar", "Error");
    }
}
private void btnCerrar3_Click_1(object sender, EventArgs e)
{
    cierra_puerto2();
}
/*****TUIOLISTENER*****/
//Agregar,actualizar y remover objeto fiducial.
public void addTuioObject(TuioObject o)
{
    //el fiducial 0 está asignado al robot 1,al obstáculo del C2.
    if (o.getFiducialID() == 0)
    {
        Xrob = o.xpos;
        Yrob = (1 - o.ypos) * 2 / 3;
        Arob = o.getAngleDegrees();
        //1 - para invertir el eje y por la proporción de la cámara
    }
    //el fiducial 1 está asignado a meta 1 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 1)
    {
        Xgol = o.xpos;
        Ygol = (1 - o.ypos) * 2 / 3;
        Agol = o.getAngleDegrees();
    }
    //el fiducial 2 está asignado a meta 2 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 2)
    {
        Xgol2 = o.xpos;
        Ygol2 = (1 - o.ypos) * 2 / 3;
        Agol2 = o.getAngleDegrees();
    }
    //1 - para invertir el eje y por la proporción de la cámara
    else if (o.getFiducialID() == 3)
    {
        Xrob2 = o.xpos;
        Yrob2 = (1 - o.ypos) * 2 / 3;
        Arob2= o.getAngleDegrees();
    }
    //el fiducial 4 está asignado a meta 3 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 4)
    {
        Xgol3 = o.xpos;
        Ygol3 = (1 - o.ypos) * 2 / 3;
        Agol3 = o.getAngleDegrees();
    }
}

```

```

}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 5)
{
    Xgol4 = o.xpos;
    Ygol4 = (1 - o.ypos) * 2 / 3;
    Agol4 = o.getAngleDegrees();
}
//METAS DE DESCARGA
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 6)
{
    Xgol5 = o.xpos;
    Ygol5 = (1 - o.ypos) * 2 / 3;
    Agol5 = o.getAngleDegrees();
}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 7)
{
    Xgol6 = o.xpos;
    Ygol6 = (1 - o.ypos) * 2 / 3;
    Agol6 = o.getAngleDegrees();
}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 8)
{
    Xgol7 = o.xpos;
    Ygol7 = (1 - o.ypos) * 2 / 3;
    Agol7 = o.getAngleDegrees();
}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 9)
{
    Xgol8 = o.xpos;
    Ygol8 = (1 - o.ypos) * 2 / 3;
    Agol8 = o.getAngleDegrees();
}
}
}
public void updateTuioObject(TuioObject o)
{
    //el fiducial 0 está asignado al robot 1,al obstáculo del C2.
    if (o.getFiducialID() == 0)
    {
        Xrob = o.xpos;
        Yrob = (1 - o.ypos) * 2 / 3;
        Arob = o.getAngleDegrees();
        //1 - para invertir el eje y por la proporción de la cámara
    }
    //el fiducial 1 está asignado a meta 1 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 1)
    {
        Xgol = o.xpos;
        Ygol = (1 - o.ypos) * 2 / 3;
        Agol = o.getAngleDegrees();
    }
    //el fiducial 2 está asignado a meta 2 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 2)
    {
        Xgol2 = o.xpos;

```

```

        Ygol2 = (1 - o.ypos) * 2 / 3;
        Agol2 = o.getAngleDegrees();
    }
    else if (o.getFiducialID() == 3)
    {
        Xrob2 = o.xpos;
        Yrob2 = (1 - o.ypos) * 2 / 3;
        Arob2 = o.getAngleDegrees();
        //1 - para invertir el eje y por la proporción de la cámara
    }
    //el fiducial 4 está asignado a meta 3 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 4)
    {
        Xgol3 = o.xpos;
        Ygol3 = (1 - o.ypos) * 2 / 3;
        Agol3 = o.getAngleDegrees();
    }
    //el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 5)
    {
        Xgol4 = o.xpos;
        Ygol4 = (1 - o.ypos) * 2 / 3;
        Agol4 = o.getAngleDegrees();
    }
    //METAS DE DESCARGA
    //el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 6)
    {
        Xgol5 = o.xpos;
        Ygol5 = (1 - o.ypos) * 2 / 3;
        Agol5 = o.getAngleDegrees();
    }
    //el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 7)
    {
        Xgol6 = o.xpos;
        Ygol6 = (1 - o.ypos) * 2 / 3;
        Agol6 = o.getAngleDegrees();
    }
    //el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 8)
    {
        Xgol7 = o.xpos;
        Ygol7 = (1 - o.ypos) * 2 / 3;
        Agol7 = o.getAngleDegrees();
    }
    //el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 9)
    {
        Xgol8 = o.xpos;
        Ygol8 = (1 - o.ypos) * 2 / 3;
        Agol8 = o.getAngleDegrees();
    }
}
public void removeTuioObject(TuioObject o)
{
    //el fiducial 0 está asignado al robot 1,al obstáculo del C2.
    if (o.getFiducialID() == 0)
    {

```

```

    Xrob = o.xpos;
    Yrob = (1 - o.ypos) * 2 / 3;
    Arob = o.getAngleDegrees();
    //1 - para invertir el eje y por la proporción de la cámara
}
//el fiducial 1 está asignado a meta 1 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 1)
{
    Xgol = o.xpos;
    Ygol = (1 - o.ypos) * 2 / 3;
    Agol = o.getAngleDegrees();
}
//el fiducial 2 está asignado a meta 2 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 2)
{
    Xgol2 = o.xpos;
    Ygol2 = (1 - o.ypos) * 2 / 3;
    Agol2 = o.getAngleDegrees();
}
else if (o.getFiducialID() == 3)
{
    Xrob2 = o.xpos;
    Yrob2 = (1 - o.ypos) * 2 / 3;
    Arob2 = o.getAngleDegrees();
    //1 - para invertir el eje y por la proporción de la cámara
}
//el fiducial 4 está asignado a meta 3 u obstáculo de C1 o C2.;
else if (o.getFiducialID() == 4)
{
    Xgol3 = o.xpos;
    Ygol3 = (1 - o.ypos) * 2 / 3;
    Agol3 = o.getAngleDegrees();
}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 5)
{
    Xgol4 = o.xpos;
    Ygol4 = (1 - o.ypos) * 2 / 3;
    Agol4 = o.getAngleDegrees();
}
//METAS DE DESCARGA
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 6)
{
    Xgol5 = o.xpos;
    Ygol5 = (1 - o.ypos) * 2 / 3;
    Agol5 = o.getAngleDegrees();
}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 7)
{
    Xgol6 = o.xpos;
    Ygol6 = (1 - o.ypos) * 2 / 3;
    Agol6 = o.getAngleDegrees();
}
//el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
else if (o.getFiducialID() == 8)
{
    Xgol7 = o.xpos;

```

```

        Ygol7 = (1 - o.ypos) * 2 / 3;
        Agol7 = o.getAngleDegrees();
    }
    //el fiducial 5 está asignado a meta 4 u obstáculo de C1 o C2.
    else if (o.getFiducialID() == 9)
    {
        Xgol8 = o.xpos;
        Ygol8 = (1 - o.ypos) * 2 / 3;
        Agol8 = o.getAngleDegrees();
    }
}
public void addTuioCursor(TuioCursor tuioCursor)
{
    throw new Exception("The method or operation is not implemented.");
}
public void updateTuioCursor(TuioCursor tuioCursor)
{
    throw new Exception("The method or operation is not implemented.");
}
public void removeTuioCursor(TuioCursor tuioCursor)
{
    throw new Exception("The method or operation is not implemented.");
}
public void refresh(long timestamp)
//Actualizando los datos obtenidos por Reactivision.
{
    //Actualiza etiquetas
    UpdateLblTxt(lblX, Convert.ToString(Xrob));
    UpdateLblTxt(lblY, Convert.ToString(Yrob));
    UpdateLblTxt(lblA, Convert.ToString(Arob));
    UpdateLblTxt(lblX2, Convert.ToString(Xrob2));
    UpdateLblTxt(lblY2, Convert.ToString(Yrob2));
    UpdateLblTxt(lblA2, Convert.ToString(Arob2));
}
/*****METAS*****/
private void meta()//Asignación de cajas.
{
    if (metaflag == true)
    {
        //Calculando distancias de cajas al C1
        if (meta1 == true)
        {
            //distancia C1 al goal 1
            d1m1 = Math.Sqrt((Xrob - Xgol) * (Xrob - Xgol) + (Yrob - Ygol) *
(Yrob - Ygol));
        }
        else { d1m1 = 2; }
        if (meta2 == true)
        {
            //distancia C1 al goal 2
            d1m2 = Math.Sqrt((Xrob - Xgol2) * (Xrob - Xgol2) + (Yrob - Ygol2) *
(Yrob - Ygol2));
        }
        else { d1m2 = 2; }
        if (meta3 == true)
        {
            //distancia C1 al goal 3
            d1m3 = Math.Sqrt((Xrob - Xgol3) * (Xrob - Xgol3) + (Yrob - Ygol3) *
(Yrob - Ygol3));
        }
    }
}

```

```

    }
    else { d1m3 = 2; }
    if (meta4 == true)
    {
        //distancia C1 al goal 4
        d1m4 = Math.Sqrt((Xrob - Xgol4) * (Xrob - Xgol4) + (Yrob - Ygol4) *
(Yrob - Ygol4));
    }
    else { d1m4 = 2; }
    double A = 0;
    if (d1m1 == 2 && d1m2 == 2 && d1m3 == 2 && d1m4 == 2)
    { A = 1; }
    else { A = 20; }
    //Código de arreglo para seleccionar la caja con la menor distancia al C1
    double[] arreglo1 = { d1m1, d1m2, d1m3, d1m4,A};
    Array.Sort(arreglo1);
    double minimo = arreglo1[0];
    if (minimo == d1m1)
    {
        XgolC1 = Xgol;
        YgolC1 = Ygol;
        meta1 = false;
        metaflag = false;
        metafM1C1 = true;
    }
    else if (minimo == d1m2)
    {
        XgolC1 = Xgol2;
        YgolC1 = Ygol2;
        meta2 = false;
        metaflag = false;
        metafM2C1 = true;
    }
    else if (minimo == d1m3)
    {
        XgolC1 = Xgol3;
        YgolC1 = Ygol3;
        meta3 = false;
        metaflag = false;
        metafM3C1 = true;
    }
    else if (minimo == d1m4)
    {
        XgolC1 = Xgol4;
        YgolC1 = Ygol4;
        meta4 = false;
        metaflag = false;
        metafM4C1 = true;
    }
    else if (minimo == A)
    {
        XgolC1 = 0.60f;
        YgolC1 = 0.54f;
    }
}
if(metaflag2==true)
{
    //Calculando distancias de cajas al C2
    if (meta1 == true)

```



```

    {
        //distancia C2 al goal 1
        d2m1 = Math.Sqrt((Xrob2 - Xgol) * (Xrob2 - Xgol) + (Yrob2 - Ygol) *
(Yrob2 - Ygol));
    }
    else { d2m1 = 2; }
    if (meta2 == true)
    {
        //distancia C2 al goal 2
        d2m2 = Math.Sqrt((Xrob2 - Xgol2) * (Xrob2 - Xgol2) + (Yrob2 - Ygol2)
* (Yrob2 - Ygol2));
    }
    else { d2m2 = 2; }
    if (meta3 == true)
    {
        //distancia C2 al goal 3
        d2m3 = Math.Sqrt((Xrob2 - Xgol3) * (Xrob2 - Xgol3) + (Yrob2 - Ygol3)
* (Yrob2 - Ygol3));
    }
    else { d2m3 = 2; }
    if (meta4 == true)
    {
        //distancia C2 al goal 4
        d2m4 = Math.Sqrt((Xrob2 - Xgol4) * (Xrob2 - Xgol4) + (Yrob2 - Ygol4)
* (Yrob2 - Ygol4));
    }
    else { d2m4 = 2; }
    double B = 0;
    if (d2m1 == 2 && d2m2 == 2 && d2m3 == 2 && d2m4 == 2)
    { B = 1; }
    else { B = 20; }
    //Código de arreglo para seleccionar la caja con la menor distancia al C2
    double[] arreglo2 = { d2m1, d2m2, d2m3, d2m4, B };
    Array.Sort(arreglo2);
    double minimo2 = arreglo2[0];
    if (minimo2 == d2m1)
    {
        XgolC2 = Xgol;
        YgolC2 = Ygol;
        meta1 = false;
        metaflag2 = false;
        metafM1C2 = true;
    }
    else if (minimo2 == d2m2)
    {
        XgolC2 = Xgol2;
        YgolC2 = Ygol2;
        meta2 = false;
        metaflag2 = false;
        metafM2C2 = true;
    }
    else if (minimo2 == d2m3)
    {
        XgolC2 = Xgol3;
        YgolC2 = Ygol3;
        meta3 = false;
        metaflag2 = false;
        metafM3C2 = true;
    }
}

```

```

else if (minimo2 == d2m4)
{
    XgolC2 = Xgol4;
    YgolC2 = Ygol4;
    meta4 = false;
    metaflag2 = false;
    metafM4C2 = true;
}
else if (minimo2 == B)
{
    XgolC2 = 0.60f;
    YgolC2 = 0.15f;
}
}
}
//Asignación de lugares para descargar cajas de los Cobots.
private void metadescarga()
{
    if (metafflag == true)
    {
        //COBOT 1
        if (metafM1C1 == true)
        {
            XgolC1 = Xgol5;
            YgolC1 = Ygol5;
            metafflag = false;
            metafM1C1 = false;
        }
        else if (metafM2C1 == true)
        {
            XgolC1 = Xgol6;
            YgolC1 = Ygol6;
            metafflag = false;
            metafM2C1 = false;
        }
        else if (metafM3C1 == true)
        {
            XgolC1 = Xgol7;
            YgolC1 = Ygol7;
            metafflag = false;
            metafM3C1 = false;
        }
        else if (metafM4C1 == true)
        {
            XgolC1 = Xgol8;
            YgolC1 = Ygol8;
            metafflag = false;
            metafM4C1 = false;
        }
        UpdateLblTxt(lblDm, Convert.ToString(dg));
    }
    if (metafflag2 == true)
    {
        //COBOT 2
        if (metafM1C2 == true)
        {
            XgolC2 = Xgol5;
            YgolC2 = Ygol5;
            metafflag2 = false;
        }
    }
}

```

```

        metafM1C2 = false;
    }
    else if (metafM2C2 == true)
    {
        XgolC2 = Xgol6;
        YgolC2 = Ygol6;
        metafflag2 = false;
        metafM2C2 = false;
    }
    else if (metafM3C2 == true)
    {
        XgolC2 = Xgol7;
        YgolC2 = Ygol7;
        metafflag2 = false;
        metafM3C2 = false;
    }
    else if (metafM4C2 == true)
    {
        XgolC2 = Xgol8;
        YgolC2 = Ygol8;
        metafflag2 = false;
        metafM4C2 = false;
    }
    }
    UpdateLblTxt(lblDm2, Convert.ToString(dg2));
}
/*****CAMPOS POTENCIALES*****/
private void campospotenciales()//Campos potenciales Cobot 1.
{
    fuerzatraccion();
    fuerzarepulsion();
    atrax = atrax + repx;
    atray = atray + repy;
    UpdateLblTxt(lblVLin, Convert.ToString(atrax));
    UpdateLblTxt(lblVAng, Convert.ToString(atray));
    //calcula ángulo evitando division entre 0
    if ((atrax == 0) && (atray == 0)) anguloatr = 0;
    else if ((atrax == 0) && (atray > 0)) anguloatr = (Math.PI / 2);
    else if ((atrax == 0) && (atray < 0)) anguloatr = ((3 * Math.PI) / 2);
    else if ((atrax > 0) && (atray == 0)) anguloatr = 0;
    else if ((atrax < 0) && (atray == 0)) anguloatr = Math.PI;
    else if ((atrax != 0) && (atray != 0))
    {
        if ((atrax > 0) && (atray > 0)) anguloatr = Math.Atan(atray / atrax);
        else if ((atrax > 0) && (atray < 0)) anguloatr = (2 * Math.PI) +
(Math.Atan(atray / atrax));
        else if ((atrax < 0) && (atray > 0)) anguloatr = (Math.PI) +
(Math.Atan(atray / atrax));
        else anguloatr = (Math.PI) + (Math.Atan(atray / atrax));
    }
    else
    {
        anguloatr = Math.Atan2(atrax, atray);
    }
    anguloatr = anguloatr * 180 / Math.PI;
    errorangulo = anguloatr - Arob; //Calcula el error
    if (errorangulo < 0) errorangulo = errorangulo + 360;
    //imprime los valores a la forma
    UpdateLblTxt(lblXpot, Convert.ToString(atrax));
}

```

```

UpdateLblTxt(lblYpot, Convert.ToString(atray));
UpdateLblTxt(lblApot, Convert.ToString(anguloatr));
}
private void campospotenciales2()//Campos potenciales Cobot 2.
{
    fuerzatraccion2();
    fuerzarepulsion2();
    atrax2 = atrax2 + repx2;
    atray2 = atray2 + repy2;
    //calcula ángulo evitando division entre 0
    if ((atrax2 == 0) && (atray2 == 0)) anguloatr2 = 0;
    else if ((atrax2 == 0) && (atray2 > 0)) anguloatr2 = (Math.PI / 2);
    else if ((atrax2 == 0) && (atray2 < 0)) anguloatr2 = ((3 * Math.PI) / 2);
    else if ((atrax2 > 0) && (atray2 == 0)) anguloatr2 = 0;
    else if ((atrax2 < 0) && (atray2 == 0)) anguloatr2 = Math.PI;
    else if ((atrax2 != 0) && (atray2 != 0))
    {
        if ((atrax2 > 0) && (atray2 > 0)) anguloatr2 = Math.Atan(atray2 /
atrax2);
        else if ((atrax2 > 0) && (atray2 < 0)) anguloatr2 = (2 * Math.PI) +
(Math.Atan(atray2 / atrax2));
        else if ((atrax2 < 0) && (atray2 > 0)) anguloatr2 = (Math.PI) +
(Math.Atan(atray2 / atrax2));
        else anguloatr2 = (Math.PI) + (Math.Atan(atray2 / atrax2));
    }
    else
    {
        anguloatr2 = Math.Atan2(atrax2, atray2);
    }
    anguloatr2 = anguloatr2 * 180 / Math.PI;
    errorangulo2 = anguloatr2 - Arob2; //Calcula el error
    if (errorangulo2 < 0) errorangulo2 = errorangulo2 + 360;
    //imprime los valores a la forma
    UpdateLblTxt(lblXpot2, Convert.ToString(atrax2));
    UpdateLblTxt(lblYpot2, Convert.ToString(atray2));
    UpdateLblTxt(lblApot2, Convert.ToString(anguloatr2));
}
/*****FUERZAS*****/
private void fuerzatraccion()//Fuerza de atracción Cobot 1.
{
    atrax = XgolC1 - Xrob;
    atray = YgolC1 - Yrob;
    //distancia al goal y magnitud del vector de atracción
    dg = Math.Sqrt((atrax * atrax) + (atray * atray));
    if (dg > Kr)
    {
        // F =(E2 * vector atractivo) / magnitud del vector atractivo
        atrax = (atrax * E1) / dg;
        atray = (atray * E1) / dg;
    }
    else
    {
        // F = E1*vector atractivo
        atrax = (atrax * E1);
        atray = (atray * E1);
    }
    UpdateLblTxt(lblD, Convert.ToString(dg));
}

```

```

private void fuerzatraccion2()//Fuerza de atracción Cobot 2.
{
    atrax2 = XgolC2 - Xrob2;
    atray2 = YgolC2 - Yrob2;
    //distancia al goal y magnitud del vector de atracción
    dg2 = Math.Sqrt((atrax2 * atrax2) + (atray2 * atray2));
    if (dg2 > Kr)
    {
        // F =(E2 * vector atractivo) / magnitud del vector atractivo
        atrax2 = (atrax2 * E1) / dg2;
        atray2 = (atray2 * E1) / dg2;
    }
    else
    {
        // F = E1*vector atractivo
        atrax2 = (atrax2 * E1);
        atray2 = (atray2 * E1);
    }
    UpdateLblTxt(lblD2, Convert.ToString(dg2));
}
private void fuerzarepulsion()//Fuerza de repulsión Cobot 1.
{
    repx = Xrob2 - Xrob;
    repy = Yrob2 - Yrob;
    dobs = Math.Sqrt((repx * repx) + (repy * repy)); //distancia C1 al C2
    //si se encuentra en el radio de influencia del obstáculo, dobs de .15 está
cerca como para colisionar
    if (dobs < Ki)
    {
        //Frep = -Nu * (1/dobs - 1/dg) * (1/(dobs*dobs)) * (vector repusivo/
magnitud del vector repulsivo)
        repx = -Nu * (1 / dobs - 1 / Ki) * (1 / (dobs * dobs)) * (repx / dobs);
        repy = - Nu * (1 / dobs - 1 / Ki) * (1 / (dobs * dobs)) * (repy / dobs);
    }
    else // no está dentro del radio de influencia
    {
        repx = 0;
        repy = 0;
    }
    if (meta1 == true && d1m1 < Ki)
    {
        double repxm1 = -Nu * (1 / d1m1 - 1 / Ki) * (1 / (d1m1 * d1m1)) * ((Xgol
- Xrob) / d1m1);
        double repym1 = -Nu * (1 / d1m1 - 1 / Ki) * (1 / (d1m1 * d1m1)) * ((Ygol
- Yrob) / d1m1);
        repx = repxm1 + repx;
        repy = repym1 + repy;
    }
    if (meta2 == true && d1m2 < Ki)
    {
        double repxm2 = -Nu * (1 / d1m2 - 1 / Ki) * (1 / (d1m2 * d1m2)) * ((Xgol2
- Xrob) / d1m2);
        double repym2 = -Nu * (1 / d1m2 - 1 / Ki) * (1 / (d1m2 * d1m2)) * ((Ygol2
- Yrob) / d1m2);
        repx = repxm2 + repx;
        repy = repym2 + repy;
    }
    if (meta3 == true && d1m3 < Ki)
    {

```

```

        double repxm3 = -Nu * (1 / d1m3 - 1 / Ki) * (1 / (d1m3 * d1m3)) * ((Xgol3
- Xrob) / d1m3);
        double repym3 = -Nu * (1 / d1m3 - 1 / Ki) * (1 / (d1m3 * d1m3)) * ((Ygol3
- Yrob) / d1m3);
        repx = repxm3 + repx;
        repy = repym3 + repy;
    }
    if (meta4 == true && d1m4 < Ki)
    {
        double repxm4 = -Nu * (1 / d1m4 - 1 / Ki) * (1 / (d1m4 * d1m4)) * ((Xgol4
- Xrob) / d1m4);
        double repym4 = -Nu * (1 / d1m4 - 1 / Ki) * (1 / (d1m4 * d1m4)) * ((Ygol4
- Yrob) / d1m4);
        repx = repxm4 + repx;
        repy = repym4 + repy;
    }
    UpdateLblTxt(lblDo, Convert.ToString(dobs)); //distancia C1 al C2
}
private void fuerzarepulsion2() //Fuerza de repulsión Cobot 2.
{
    repx2 = Xrob2 - Xrob;
    repy2 = Yrob2 - Yrob;
    //distancia al obstáculo o al cobot1
    dobs2 = Math.Sqrt((repx2 * repx2) + (repy2 * repy2));
    disx2 = XgolC1 - Xrob2;
    disy2 = YgolC1 - Yrob2;
    dist2 = Math.Sqrt((disx2 * disx2) + (disy2 * disy2));
    if (dobs2 < Ki) //si se encuentra en el radio de influencia del obstáculo
    {
        //Frep = -Nu * (1/dobs - 1/dg) * (1/(dobs*dobs)) * (vector repusivo/
magnitud del vector repulsivo)
        repx2 = -Nu * (1 / dobs2 - 1 / Ki) * (1 / (dobs2 * dobs2)) * (repx2 /
dobs2);
        repy2 = -Nu * (1 / dobs2 - 1 / Ki) * (1 / (dobs2 * dobs2)) * (repy2 /
dobs2);
    }
    else // no está dentro del radio de influencia
    {
        repx2 = 0;
        repy2 = 0;
    }
    if (meta1 == true && d2m1 < Ki)
    {
        double repxm1 = -Nu * (1 / d2m1 - 1 / Ki) * (1 / (d2m1 * d2m1)) * ((Xgol1
- Xrob2) / d2m1);
        double repym1 = -Nu * (1 / d2m1 - 1 / Ki) * (1 / (d2m1 * d2m1)) * ((Ygol1
- Yrob2) / d2m1);
        repx2 = repxm1 + repx2;
        repy2 = repym1 + repy2;
    }
    if (meta2 == true && d2m2 < Ki)
    {
        double repxm2 = -Nu * (1 / d2m2 - 1 / Ki) * (1 / (d2m2 * d2m2)) * ((Xgol2
- Xrob2) / d2m2);
        double repym2 = -Nu * (1 / d2m2 - 1 / Ki) * (1 / (d2m2 * d2m2)) * ((Ygol2
- Yrob2) / d2m2);
        repx2 = repxm2 + repx2;
        repy2 = repym2 + repy2;
    }
}

```

```

        if (meta3 == true && d2m3 < Ki)
        {
            double repxm3 = -Nu * (1 / d2m3 - 1 / Ki) * (1 / (d2m3 * d2m3)) * ((Xgol3
- Xrob2) / d2m3);
            double repym3 = -Nu * (1 / d2m3 - 1 / Ki) * (1 / (d2m3 * d2m3)) * ((Ygol3
- Yrob2) / d2m3);
            repx2 = repxm3 + repx2;
            repy2 = repym3 + repy2;
        }
        if (meta4 == true && d2m4 < Ki)
        {
            double repxm4 = -Nu * (1 / d2m4 - 1 / Ki) * (1 / (d2m4 * d2m4)) * ((Xgol4
- Xrob2) / d2m4);
            double repym4 = -Nu * (1 / d2m4 - 1 / Ki) * (1 / (d2m4 * d2m4)) * ((Ygol4
- Yrob2) / d2m4);
            repx2 = repxm4 + repx2;
            repy2 = repym4 + repy2;
        }
        UpdateLblTxt(lblDo2, Convert.ToString(dobs2));
    }
/*****SALIDAS DE CONTROL*****/
private void salidasdecontrol()//Salidas Cobot 1.
{
    //Determinación de la velocidad angular de acuerdo con el modelo
    if (errorangulo <= 90 || errorangulo >= 270)
    {
        velangctrl = Wmax * (Math.Sin(errorangulo * Math.PI / 180));
    }
    //Corrección para ángulos fuera del rango
    else if (errorangulo <= 180)
    {
        velangctrl = Wmax; //quitar 0.7 a los 2
    }
    else
    {
        velangctrl = -Wmax;
    }
    //Para la velocidad lineal y para limitar mejorar la respuesta en vueltas cerradas
    if (dg > Kr)
    {
        vellinctrl = Vmax;
    }
    else
    {
        vellinctrl = Vmax * dg / Kr;
    }
    //de acuerdo con el modelo de control
    if (dg > dis)
    {
        wi = ((vellinctrl - ((b / 2) * velangctrl)) / c) * kprop;
        wd = ((vellinctrl + ((b / 2) * velangctrl)) / c) * kprop;
        int wie = Convert.ToInt16(wi);
        int wde = Convert.ToInt16(wd);
        motora = 128 - wde;
        motorb = wie + 128;
    }
    else
    {
        wi = ((vellinctrl - ((b / 2) * velangctrl)) / c) * kprop * dg;

```

```

wd = ((vellinctrl + ((b / 2) * velangctrl)) / c) * kprop * dg;
int wie = Convert.ToInt16(wi);
int wde = Convert.ToInt16(wd);
motora = 128 - wde;
motorb = wie + 128;
if (dg <= 0.080) //(wi <= 0.4 || wd <= 0.4)
{
    motora = 128;
    motorb = 128;
    enviarMensaje(motores, motora, motorb);
    enviarMensaje(motores, motora, motorb);
    if (contador <= 3)
    {
        motores = 0x13;
        enviarMensaje(motores, motora, motorb);
        enviarMensaje(motores, motora, motorb);
        contador++;
    }
    motores = 0xFC;
    estrategial.Stop();
    cajal.Start();
}
else
{
    motora = 128 - wde;
    motorb = wie + 128;
}
}
UpdateLblTxt(lblVi, Convert.ToString(motora));
UpdateLblTxt(lblVd, Convert.ToString(motorb));
UpdateLblTxt(lblVLin, Convert.ToString(vellinctrl));
UpdateLblTxt(lblVAng, Convert.ToString(velangctrl));
}
private void salidasdecontrol2() //Salidas Cobot 2.
{
    //Determinación de la velocidad angular de acuerdo con el modelo
    if (errorangulo2 <= 90 || errorangulo2 >= 270)
    {
        velangctrl2 = Wmax * (Math.Sin(errorangulo2 * Math.PI / 180));
    }
    //Corrección para ángulos fuera del rango
    else if (errorangulo2 <= 180)
    {
        velangctrl2 = Wmax; //quitar 0.7 a los 2
    }
    else
    {
        velangctrl2 = -Wmax;
    }
}
//Para la velocidad lineal y para limitar mejorar la respuesta en vueltas cerradas
if (dg2 > Kr)
{
    vellinctrl2 = Vmax;
}
else
{
    vellinctrl2 = Vmax * dg2 / Kr;
}
//de acuerdo con el modelo de control

```



```

//dist entre C1-C2 < el R d influencia del C1 o bien la distancia entre el C1-M1 > C2-M2
if (dobs2 > Ki || dg>dist2)
{
    if (dg2 > dis)
    {
        wi2 = ((vellinctrl2 - ((b / 2) * velangctrl2)) / c) * kprop;
        wd2 = ((vellinctrl2 + ((b / 2) * velangctrl2)) / c) * kprop;
        int wie2 = Convert.ToInt16(wi2);
        int wde2 = Convert.ToInt16(wd2);
        motora2 = 128 - wde2;
        motorb2 = wie2 + 128;
    }
    else
    {
        wi2 = ((vellinctrl2 - ((b / 2) * velangctrl2)) / c) * kprop * dg2;
        wd2 = ((vellinctrl2 + ((b / 2) * velangctrl2)) / c) * kprop * dg2;
        int wie2 = Convert.ToInt16(wi2);
        int wde2 = Convert.ToInt16(wd2);
        motora2 = 128 - wde2;
        motorb2 = wie2 + 128;
        if (dg2<=0.08)//wi2 <= 0.4 || wd2 <= 0.4)
        {
            motora2 = 128;
            motorb2 = 128;
            mandarDatosRF(motores2, motora2, motorb2);
            mandarDatosRF(motores2, motora2, motorb2);
            if (contador2 <= 3)
            {
                motores2 = 0x13;
                mandarDatosRF(motores2, motora2, motorb2);
                mandarDatosRF(motores2, motora2, motorb2);
                contador2++;
            }
            motores2 = 0xFC;
            estrategia2.Stop();
            caja2.Start();
        }
        else
        {
            motora2 = 128 - wde2;
            motorb2 = wie2 + 128;
        }
    }
}
else
{
    motora2 = 128;
    motorb2 = 128;
}
UpdateLblTxt(lblVd2, Convert.ToString(motora2));
UpdateLblTxt(lblVi2, Convert.ToString(motorb2));
UpdateLblTxt(lblVLin2, Convert.ToString(vellinctrl2));
UpdateLblTxt(lblVAng2, Convert.ToString(velangctrl2));
}
/*****MENSAJES A ENVIAR*****/
//Envío de mensaje por Xbee a Cobot 1.
void enviarMensaje(int motores,int motora, int motorb)
{
    byte[] bdata = new byte[21];

```

```

byte checksum = 0;
int checksumInt = 0;
//Inicio de comunicación
bdata[0] = (byte)0x7e;
//Numero de bytes a enviar
bdata[1] = (byte)0x00;
bdata[2] = (byte)0x11;
//Identificador
bdata[3] = (byte)0x10;
//Ack
bdata[4] = (byte)0x01;
//Dirección del dispositivo a enviar (Router 25)
bdata[5] = (byte)0x00;
bdata[6] = (byte)0x13;
bdata[7] = (byte)0xA2;
bdata[8] = (byte)0x00;
bdata[9] = (byte)0x40;
bdata[10] = (byte)0x64;
bdata[11] = (byte)0x9D;
bdata[12] = (byte)0xAA;
//Dirección de 16 bits
bdata[13] = (byte)0xff;
bdata[14] = (byte)0xfe;
//Broadcast al máximo
bdata[15] = (byte)0x01;
//Opciones
bdata[16] = (byte)0x00;
//Mensaje
bdata[17] = (byte)motores;           //montacargas
bdata[18] = (byte)motora;           //velocidad angular izq.
bdata[19] = (byte)motorb;           //velocidad angular der.
for (int i = 3; i <= 19; i++)
{
    checksumInt = checksumInt + bdata[i];
}
while (checksumInt > 255)
{
    checksumInt = checksumInt - 256;
}
checksumInt = 255 - checksumInt;
checksum = Convert.ToByte(checksumInt);
//Checksum
bdata[20] = checksum;
enviar(bdata);
}
//Envío de mensaje por Xbee a Cobot 2.
void mandarDatosRF(int motores2,int motora2, int motorb2)
{
    byte[] bdata = new byte[21];
    byte checksum = 0;
    int checksumInt = 0;
    //Inicio de comunicación
    bdata[0] = (byte)0x7e;
    //Numero de bytes a enviar
    bdata[1] = (byte)0x00;
    bdata[2] = (byte)0x11;
    //Identificador
    bdata[3] = (byte)0x10;
    //Ack

```

```

    bdato[4] = (byte)0x01;
    //Dirección del dispositivo a enviar
    bdato[5] = (byte)0x00;
    bdato[6] = (byte)0x13;
    bdato[7] = (byte)0xA2;
    bdato[8] = (byte)0x00;
    bdato[9] = (byte)0x40;
    bdato[10] = (byte)0x3c;
    bdato[11] = (byte)0x8c;
    bdato[12] = (byte)0xb1;
    //dirección de 16 bits
    bdato[13] = (byte)0xff;
    bdato[14] = (byte)0xfe;
    //Broadcast al máximo
    bdato[15] = (byte)0x01;
    //Opciones
    bdato[16] = (byte)0x00;
    //Mensaje
    bdato[17] = (byte)motores2;        //montacargas
    bdato[18] = (byte)motora2;        //velocidad angular izq.
    bdato[19] = (byte)motorb2;        //velocidad angular der.
    for (int i = 3; i <= 19; i++)
    {
        checkSumInt = checkSumInt + bdato[i];
    }
    while (checkSumInt > 255)
    {
        checkSumInt = checkSumInt - 256;
    }
    checkSumInt = 255 - checkSumInt;
    checkSum = Convert.ToByte(checkSumInt);
    //Checksum
    bdato[20] = checkSum;
    enviar2(bdato);
}
//Para salir del hilo se crea un delegate generic sin argumentos.
delegate void GenericDelegate();
void UpdateLblTxt(Label lbl, string text)
{
    GenericDelegate dlg = delegate()
    {
        lbl.Text = text;
    };
    lbl.Invoke(dlg);
}
/*****BOTONES DE COMIENZO*****/
//Sale del programa pero no lo cierra.
private void Salir_Click(object sender, EventArgs e)
{
    Application.Exit();
}
//Botón de comienzo/termino del programa
private void button1_Click(object sender, EventArgs e)
{
    if (estrategia1.Enabled == false)
    {
        registra = true;
        excel.Enabled=true;
        nreg=0;
    }
}

```

```

        btnExportar.Enabled=false;
        btnExportar.BackColor = Color.Chartreuse;
        estrategia1.Start();
        estrategia2.Start();
        boton1.Text = "DETENER";
        boton1.BackColor=Color.Red;
    }
    else
    {
        estrategia1.Stop();
        estrategia2.Stop();
        boton1.Text = "EMPEZAR";
        boton1.BackColor = Color.Lime;
        motora = 128;
        motorb = 128;
        enviarMensaje(motores,motora,motorb);
        motora2 = 128;
        motorb2 = 128;
        mandarDatosRF(motores2,motora2, motorb2);
    }
}

/*****TIMERS*****/
//Timer 100ms envía cada mensaje.
private void timer1_Tick(object sender, EventArgs e)
{
    enviarMensaje(motores, motora, motorb);
    mandarDatosRF(motores2, motora2, motorb2);
}
private void estrategia1_Tick(object sender, EventArgs e)
{
    meta();
    campospotenciales();
    salidasdecontrol();
}
private void estrategia2_Tick(object sender, EventArgs e)
{
    meta();
    campospotenciales2();
    salidasdecontrol2();
}
private void cajal_Tick(object sender, EventArgs e)
{
    if (metaflag == false && ciclo == false)
    {
        metaflag = true;
        ciclo = true;
        metadescarga();
        cajal.Stop();
        estrategia1.Start();
    }
    else if (metaflag == false && ciclo == true)
    {
        cajal.Stop();
        reversal.Start();
        motora = 146;
        motorb = 114;
        //Se establecen laS vels de los motores del C1 para que vaya en reversa
    }
}

```

```

}
private void caja2_Tick(object sender, EventArgs e)
{
    if (metaflag2 == false && ciclo2 == false)
    {
        metaflag2 = true;
        ciclo2 = true;
        metadescarga();
        caja2.Stop();
        estrategia2.Start();
    }
    else if (metaflag2 == false && ciclo2 == true)
    {
        caja2.Stop();
        reversa2.Start();
        motora2 = 146;
        motorb2 = 114;
        //Se establecen las vels de los motores del C2 para que vaya en reversa
    }
}
private void reversa1_Tick(object sender, EventArgs e)
{
    metaflag = true;
    ciclo = false;
    meta();
    reversa1.Stop();
    estrategial.Start();
}
private void reversa2_Tick(object sender, EventArgs e)
{
    metaflag2 = true;
    ciclo2 = false;
    meta();
    reversa2.Stop();
    estrategia2.Start();
}
private void excel_Tick(object sender, EventArgs e)
{
    if (registra)
    {
        registros[nreg, 0] = Xrob;
        registros[nreg, 1] = Yrob;
        registros[nreg, 2] = Arob;
        registros[nreg, 3] = XgolC1;
        registros[nreg, 4] = YgolC1;
        registros[nreg, 5] = vellinctrl;
        registros[nreg, 6] = velangctrl;
        registros[nreg, 7] = Xrob2;
        registros[nreg, 8] = Yrob2;
        registros[nreg, 9] = Arob2;
        registros[nreg, 10] = XgolC2;
        registros[nreg, 11] = YgolC2;
        registros[nreg, 12] = vellinctrl2;
        registros[nreg, 13] = velangctrl2;
        nreg = nreg + 1;
        if (nreg >= 2100) //Termina el muestreo
        {
            registra = false;
            excel.Enabled = false;
        }
    }
}

```

```

        btnExportar.Enabled = true;
        btnExportar.BackColor = Color.Yellow;
    }
}
}
/*****REGISTRO DE EXCEL*****/
private void btnExportar_Click(object sender, EventArgs e)
{
    btnExportar.Text = "EXPORTADO";
    try
    {
        AplicacionExcel = new Excel.ApplicationClass();
        AplicacionExcel.Visible = true;
        AplicacionExcel.UserControl = true;
        string workbookPath = "C:/Users/ALFRED/Documents/UNAM/TESIS/ESCRITURA &
COMPILACIÓN/DataCobots_M4.xlsx";
        Excel.Workbook Librodetrabajo =
AplicacionExcel.Workbooks.Open(workbookPath, 0, false, 5, "", "", false,
Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "", true, false, 0, true, false,
false);
        Excel.Sheets Hojaexcel = Librodetrabajo.Worksheets;
        string hojaactual = "Datos";
        Excel.Worksheet Hojadetrabajo =
(Excel.Worksheet)Hojaexcel.get_Item(hojaactual);
        Hojadetrabajo.get_Range("A2", "N2102").Value2 = registros;
        registra = false;
    }
    catch (Exception laException) //Cacha error
    {
        String errorMessage;
        errorMessage = "Error: ";
        errorMessage = String.Concat(errorMessage, laException.Message);
        errorMessage = String.Concat(errorMessage, " Line: ");
        errorMessage = String.Concat(errorMessage, laException.Source);
        MessageBox.Show(errorMessage, "Error");
    }
}
}
}
}

```

Apéndice B – Código grabado en los PIC-16F887

```
1: //////////////////////////////////////
2: //PROGRAMA PARA EL CONTROL DE LOS MOTORES DE LOS //
3: //COBOTS Y ENVIO Y RECEPCION DE DATOS POR ZIGBEE //
4: //DESARROLLADO POR: //
5: //DANIEL LIMA ROBLEDA //
6: //ALFREDO MIGUEL ANGELES GARCIA //
7: //TESIS: SISTEMA SEMI-AUTONOMO DE ROBOTS MOVILES //
8: //COLABORATIVOS PARA EL MANEJO DE MATERIALES //
9: //FACULTAD DE INGENIERÍA, UNAM 2010 //
10: //DEPARTAMENTO DE INGENIERÍA MECATRÓNICA //
11: //////////////////////////////////////
12:
13: #include <16F887.h>
14: #fuses HS, NOWDT, NOPUT, MCLR, NOPROTECT, NOCPD, NOBROWNOUT, NOIESO, NOFCMEN;
15: #use delay(clock=20000000)
16: #use standard_io(B)
17: #use standard_io(D)
18: #use i2c(Master, sda=PIN_C4, scl=PIN_C3) //Configuración de I2C
19: #use RS232(baud=115200, xmit=pin_C6, rcv=pin_C7, bits=8, PARITY=N)//Conf. RS232
20: #org 0x1F00, 0x1FFF void loader16F887(void){} //Memoria reservada tinyloader
21:
22: //Variables y banderas
23: char c;
24: int indice=0, checksum=0, byte1=0xFF, byte2=128, byte3=128, bandera=0;
25: int registro, valor, dato, flag=0, flagC=0, cont=0, derecha=0;
26: int1 s=0;
27:
28: //Instrucción para escribir en la tarjeta MD25
29: void write_md25(long int registro, BYTE data)
30: {
31:     short int status; //Variable status
32:     i2c_start(); //Inicializa la transmisión
33:     i2c_write(0xB0); //Escribe la palabra de control
34: // (dirección 0h + 0 para escritura)
35:     i2c_write(registro); //Dirección del registro a escribir en la MD25
36:     i2c_write(data); //Dato a escribir
37:     i2c_stop(); //Finaliza la transmisión
38:     i2c_start(); //Reinicio
39:     status=i2c_write(0xB0); //Lectura de bit ACK, evitar escrituras incorrectas
40:     while (status==1) //Espera a que la tarjeta MD25 responda
41:     {
42:         i2c_start();
43:         status=i2c_write(0xB0);
44:     }
45: }
46:
47: //Instrucción para leer la tarjeta MD25
48: BYTE read_md25(long int registro)
49: {
50:     BYTE data; //Variable (dato a leer)
51:     i2c_start(); //Inicializa la transmisión
52:     i2c_write(0xB0); //Escribe la palabra de control
53: // (dirección 0h + 0 para lectura)
54:     i2c_write(registro); //Dirección del registro a leer en la MD25
55:     i2c_start(); //Reinicio
```

```

56:  i2c_write(0xB1);           //Escribe la palabra de control
57:                               //(direccion 0h + 1 para lectura)
58:  data=i2c_read(0);          //Lectura del dato
59:  i2c_stop();                 //Finalizacion de la transmision
60:  return(data);
61: }
62:
63: //Servicio de interrupcion por recepcion RS232
64: #int_rda
65: void rda_service(void)
66: {
67:  c=getchar();                //Dato recibido en RS232
68:  output_toggle(pin_b7);      //Indicador de recepcion por RS232
69:  if(bandera==1)              //Activa bandera de inicio de recepción RS232
70:  {
71:      if(indice==15)          //Si indice es igual a 15
72:      {
73:          if(c==0xfc)         //Si dato recibido igual a 0xfc
74:          {
75:              flag=1;         //Activa bandera de escritura
76:              indice++;        //Aumenta indice.
77:          }
78:          else if(c==0x13)     //Si dato recibido igual a 0x13
79:          {
80:              flagC=1;        //Activa bandera de motor montacarga
81:              indice++;        //Incrementa el indice
82:          }
83:      }
84:      else if(flag==1 && indice==16) //Si escritura esta activada e indice 16
85:      {
86:          byte2=c;            //byte2 igual a dato recibido
87:          registro=0;         //Se escribira en registro cero de MD25
88:          write_md25(registro, byte2); //llama para escribir en la Tarjeta MD25
89:          indice++;           //Incrementa el indice
90:      }
91:      else if(flag==1 && indice==17) //Si escritura esta activada e indice 17
92:      {
93:          byte3=c;            //byte2 igual a dato recibido
94:          registro=1;         //Se escribira en registro uno de MD25
95:          write_md25(registro, byte3); //llama para escribir en la Tarjeta MD25
96:          indice++;           //Incrementa el indice
97:      }
98:      else if(indice>18)      //Si indice es mayor a 18
99:      {
100:         bandera=0;           //Reset bandera inicio de RS232
101:         indice=0;           //Riset indice
102:         flag=0;             //Reset bandera de escritura
103:     }
104:     else
105:     {
106:         indice++;           //Incrementa indice
107:     }
108: }
109: else
110: {
111:     if(c==0x7E && indice==0) //Si dato recibido 0x7e e indice cero
112:     {
113:         bandera=1;          //Activa bandera inicio de recepcion RS232
114:         indice++;           //Incrementa indice

```



```

115:     }
116: }
117: }
118:
119: //Servicio de interrupcion por desborde del TMR1
120: #int_timer1
121: timer1_service()
122: {
123:     if(cont==200)                //Si contador es 200 (20 segundos)
124:     {
125:         flagC=0;                //Reset bandera de motor de montacarga
126:         cont=0;                 //Reset contador
127:         output_low(pin_d1);     //Pone en bajo las salidas de control Puente H
128:         output_low(pin_d2);
129:         output_low(pin_d3);
130:         indice=0;              //Reset indice
131:         if(derecha==1)
132:         {
133:             derecha=0;         //Sentido de giro del motor -> Derecha
134:         }
135:         else
136:         {
137:             derecha=0;         //Sentido de giro del motor -> Derecha
138:         }
139:         else
140:         {
141:             derecha=1;         //Sentido de giro del motor -> Izquierda
142:         }
143:     }
144: }
145:
146: //Programa Principal
147: void main()
148: {
149:     setup_timer_1(t1_internal | t1_div_by_8);    //Configuración TMR1
150:     enable_interrupts(int_rda); //Habilita interrupcion por recepcion RS232
151:     enable_interrupts(global); //Habilita las interrupciones
152:     while(1)
153:     {
154:         if(s==0)                //Si bandera de inicio de programa esta activada
155:         {
156:             output_high(pin_b7); //Indicador Configuracion Inicial
157:             delay_ms(500);
158:             registro=0;          //Se escribe en la tarjeta MD25
159:             valor=0x80;         //velocidad de stop en motor A
160:             write_md25(registro,valor);
161:             registro=1;         //Se escribe en la tarjeta MD25
162:             valor=0x80;         //velocidad de Stop en motor B
163:             write_md25(registro,valor);
164:             registro=14;        //Se desactiva paro inactividad I2C
165:             valor=2;            //En la tarjeta MD25
166:             write_md25(registro,valor);
167:             delay_ms(500);
168:             registro=16;        //Se resetean registro en MD25
169:             valor=0x32;

```

```

170:         write_md25(registro,valor);
171:         delay_ms(500);
172:         registro=14;           //Se lee la tarjeta MD25
173:         dato=read_md25(registro);
174:         if(dato==2) //Si dato leido es 2
175:         {
176:             s=1;           //Se desactiva bandera de inicio de programa
177:             output_low(pin_b7); //Se apaga indicador de Configuracion Inicial
178:         }
179:     }
180:     else
181:     {
182:         delay_ms(500);
183:         output_toggle(pin_b6);           //Indicador funcionamiento programa
184:         if(flagC==1) //Si bandera montacargas activada
185:         {
186:             set_timer1(3936);           //100 milisegundos
187:             bandera=0;           //Desactiva bandera de RS232
188:             enable_interrupts(int_timer1); //Habilita interrup desborde TMR1
189:             if(derecha==0)           //Giro a la derecha
190:             {
191:                 output_high(pin_d1); //Activa Enable motor montacargas
192:                 output_low(pin_d2); //Gira a la derecha
193:                 output_high(pin_d3); //Gira a la derecha
194:             }
195:             else if(derecha==1)           //Giro a la izquierda
196:             {
197:                 output_high(pin_d1); //Activa Enable motor montacargas
198:                 output_high(pin_d2); //Gira a la izquierda
199:                 output_low(pin_d3); //Gira a la izquierda
200:             }
201:         }
202:     }
203:     {
204:         output_low(pin_d1);           //Desactiva Enable motor montacargas
205:         output_low(pin_d2);
206:         output_low(pin_d3);
207:         disable_interrupts(int_timer1); //Deshabilita interrup del TMR1
208:     }
209: }
210: }
211: }

```

Apéndice C - Datos Prueba 1

Xcobot 1	Ycobot 1	Acobot 1	Xgol C1	Ygol C1	Vel. Angular C1	Vel. Lineal C1	Xcobot 2	Ycobot 2	Acobot 2	Xgol C2	Ygol C2	Vel. Lineal C2	Vel. Angular C2	Muestra
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	1
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	2
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	3
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	4
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	5
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	6
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	7
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	8
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	9
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	10
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	11
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	12
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	13
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	14
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	15
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	16
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	17
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	18
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	19
0.276646286	0.510659456	0.173098505	0.742043316	0.591581941	0.07	0.100998465	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	20
0.282818556	0.510659456	0.17362304	0.742043316	0.591581941	0.07	0.10433391	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	21
0.294888794	0.510659456	0.17362304	0.742043316	0.591581941	0.07	0.10446579	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	22
0.301095098	0.510659456	0.17362304	0.742043316	0.591581941	0.07	0.10669562	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	23
0.307223678	0.510659456	0.17362304	0.742043316	0.591581941	0.07	0.107990602	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	24
0.314741164	0.512203753	3.075621605	0.742043316	0.591581941	0.07	0.077776046	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	25
0.32160306	0.513913155	3.075621605	0.742043316	0.591581941	0.07	0.077181307	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	26
0.330774099	0.515561879	6.452462673	0.742043316	0.591581941	0.07	0.043734376	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	27
0.337839931	0.515561879	6.452462673	0.742043316	0.591581941	0.07	0.043931936	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	28
0.344089925	0.517132878	6.452462673	0.742043316	0.591581941	0.07	0.043357371	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	29
0.353955418	0.51888597	6.452462673	0.742043316	0.591581941	0.07	0.042848508	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	30
0.360470086	0.52045697	9.476168633	0.742043316	0.591581941	0.07	0.011335754	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	31
0.367453396	0.522149384	9.476168633	0.742043316	0.591581941	0.07	0.010730904	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	32
0.373972923	0.522149384	9.476168633	0.742043316	0.591581941	0.07	0.012633882	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	33
0.383733958	0.523902476	9.476168633	0.742043316	0.591581941	0.07	0.012044328	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	34
0.39108631	0.525534153	9.476168633	0.742043316	0.591581941	0.07	0.012375539	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	35
0.398312449	0.527005613	9.476168633	0.742043316	0.591581941	0.07	0.012187514	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	36
0.405405015	0.528763592	9.476168633	0.742043316	0.591581941	0.07	0.01145495	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033587605	37
0.41497913	0.530235052	9.476168633	0.742043316	0.591581941	0.07	0.012013376	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.033770793	38
0.422501445	0.530235052	9.476168633	0.742043316	0.591581941	0.07	0.014569695	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.01749617	39
0.431871593	0.531871617	9.476168633	0.742043316	0.591581941	0.07	0.01409442	0.265139431	0.221285656	358.9215088	0.720102429	0.238211438	0.07	0.018088601	40
0.439168125	0.533624709	9.476168633	0.742043316	0.591581941	0.07	0.014207163	0.265139431	0.222820878	0.511555851	0.720102429	0.238211438	0.07	0.016255092	41
0.446250945	0.535278261	9.476168633	0.742043316	0.591581941	0.07	0.013623682	0.309134292	0.222820878	0.511555851	0.720102429	0.238211438	0.07	0.016648259	42
0.452544659	0.536705971	9.476168633	0.742043316	0.591581941	0.068752546	0.013164291	0.309134292	0.224427983	0.511555851	0.720102429	0.238211438	0.07	0.014787978	43
0.459914058	0.538155615	9.476168633	0.742043316	0.591581941	0.067000111	0.013056049	0.316215694	0.224427983	0.511555851	0.720102429	0.238211438	0.07	0.015111765	44

0.715554655	0.546505451	333.3610535	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	374
0.717870235	0.540938735	333.3610535	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	375
0.719676614	0.537151694	333.3610535	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	376
0.724455178	0.52681762	311.0734863	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	377
0.724455178	0.517191172	301.9959717	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	378
0.724455178	0.51543802	301.9959717	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	379
0.724455178	0.502643824	301.9959717	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	380
0.721335948	0.49853465	285.7086487	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	381
0.716839075	0.488005042	273.6677856	0.145529509	0.411727875	0.07	-0.6	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	382
0.716839075	0.488005042	273.6677856	0.145529509	0.411727875	0.07	-0.59858412	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	383
0.709460974	0.475743562	263.0778503	0.145529509	0.411727875	0.07	-0.58276846	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	384
0.703260958	0.469939798	252.1496735	0.145529509	0.411727875	0.07	-0.548938833	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	385
0.703260958	0.466130644	252.1496735	0.145529509	0.411727875	0.07	-0.55056332	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	386
0.689029753	0.459812075	236.5103455	0.145529509	0.411727875	0.07	-0.50016109	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	387
0.68275547	0.455540985	233.0240479	0.145529509	0.411727875	0.07	-0.4484121	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	388
0.675886273	0.453282803	226.7614594	0.145529509	0.411727875	0.07	-0.40366287	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	389
0.666712761	0.448489696	222.6317444	0.145529509	0.411727875	0.07	-0.37295065	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	390
0.659513414	0.446556926	217.6206055	0.145529509	0.411727875	0.07	-0.3315877	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	391
0.649861634	0.444789231	209.4501953	0.145529509	0.411727875	0.07	-0.2940807	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	392
0.642623365	0.443082243	205.9819031	0.145529509	0.411727875	0.07	-0.25984807	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	393
0.635496751	0.440209776	196.1116943	0.145529509	0.411727875	0.07	-0.19789144	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	394
0.628435791	0.440209776	196.1116943	0.145529509	0.411727875	0.07	-0.16796908	0.640413463	0.238992646	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	395
0.619143367	0.440209776	196.1116943	0.145529509	0.411727875	0.07	-0.12998912	0.644580126	0.239282683	0.475801736	0.720102429	0.238211438	0.018594138	-0.01031927	396
0.611977935	0.440209776	196.1116943	0.145529509	0.411727875	0.07	-0.13123384	0.651162744	0.239282683	360	0.15319702	0.238211438	0.018594138	-0.01031927	397
0.604200661	0.440209776	192.6361389	0.145529509	0.411727875	0.07	-0.09492143	0.658668578	0.235395387	356.8619995	0.15319702	0.238211438	0.018594138	-0.01031927	398
0.597112954	0.438646078	192.6361389	0.145529509	0.411727875	0.07	-0.0961857	0.664923906	0.231412292	348.2528076	0.15319702	0.238211438	0.018594138	-0.01031927	399
0.587594688	0.438646078	189.5032349	0.145529509	0.411727875	0.07	-0.0630909	0.671712995	0.225653931	344.0379944	0.15319702	0.238211438	0.018594138	-0.01031927	400
0.580060244	0.436960965	189.5032349	0.145529509	0.411727875	0.07	-0.06479276	0.67351532	0.225653931	340.3813477	0.15319702	0.238211438	0.018594138	-0.01031927	401
0.572664142	0.436960965	189.5032349	0.145529509	0.411727875	0.07	-0.06418231	0.680390596	0.214700416	330.2240601	0.15319702	0.238211438	0.018594138	-0.01031927	402
0.563135876	0.436960965	186.2213287	0.145529509	0.411727875	0.07	-0.06339393	0.68881912	0.206593871	324.9399719	0.15319702	0.238211438	0.018594138	-0.01031927	403
0.555533707	0.436960965	186.2213287	0.145529509	0.411727875	0.07	-0.02848687	0.68881912	0.202466443	317.2834778	0.15319702	0.238211438	0.018594138	-0.01031927	404
0.548237205	0.435518652	186.2213287	0.145529509	0.411727875	0.07	-0.02780909	0.69066143	0.194633842	317.2834778	0.15319702	0.238211438	0.018594138	-0.01031927	405
0.541253865	0.435518652	186.2213287	0.145529509	0.411727875	0.07	-0.02931262	0.693959892	0.188397646	308.5769348	0.15319702	0.238211438	0.018594138	-0.01031927	406
0.53134954	0.435518652	186.2213287	0.145529509	0.411727875	0.07	-0.02844163	0.690307081	0.17741859	292.3899841	0.15319702	0.238211438	0.018594138	-0.01031927	407
0.524504662	0.435518652	183.2025909	0.145529509	0.411727875	0.07	-0.02774819	0.690307081	0.171469256	287.6597595	0.15319702	0.238211438	0.018594138	-0.01031927	408
0.517498829	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.004547598	0.690307081	0.169228911	284.7389526	0.15319702	0.238211438	0.018594138	-0.01031927	409
0.507391214	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.005298048	0.683398856	0.155635357	270.1836548	0.15319702	0.238211438	0.018594138	-0.01031927	410
0.500325322	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.006381054	0.681284606	0.150779292	266.7016907	0.15319702	0.238211438	0.018594138	-0.01031927	411
0.492890477	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.007201072	0.677556574	0.146815822	266.7016907	0.15319702	0.238211438	0.018594138	-0.01031927	412
0.485290408	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.008098062	0.669880807	0.139338568	249.020874	0.15319702	0.238211438	0.018594138	-0.01031927	413
0.476199348	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.009308223	0.666267574	0.13787822	244.8321075	0.15319702	0.238211438	0.018594138	-0.01031927	414
0.468189448	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.010272373	0.655507681	0.129233167	233.6626434	0.15319702	0.238211438	0.018594138	-0.01031927	415
0.460494369	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.011343139	0.65171949	0.127211049	229.8762665	0.15319702	0.238211438	0.018594138	-0.01031927	416
0.453120112	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.01240293	0.642816067	0.123818636	225.1211243	0.15319702	0.238211438	0.018594138	-0.01031927	417
0.443458647	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.013883434	0.633139133	0.120186768	216.9731445	0.15319702	0.238211438	0.018594138	-0.01031927	418
0.435902268	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.015095089	0.62966238	0.118726052	213.4418793	0.15319702	0.238211438	0.018594138	-0.01031927	419
0.428693175	0.435518652	183.2025909	0.145529509	0.411727875	0.07	0.016327703	0.619875491	0.116903462	210.3078613	0.15319702	0.238211438	0.018594138	-0.01031927	420

0.36811626	0.536179066	47.90016937	0.864920795	0.426483572	0.07	-0.54259634	0.32825424	0.125619933	172.5685883	0.15319702	0.128186703	0.018632601	0.033763106	797
0.375393331	0.537963748	43.99930191	0.864920795	0.426483572	0.07	-0.50190045	0.334304482	0.124020852	172.5685883	0.15319702	0.128186703	0.018632601	0.033763106	798
0.38258791	0.539813995	36.86989594	0.864920795	0.426483572	0.07	-0.48702552	0.338604182	0.124020852	172.5685883	0.15319702	0.128186703	0.018632601	0.033763106	799
0.392499506	0.541664243	29.07941628	0.864920795	0.426483572	0.07	-0.43671301	0.344474372	0.122584067	169.6351776	0.866035402	0.113298856	0.018632601	0.033763106	800
0.40037635	0.543176949	26.07535553	0.864920795	0.426483572	0.07	-0.38510451	0.347432405	0.122584067	169.6351776	0.866035402	0.113298856	0.07	-0.6	801
0.407580614	0.543176949	21.77009201	0.864920795	0.426483572	0.07	-0.35275317	0.353187531	0.122584067	169.6351776	0.866035402	0.113298856	0.07	-0.6	802
0.415557027	0.543176949	16.77233505	0.864920795	0.426483572	0.07	-0.31123439	0.357786641	0.121171236	169.6351776	0.866035402	0.113298856	0.07	-0.6	803
0.432808846	0.541715264	9.666851997	0.864920795	0.426483572	0.07	-0.24900754	0.36141175	0.121171236	169.6351776	0.866035402	0.113298856	0.07	-0.6	805
0.439610034	0.540180683	6.680631161	0.864920795	0.426483572	0.07	-0.22246567	0.36141175	0.121171236	169.6351776	0.866035402	0.113298856	0.07	-0.6	806
0.444665888	0.538619399	2.995903154	0.864920795	0.426483572	0.07	-0.18665577	0.359405173	0.121171236	166.6947632	0.866035402	0.113298856	0.07	-0.6	807
0.4561553	0.537096918	359.669281	0.864920795	0.426483572	0.07	-0.15367934	0.350175589	0.128187537	159.3462067	0.866035402	0.113298856	0.07	-0.6	808
0.462895811	0.535695991	356.0263672	0.864920795	0.426483572	0.07	-0.11594767	0.345679015	0.133703187	155.6561869	0.866035402	0.113298856	0.07	-0.6	809
0.469757676	0.532337785	352.5792236	0.864920795	0.426483572	0.07	-0.1161629	0.340564311	0.139484882	148.0767212	0.866035402	0.113298856	0.07	-0.6	810
0.476437449	0.530402601	352.5792236	0.864920795	0.426483572	0.07	-0.07888849	0.338598877	0.145247981	143.7948914	0.866035402	0.113298856	0.07	-0.6	811
0.483525157	0.528770864	352.5792236	0.864920795	0.426483572	0.07	-0.0806342	0.336209565	0.151857182	135.4907532	0.866035402	0.113298856	0.07	-0.6	812
0.492271245	0.526774943	349.401062	0.864920795	0.426483572	0.07	-0.04593066	0.334424227	0.159884527	128.6736145	0.866035402	0.113298856	0.07	-0.6	813
0.49917686	0.525143266	349.401062	0.864920795	0.426483572	0.07	-0.0461439	0.332995415	0.166751504	123.8875122	0.866035402	0.113298856	0.07	-0.6	814
0.505812943	0.521758437	346.2851868	0.864920795	0.426483572	0.07	-0.01197918	0.332995415	0.173033759	114.6735077	0.866035402	0.113298856	0.07	-0.6	815
0.515005827	0.519922793	346.2851868	0.864920795	0.426483572	0.07	-0.01201561	0.332995415	0.181385681	108.7594833	0.866035402	0.113298856	0.07	-0.6	816
0.521603048	0.517613649	346.2851868	0.864920795	0.426483572	0.07	-0.01117085	0.336323977	0.187763408	99.23104095	0.866035402	0.113298856	0.07	-0.6	817
0.526838203	0.514267633	346.2851868	0.864920795	0.426483572	0.07	-0.01160406	0.338035435	0.194279477	94.27065277	0.866035402	0.113298856	0.07	-0.6	818
0.537657857	0.511880815	346.2851868	0.864920795	0.426483572	0.07	-0.00851182	0.3439989462	0.201790646	81.7749939	0.866035402	0.113298856	0.07	-0.6	819
0.545159499	0.510227263	346.2851868	0.864920795	0.426483572	0.07	-0.00887607	0.348738819	0.207213163	76.56344604	0.866035402	0.113298856	0.07	-0.6	820
0.552076042	0.508833528	346.2851868	0.864920795	0.426483572	0.07	-0.00975832	0.353307247	0.211632609	68.90515137	0.866035402	0.113298856	0.07	-0.59469244	821
0.561654985	0.504396397	346.2851868	0.864920795	0.426483572	0.07	-0.01007932	0.361760318	0.217627168	58.2870636	0.866035402	0.113298856	0.07	-0.57928562	822
0.568599463	0.502957794	346.2851868	0.864920795	0.426483572	0.07	-0.00756917	0.36818859	0.220168188	53.90481949	0.866035402	0.113298856	0.07	-0.56392335	823
0.57568717	0.500912964	346.2851868	0.864920795	0.426483572	0.07	-0.00638216	0.374398679	0.223698929	45.97743225	0.866035402	0.113298856	0.07	-0.52813288	824
0.58285743	0.499055475	346.2851868	0.864920795	0.426483572	0.07	-0.007958277	0.381816745	0.225659847	41.6557579	0.866035402	0.113298856	0.07	-0.51264893	825
0.585144699	0.497601002	346.2851868	0.864920795	0.426483572	0.07	-0.0067357135	0.391408592	0.228730321	33.83713531	0.866035402	0.113298856	0.07	-0.46169287	826
0.599541068	0.494823247	346.2851868	0.864920795	0.426483572	0.07	-0.006440459	0.398741484	0.228730321	26.52677727	0.866035402	0.113298856	0.07	-0.4186219	827
0.606264591	0.492557794	346.2851868	0.864920795	0.426483572	0.07	-0.00522135	0.406327128	0.230302766	22.00708008	0.866035402	0.113298856	0.07	-0.38951456	828
0.613595068	0.489146262	346.2851868	0.864920795	0.426483572	0.07	-0.00437943	0.412704885	0.230302766	18.04646801	0.866035402	0.113298856	0.07	-0.35619024	829
0.621624887	0.487597138	346.2851868	0.864920795	0.426483572	0.07	-0.00290057	0.422730416	0.230302766	14.62262726	0.866035402	0.113298856	0.07	-0.3237562	830
0.628139555	0.48602128	346.2851868	0.864920795	0.426483572	0.07	-0.00278539	0.431106299	0.230302766	10.71422768	0.866035402	0.113298856	0.07	-0.25923182	831
0.634549856	0.484307021	346.2851868	0.864920795	0.426483572	0.07	-0.00271092	0.437798023	0.230302766	6.683546543	0.866035402	0.113298856	0.07	-0.26165375	832
0.641967773	0.480730385	343.3086243	0.864920795	0.426483572	0.07	0.029232873	0.446982741	0.228860691	3.527748108	0.866035402	0.113298856	0.07	-0.22596331	833
0.647848666	0.479241937	343.3086243	0.864920795	0.426483572	0.07	0.033031113	0.454350233	0.227370759	359.7561951	0.866035402	0.113298856	0.07	-0.19566388	834
0.653525651	0.477831215	343.3086243	0.864920795	0.426483572	0.07	0.032901269	0.461137742	0.227370759	359.7561951	0.866035402	0.113298856	0.07	-0.15860528	835
0.659224451	0.476282036	343.3086243	0.864920795	0.426483572	0.07	0.03350107	0.467973083	0.225572035	356.6407471	0.866035402	0.113298856	0.07	-0.15867453	836
0.666312158	0.474849433	343.3086243	0.864920795	0.426483572	0.07	0.0324399747	0.47802788	0.22517969	352.8251038	0.866035402	0.113298856	0.07	-0.12775314	837
0.671173275	0.473013788	343.3086243	0.864920795	0.426483572	0.07	0.046854664	0.484706253	0.22203064	352.8251038	0.866035402	0.113298856	0.07	-0.08950545	838
0.676624417	0.471399099	343.3086243	0.864920795	0.426483572	0.07	0.045603991	0.490708798	0.220200107	352.8251038	0.866035402	0.113298856	0.07	-0.08940781	839
0.681789041	0.469983488	343.3086243	0.864920795	0.426483572	0.07	0.044276406	0.498190731	0.218611643	348.7940369	0.866035402	0.113298856	0.07	-0.04874058	840
0.68781811	0.468147844	343.3086243	0.864920795	0.426483572	0.07	0.042800172	0.50758034	0.215658262	348.7940369	0.866035402	0.113298856	0.07	-0.04973717	841
0.692414522	0.466814784	343.3086243	0.864920795	0.426483572	0.07	0.041728868	0.5143013	0.214008614	348.7940369	0.866035402	0.113298856	0.07	-0.05047186	842
0.697435915	0.466593832	343.3086243	0.864920795	0.426483572	0.07	0.040565147	0.52093178	0.212305784	348.7940369	0.866035402	0.113298856	0.07	-0.05084521	843

0.385188431	0.467820904	159.1306458	0.150798216	0.561539471	0.0606631828	-0.017364	0.728107691	0.245793417	166.9024506	0.143548474	0.269653291	0.07	0.112116401	1220
0.379203081	0.463714996	159.1306458	0.150798216	0.561539471	0.0587633223	-0.01672698	0.713579327	0.245793417	170.4185944	0.143548474	0.269653291	0.07	0.075353836	1221
0.373730093	0.469558567	159.1306458	0.150798216	0.561539471	0.057270823	-0.01830733	0.720097037	0.245793417	170.4185944	0.143548474	0.269653291	0.07	0.075040094	1222
0.366581678	0.472639859	159.1306458	0.150798216	0.561539471	0.055883798	-0.01796162	0.705371976	0.245793417	173.5205994	0.143548474	0.269653291	0.07	0.042447958	1223
0.360982418	0.47472319	159.1306458	0.150798216	0.561539471	0.054032051	-0.01786781	0.699079394	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.042156432	1224
0.355757087	0.476216495	159.1306458	0.150798216	0.561539471	0.052661561	-0.01836159	0.691621423	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.005490026	1225
0.350683746	0.478438228	159.1306458	0.150798216	0.561539471	0.051471276	-0.01974345	0.684363067	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.005140145	1226
0.344233185	0.482014865	159.1306458	0.150798216	0.561539471	0.04994504	-0.01566185	0.674593417	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.004804369	1227
0.339923263	0.484236598	159.1306458	0.150798216	0.561539471	0.048800288	-0.01548732	0.667853355	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.004308801	1228
0.335183561	0.485647321	159.1306458	0.150798216	0.561539471	0.047360237	-0.01588663	0.660861015	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.003940016	1229
0.330669671	0.489202112	156.1848602	0.150798216	0.561539471	0.046030272	-0.01295139	0.650758326	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.003520153	1230
0.325626463	0.490955234	156.1848602	0.150798216	0.561539471	0.045236849	0.019966907	0.642696381	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.003101073	1231
0.321190268	0.492900163	156.1848602	0.150798216	0.561539471	0.043682708	0.01743711	0.635637522	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.00252471	1232
0.31866017	0.494818389	156.1848602	0.150798216	0.561539471	0.042862772	0.01962103	0.628897965	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.0021217	1233
0.312123626	0.496510774	156.1848602	0.150798216	0.561539471	0.041612425	0.019316315	0.618523836	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.00163967	1234
0.309775651	0.498164326	156.1848602	0.150798216	0.561539471	0.040585662	0.019487615	0.610994041	0.245793417	176.9829865	0.143548474	0.269653291	0.07	0.001165438	1235
0.304848969	0.499800891	156.1848602	0.150798216	0.561539471	0.039472974	0.018993491	0.606452227	0.247603253	176.9829865	0.143548474	0.269653291	0.07	0.003035162	1236
0.300908118	0.501520038	156.1848602	0.150798216	0.561539471	0.038405917	0.018692329	0.596549034	0.247603253	176.9829865	0.143548474	0.269653291	0.07	0.002411806	1237
0.296333522	0.503090978	156.1848602	0.150798216	0.561539471	0.037721648	0.02116786	0.586555421	0.247603253	176.9829865	0.143548474	0.269653291	0.07	0.001926847	1238
0.293781549	0.504662037	156.1848602	0.150798216	0.561539471	0.036459471	0.025845399	0.579211891	0.247603253	176.9829865	0.143548474	0.269653291	0.07	0.001442714	1239
0.289063722	0.506463647	156.1848602	0.150798216	0.561539471	0.035392292	0.018763387	0.571591675	0.247603253	176.9829865	0.143548474	0.269653291	0.07	0.00071324	1240
0.285200761	0.507874429	156.1848602	0.150798216	0.561539471	0.034417988	0.019797346	0.5692662	0.247603253	176.9829865	0.143548474	0.269653291	0.07	0.000544854	1241
0.281038761	0.509345889	156.1848602	0.150798216	0.561539471	0.033167948	0.023802088	0.554725409	0.247603253	176.9829865	0.143548474	0.269653291	0.07	-0.00036841	1242
0.278994262	0.510817289	156.1848602	0.150798216	0.561539471	0.032738895	0.020697052	0.547557473	0.247603253	176.9829865	0.143548474	0.269653291	0.07	-0.0011203	1243
0.274072806	0.512822807	159.1723022	0.150798216	0.561539471	0.031735036	0.020084249	0.539985812	0.247603253	176.9829865	0.143548474	0.269653291	0.07	-0.00174391	1244
0.271008164	0.515222807	159.1723022	0.150798216	0.561539471	0.031121996	-0.00910815	0.529600955	0.2490004915	176.9829865	0.143548474	0.269653291	0.07	-0.00026468	1245
0.268764555	0.51732022	159.1723022	0.150798216	0.561539471	0.030317216	-0.01545768	0.526585814	0.2490004915	176.9829865	0.143548474	0.269653291	0.07	-0.00085107	1246
0.265270501	0.513760209	159.1723022	0.150798216	0.561539471	0.029295179	-0.01613024	0.515812576	0.2490004915	176.9829865	0.143548474	0.269653291	0.07	-0.00165204	1247
0.263556212	0.515231609	159.1723022	0.150798216	0.561539471	0.028812956	-0.01253696	0.507859707	0.2490004915	176.9829865	0.143548474	0.269653291	0.07	-0.00237626	1248
0.260470092	0.516659379	159.1723022	0.150798216	0.561539471	0.028116886	-0.01855609	0.498137504	0.2490004915	176.9829865	0.143548474	0.269653291	0.07	-0.00380729	1249
0.255201042	0.518439174	159.1723022	0.150798216	0.561539471	0.026738167	-0.01325436	0.48331207	0.250641257	176.9829865	0.143548474	0.269653291	0.07	-0.00194492	1250
0.252219319	0.519966483	159.1723022	0.150798216	0.561539471	0.025889163	-0.01232826	0.476253211	0.250641257	176.9829865	0.143548474	0.269653291	0.07	-0.00265497	1251
0.249155	0.521437943	159.1723022	0.150798216	0.561539471	0.025222964	-0.01874448	0.46646446	0.250641257	176.9829865	0.143548474	0.269653291	0.07	-0.00342424	1252
0.246945426	0.521437943	159.1723022	0.150798216	0.561539471	0.024784119	-0.0141752	0.459001154	0.252072692	176.9829865	0.143548474	0.269653291	0.07	-0.00181015	1253
0.245313704	0.522953093	159.1723022	0.150798216	0.561539471	0.024173597	-0.01088165	0.451906661	0.252072692	176.9829865	0.143548474	0.269653291	0.07	-0.00258988	1254
0.24165453	0.524524093	159.1723022	0.150798216	0.561539471	0.023485376	-0.0179528	0.442557991	0.252072692	176.9829865	0.143548474	0.269653291	0.07	-0.00338924	1255
0.239818871	0.524524093	159.1723022	0.150798216	0.561539471	0.022891663	-0.01401613	0.435166538	0.252072692	176.9829865	0.143548474	0.269653291	0.068741619	-0.10863404	1257
0.23773554	0.526034355	159.1723022	0.150798216	0.561539471	0.022362648	-0.00959672	0.427437216	0.252072692	176.9829865	0.143548474	0.269653291	0.066367603	-0.1134134	1258
0.235834315	0.52748394	159.1723022	0.150798216	0.561539471	0.021911865	-0.01452696	0.419236928	0.253504157	176.9829865	0.143548474	0.269653291	0.064437575	-0.10594227	1259
0.234384701	0.52748394	159.1723022	0.150798216	0.561539471	0.021060165	-0.01402796	0.410613537	0.255004157	176.9829865	0.143548474	0.269653291	0.062903081	-0.10914793	1260
0.234384701	0.52748394	159.1723022	0.150798216	0.561539471	0.021060165	-0.01402796	0.403735638	0.255201697	176.9829865	0.143548474	0.269653291	0.060803913	-0.09912221	1261
0.234384701	0.52748394	159.1723022	0.150798216	0.561539471	0.021060165	-0.01402796	0.396724671	0.255201697	176.9829865	0.143548474	0.269653291	0.059170608	-0.1025854	1262
0.232935116	0.528933585	159.1723022	0.150798216	0.561539471	0.021060165	-0.01402796	0.388276935	0.256917834	173.6598053	0.143548474	0.269653291	0.057681528	-0.05561844	1263
0.232935116	0.528933585	159.1723022	0.150798216	0.561539471	0.020620138	-0.00862752	0.381537378	0.256548856	173.6598053	0.143548474	0.269653291	0.056096586	-0.04329956	1264
0.232935116	0.528933585	159.1723022	0.150798216	0.561539471	0.020620138	-0.00862752	0.375686467	0.260179877	173.6598053	0.143548474	0.269653291	0.05466094	-0.04611371	1265
0.232935116	0.528933585	159.1723022	0.150798216	0.561539471	0.020620138	-0.00862752	0.370239973	0.260179877	173.6598053	0.143548474	0.269653291	0.052940851	-0.03296596	1266

0.404135078	0.559697926	26.15096092	0.600000024	0.540000021	0.046750382	-0.35233281	0.238628134	0.269191653	176.4025574	0.143548474	0.269653291	0.018536323	-0.00774483	1596
0.409867913	0.559697926	21.91039658	0.600000024	0.540000021	0.045562572	-0.31742376	0.242861131	0.269191653	176.4025574	0.143548474	0.269653291	0.018536323	-0.00774483	1597
0.416382521	0.559697926	18.48899923	0.600000024	0.540000021	0.044151336	-0.28062591	0.248376406	0.269191653	176.4025574	0.143548474	0.269653291	0.018536323	-0.00774483	1598
0.421542346	0.559697926	15.4885006	0.600000024	0.540000021	0.042749657	-0.25034834	0.251327693	0.269191653	176.4025574	0.143548474	0.269653291	0.018536323	-0.00774483	1599
0.426444225	0.559697926	12.55600929	0.600000024	0.540000021	0.041537688	-0.22322252	0.257130682	0.269191653	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1600
0.431138307	0.559697926	12.55600929	0.600000024	0.540000021	0.040387519	-0.19623709	0.260964781	0.269191653	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1601
0.437305748	0.558143973	9.366509438	0.600000024	0.540000021	0.039251834	-0.16629786	0.26690808	0.269191653	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1602
0.439776937	0.558143973	6.377180576	0.600000024	0.540000021	0.038197337	-0.1626621	0.27106747	0.269191653	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1603
0.445330739	0.556422412	3.32236528	0.600000024	0.540000021	0.037004994	-0.13442508	0.275210202	0.269191653	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1604
0.449332267	0.556422412	3.32236528	0.600000024	0.540000021	0.036292936	-0.09792376	0.279757351	0.267691046	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1605
0.453807324	0.554982543	359.8329468	0.600000024	0.540000021	0.035364026	-0.09597355	0.285451263	0.267691046	173.3469696	0.143548474	0.269653291	0.018536323	-0.00774483	1606
0.456419975	0.554982543	359.8329468	0.600000024	0.540000021	0.034290301	-0.05942993	0.290174007	0.267691046	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1607
0.460994954	0.55309099	359.8329468	0.600000024	0.540000021	0.033153142	-0.06152871	0.2949497669	0.266177088	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1608
0.465491444	0.55309099	356.0148621	0.600000024	0.540000021	0.032578128	-0.05451472	0.300276726	0.266177088	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1609
0.467713207	0.551629305	356.0148621	0.600000024	0.540000021	0.031533627	-0.01647698	0.304533839	0.264655203	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1610
0.472370327	0.551629305	356.0148621	0.600000024	0.540000021	0.030483517	-0.01183211	0.307758629	0.264655203	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1611
0.474397838	0.55022341	356.0148621	0.600000024	0.540000021	0.029903631	-0.01278664	0.313380688	0.264655203	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1612
0.477947742	0.55022341	356.0148621	0.600000024	0.540000021	0.02902524	-0.01444092	0.318151355	0.263037443	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1613
0.482483506	0.55022341	356.0148621	0.600000024	0.540000021	0.027994294	-0.00945678	0.324002236	0.263037443	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1614
0.484544963	0.55022341	356.0148621	0.600000024	0.540000021	0.027504087	-0.01039334	0.328144968	0.261619329	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1615
0.487878799	0.54879564	356.0148621	0.600000024	0.540000021	0.026679797	-0.01198505	0.332468599	0.261619329	169.900238	0.143548474	0.269653291	0.018536323	-0.00774483	1616
0.491654545	0.54879564	356.0148621	0.600000024	0.540000021	0.026241995	-0.00523991	0.338851631	0.259829793	166.2539368	0.600000024	0.150000006	0.066464985	0.6	1617
0.493572742	0.54879564	356.0148621	0.600000024	0.540000021	0.025363779	-0.00686974	0.342951804	0.258437067	166.2539368	0.600000024	0.150000006	0.065096397	0.6	1618
0.497248918	0.54879564	356.0148621	0.600000024	0.540000021	0.024508916	-0.00856874	0.347541511	0.258437067	166.2539368	0.600000024	0.150000006	0.064111023	0.6	1619
0.500252545	0.54879564	356.0148621	0.600000024	0.540000021	0.024062938	-0.00950316	0.349896252	0.258437067	166.2539368	0.600000024	0.150000006	0.063606552	0.6	1620
0.502479136	0.547346056	356.0148621	0.600000024	0.540000021	0.023373444	-0.00237589	0.349896252	0.258437067	166.2539368	0.600000024	0.150000006	0.063606552	0.6	1621
0.504193366	0.547346056	356.0148621	0.600000024	0.540000021	0.022420504	-0.00418322	0.349896252	0.258437067	166.2539368	0.600000024	0.150000006	0.063606552	0.6	1622
0.508360028	0.547346056	356.0148621	0.600000024	0.540000021	0.021906475	-0.00526277	0.344298095	0.257029623	169.6287231	0.600000024	0.150000006	0.064679569	0.6	1623
0.508360028	0.547346056	356.0148621	0.600000024	0.540000021	0.021451257	-0.00626212	0.333615363	0.253975064	177.0176697	0.600000024	0.150000006	0.065664284	0.6	1624
0.508360028	0.547346056	356.0148621	0.600000024	0.540000021	0.021451257	-0.00626212	0.329395822	0.252282888	177.0176697	0.600000024	0.150000006	0.067509093	0.6	1625
0.508360028	0.547346056	356.0148621	0.600000024	0.540000021	0.021451257	-0.00626212	0.323603123	0.249749899	184.4519043	0.600000024	0.150000006	0.068563994	0.6	1626
0.51001358	0.547346056	356.0148621	0.600000024	0.540000021	0.021451257	-0.00626212	0.315573126	0.244872853	192.1859894	0.600000024	0.150000006	0.069960912	0.6	1627
0.51001358	0.547346056	356.0148621	0.600000024	0.540000021	0.021066685	-0.00714013	0.308974564	0.237156749	200.9151306	0.600000024	0.150000006	0.07	0.6	1628
0.51001358	0.547346056	356.0148621	0.600000024	0.540000021	0.021066685	-0.00714013	0.304987133	0.231712982	207.4175415	0.600000024	0.150000006	0.07	0.6	1629
0.511446238	0.547346056	356.0148621	0.600000024	0.540000021	0.020792717	-0.00792717	0.301600456	0.225750327	210.5028992	0.600000024	0.150000006	0.07	0.6	1630
0.511446238	0.545896471	356.0148621	0.600000024	0.540000021	0.020739524	-0.00792717	0.296396583	0.218204737	217.836441	0.600000024	0.150000006	0.07	0.6	1631
0.511446238	0.545896471	356.0148621	0.600000024	0.540000021	0.020708305	-0.001839507	0.29472369	0.210115552	217.836441	0.600000024	0.150000006	0.07	0.6	1632
0.512895823	0.545896471	356.0148621	0.600000024	0.540000021	0.020370829	0.001177619	0.291211158	0.205244824	217.836441	0.600000024	0.150000006	0.07	0.6	1633
0.512895823	0.545896471	353.0788574	0.600000024	0.540000021	0.020370829	0.001177619	0.290862948	0.196267962	238.6010437	0.600000024	0.150000006	0.07	0.6	1634
0.512895823	0.545896471	353.0788574	0.600000024	0.540000021	0.020370829	0.031908329	0.293269038	0.187276244	238.6010437	0.600000024	0.150000006	0.07	0.6	1635
0.51442796	0.545896471	353.0788574	0.600000024	0.540000021	0.020014161	0.031185482	0.293269038	0.180463433	252.8871155	0.600000024	0.150000006	0.07	0.6	1636
0.51442796	0.545896471	353.0788574	0.600000024	0.540000021	0.020014161	0.031185482	0.293269038	0.172834113	262.1997986	0.600000024	0.150000006	0.07	0.6	1637
0.51442796	0.545896471	353.0788574	0.600000024	0.540000021	0.020014161	0.031185482	0.292973929	0.167813063	266.9199829	0.600000024	0.150000006	0.07	0.6	1638
0.515938822	0.545896471	353.0788574	0.600000024	0.540000021	0.020014161	0.031185482	0.305873632	0.158415914	266.9199829	0.600000024	0.150000006	0.069029121	0.595980343	1639
0.515938822	0.545896471	353.0788574	0.600000024	0.540000021	0.019662615	0.030447246	0.311342597	0.154458001	286.3268433	0.600000024	0.150000006	0.067361431	0.564308801	1640
0.515938822	0.545896471	353.0788574	0.600000024	0.540000021	0.019662615	0.030447246	0.314998388	0.150915265	291.3988511	0.600000024	0.150000006	0.066500725	0.555780708	1641
0.517509222	0.545896471	353.0788574	0.600000024	0.540000021	0.019662615	0.030447246	0.320508718	0.144565031	295.9923096	0.600000024	0.150000006	0.065226967	0.558079271	1642

0.780778944	0.485980004	108.0641937	0.129664436	0.256398082	0.07	0.6	0.709410906	0.225098126	4.423353672	0.136914641	0.543906987	0.07	0.6	468
0.776709437	0.482998987	113.9988368	0.129664436	0.256398082	0.07	0.6	0.712388277	0.225098126	4.423353672	0.136914641	0.543906987	0.07	0.6	469
0.770274878	0.492421836	121.1930008	0.129664436	0.256398082	0.07	0.598453777	0.723871625	0.230042055	6.169808388	0.136914641	0.543906987	0.07	0.6	470
0.762898266	0.497545153	125.1924057	0.129664436	0.256398082	0.07	0.589525785	0.728666425	0.235908911	16.3319931	0.136914641	0.543906987	0.07	0.6	471
0.757600069	0.499798477	129.3814087	0.129664436	0.256398082	0.07	0.56912949	0.732724011	0.239612624	19.90703011	0.136914641	0.543906987	0.07	0.6	472
0.747550011	0.501687527	138.0315247	0.129664436	0.256398082	0.07	0.556290454	0.739170909	0.247514889	27.67386246	0.136914641	0.543906987	0.07	0.6	473
0.739170611	0.503146827	145.7674255	0.129664436	0.256398082	0.07	0.522025695	0.743172646	0.254033595	34.092233856	0.136914641	0.543906987	0.07	0.6	474
0.722263515	0.505011678	154.6657257	0.129664436	0.256398082	0.07	0.46952275	0.747975223	0.268090129	44.67123413	0.136914641	0.543906987	0.07	0.6	475
0.714182734	0.505011678	162.4662476	0.129664436	0.256398082	0.07	0.417474412	0.7494403954	0.274928182	52.45985794	0.136914641	0.543906987	0.07	0.6	476
0.706648231	0.505011678	166.1028137	0.129664436	0.256398082	0.07	0.390268047	0.752877295	0.283595562	56.40498734	0.136914641	0.543906987	0.07	0.6	477
0.699074864	0.503285289	170.5912323	0.129664436	0.256398082	0.07	0.3622333457	0.750274062	0.291948706	65.34148407	0.136914641	0.543906987	0.07	0.6	478
0.688966572	0.499864012	176.1642151	0.129664436	0.256398082	0.07	0.325460939	0.748576522	0.2988213	75.18230438	0.136914641	0.543906987	0.07	0.6	479
0.682209074	0.498220176	179.1803589	0.129664436	0.256398082	0.07	0.276556146	0.746000946	0.304713011	81.06243134	0.136914641	0.543906987	0.07	0.6	480
0.675893545	0.495313734	182.9957581	0.129664436	0.256398082	0.07	-0.08102515	0.743648887	0.31277141	87.01760864	0.136914641	0.543906987	0.07	0.6	481
0.666232049	0.491304874	186.2140198	0.129664436	0.256398082	0.07	-0.44021044	0.738396645	0.320971698	98.69242096	0.136914641	0.543906987	0.07	0.6	482
0.659491539	0.488393545	189.2088623	0.129664436	0.256398082	0.07	-0.51699435	0.73380959	0.328462511	103.2379684	0.136914641	0.543906987	0.07	0.6	483
0.652993917	0.486480176	193.3057709	0.129664436	0.256398082	0.07	-0.5484988	0.726473987	0.330076605	111.6267014	0.136914641	0.543906987	0.07	0.6	484
0.646131992	0.4945008746	193.3057709	0.129664436	0.256398082	0.07	-0.56758106	0.721417692	0.334211379	116.5259171	0.136914641	0.543906987	0.07	0.6	485
0.635841608	0.485008746	189.3006592	0.129664436	0.256398082	0.07	-0.56190879	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	486
0.628224552	0.486975521	181.8659058	0.129664436	0.256398082	0.07	-0.51300353	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	487
0.620932877	0.488619357	178.8687744	0.129664436	0.256398082	0.07	-0.43509338	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	488
0.611531198	0.49470702	171.7654419	0.129664436	0.256398082	0.07	-0.35298263	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	489
0.605011642	0.497360617	164.8824615	0.129664436	0.256398082	0.07	-0.145369652	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	490
0.598960757	0.501490891	160.3350372	0.129664436	0.256398082	0.07	0.404226136	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	491
0.593725741	0.505543411	157.109436	0.129664436	0.256398082	0.07	0.444577306	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	492
0.585717797	0.510355949	157.109436	0.129664436	0.256398082	0.07	0.469968388	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	493
0.579241131	0.514517784	154.0740509	0.129664436	0.256398082	0.07	0.4916818077	0.718220532	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	494
0.571751177	0.515994072	167.27565	0.129664436	0.256398082	0.07	0.496886513	0.716812968	0.335786492	122.8356994	0.136914641	0.543906987	0.07	0.6	495
0.562810838	0.515994072	160.9154663	0.129664436	0.256398082	0.07	0.500345403	0.716812968	0.337670296	122.8356994	0.136914641	0.543906987	0.07	0.6	496
0.552911341	0.515994072	167.27565	0.129664436	0.256398082	0.07	0.438964455	0.704696119	0.347360581	130.3042145	0.136914641	0.543906987	0.07	0.6	497
0.54517287	0.514146268	171.7654419	0.129664436	0.256398082	0.07	0.419727226	0.698970258	0.350343257	130.3042145	0.136914641	0.543906987	0.07	0.6	498
0.537351906	0.511895418	176.4697266	0.129664436	0.256398082	0.07	0.388723952	0.692887902	0.355329394	134.1939087	0.136914641	0.543906987	0.07	0.6	499
0.527304292	0.508299351	179.5836029	0.129664436	0.256398082	0.07	0.323525953	0.683804274	0.359863251	140.1137848	0.136914641	0.543906987	0.07	0.6	500
0.520656781	0.50634712	183.6592712	0.129664436	0.256398082	0.07	0.28928884	0.676761389	0.361951888	144.1029358	0.136914641	0.543906987	0.07	0.6	501
0.513862193	0.502867579	187.8533173	0.129664436	0.256398082	0.07	0.250502662	0.670269251	0.365682214	144.1029358	0.136914641	0.543906987	0.07	0.6	502
0.508097768	0.498754382	191.51828	0.129664436	0.256398082	0.07	0.216815458	0.663330138	0.367323846	147.4529419	0.136914641	0.543906987	0.07	0.6	503
0.500167549	0.494051069	194.5015564	0.129664436	0.256398082	0.07	0.186963709	0.654198587	0.370450199	150.734314	0.136914641	0.543906987	0.07	0.6	504
0.496022701	0.491217464	197.9670105	0.129664436	0.256398082	0.07	-0.05047136	0.647216856	0.372496277	153.6374664	0.136914641	0.543906987	0.07	0.6	505
0.496022701	0.491217464	197.9670105	0.129664436	0.256398082	0.07	-0.35669427	0.640543818	0.373914421	153.6374664	0.136914641	0.543906987	0.07	-0.06210937	506
0.479754269	0.480859071	201.520401	0.129664436	0.256398082	0.07	-0.39404022	0.63279587	0.37714985	157.302124	0.136914641	0.543906987	0.07	-0.08640276	507
0.479754269	0.480859071	201.520401	0.129664436	0.256398082	0.07	-0.42036267	0.626189351	0.378815442	160.6389313	0.136914641	0.543906987	0.07	-0.09105118	508
0.479754269	0.480859071	201.520401	0.129664436	0.256398082	0.07	-0.4506855	0.622969866	0.378815442	160.6389313	0.136914641	0.543906987	0.07	-0.14028817	509
0.479754269	0.480859071	201.520401	0.129664436	0.256398082	0.07	-0.45606853	0.622969866	0.378815442	160.6389313	0.136914641	0.543906987	0.07	-0.14462703	510
0.449395418	0.47333917	196.1969299	0.129664436	0.256398082	0.07	-0.19475489	0.622969866	0.378815442	160.6389313	0.136914641	0.543906987	0.07	-0.05633247	511
0.442225128	0.47333917	189.6013489	0.129664436	0.256398082	0.07	0.192356473	0.622969866	0.378815442	160.6389313	0.136914641	0.543906987	0.07	0.0011138389	512
0.43462644	0.474737763	188.0566864	0.129664436	0.256398082	0.07	0.257196502	0.622969866	0.378815442	160.6389313	0.136914641	0.543906987	0.07	0.006290482	513
0.427678005	0.476680249	182.8582001	0.129664436	0.256398082	0.07	0.328291707	0.622969866	0.378815442	160.6389313	0.136914641	0.543906987	0.07	0.006290482	514

0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1267
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1268
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1269
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1270
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1271
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1272
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1273
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1274
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1275
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1276
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1277
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1278
0.781318009	0.120541967	337.133606	0.854969382	0.092243396	0.018532699	0.009254122	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1279
0.78340131	0.120541967	337.133606	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1280
0.79102541	0.120541967	340.3293762	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1281
0.801238358	0.119121514	344.1914368	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1282
0.809506118	0.122698106	352.2684021	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1283
0.816979885	0.124766864	356.5516357	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1284
0.823802948	0.12893112	3.165972471	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1285
0.825968862	0.12893112	3.165972471	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1286
0.838616967	0.139508054	14.99673229	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1287
0.84192884	0.143334791	17.82740784	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1288
0.847025573	0.151257798	28.01237488	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1289
0.85196197	0.160040334	35.25035858	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1290
0.855800807	0.167399958	39.41739273	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1291
0.857473731	0.173557714	47.3964386	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1292
0.857473731	0.183364913	56.1860733	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1293
0.857473731	0.194721357	68.01212311	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1294
0.855754673	0.204912111	75.12151515	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1295
0.851760328	0.21140492	81.98808289	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1296
0.847088695	0.219798967	90.37508392	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1297
0.842169285	0.225927547	98.62413025	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1298
0.836638033	0.230980471	108.8446121	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1300
0.828775704	0.236315057	114.8195038	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1301
0.822384894	0.24165456	123.1756668	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1302
0.815615237	0.246005252	128.0317383	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1303
0.806259751	0.24974747	135.4992981	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1304
0.799407542	0.251488447	138.553421	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1305
0.79204303	0.255118489	141.4506073	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1306
0.784518242	0.256881326	145.3590393	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1307
0.777517974	0.258318752	148.3868561	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1308
0.768065333	0.260103434	151.8635406	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1309
0.7611669374	0.261667162	156.3998108	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1310
0.75206393	0.263082743	160.1114197	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1311
0.744755208	0.264602751	160.1114197	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1312
0.737303317	0.266222358	164.0209198	0.13153097	0.395833343	0.07	0.6	0.848494053	0.612646341	0.825190783	0.927146494	0.612519443	0.01835226	-0.01251259	1313

0.388512552	0.449067622	38.30110931	0.600000024	0.540000021	0.054021379	-0.15705705	0.23817049	0.120141543	187.2787476	0.131669238	0.114529334	0.018636612	-0.00232411	1737
0.394002527	0.452119768	34.17788696	0.600000024	0.540000021	0.052257243	-0.11524996	0.241823643	0.121549048	187.2787476	0.131669238	0.114529334	0.018636612	-0.00232411	1738
0.399205983	0.453678638	34.17788696	0.600000024	0.540000021	0.050997995	-0.11361163	0.246056825	0.121549048	184.2737885	0.131669238	0.114529334	0.018636612	-0.00232411	1739
0.406288862	0.456764787	30.39471054	0.600000024	0.540000021	0.04954162	-0.07638574	0.250694454	0.123156153	184.2737885	0.131669238	0.114529334	0.018636612	-0.00232411	1740
0.410637617	0.458670855	30.39471054	0.600000024	0.540000021	0.048087365	-0.07469805	0.255199015	0.123156153	184.2737885	0.131669238	0.114529334	0.018636612	-0.00232411	1741
0.415455043	0.460676461	27.33407402	0.600000024	0.540000021	0.046869853	-0.04263143	0.259432197	0.124654092	184.2737885	0.131669238	0.114529334	0.018636612	-0.00232411	1742
0.420658499	0.463684946	27.33407402	0.600000024	0.540000021	0.045477483	-0.04480866	0.263998859	0.124654092	184.2737885	0.131669238	0.114529334	0.018636612	-0.00232411	1743
0.426357729	0.4653338469	27.33407402	0.600000024	0.540000021	0.044663861	-0.04578726	0.269045353	0.124654092	184.2737885	0.131669238	0.114529334	0.018636612	-0.00232411	1744
0.430606544	0.467074603	27.33407402	0.600000024	0.540000021	0.043032302	-0.04229013	0.273188055	0.124654092	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1745
0.434035063	0.46855072	24.46514702	0.600000024	0.540000021	0.042165336	-0.01214867	0.277445197	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1746
0.439529926	0.470347703	24.46514702	0.600000024	0.540000021	0.040818062	-0.0104904	0.281744897	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1747
0.444738269	0.473985046	24.46514702	0.600000024	0.540000021	0.040128474	-0.01261586	0.285983413	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1748
0.448778659	0.475494631	24.46514702	0.600000024	0.540000021	0.038366555	-0.01407964	0.291605473	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1749
0.452945322	0.477253288	24.46514702	0.600000024	0.540000021	0.037305788	-0.01421649	0.295639098	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1750
0.45551914	0.4786846163	24.46514702	0.600000024	0.540000021	0.036607703	-0.01595588	0.298326403	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1751
0.460193306	0.480400145	24.46514702	0.600000024	0.540000021	0.03546211	-0.0144133	0.305337369	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1752
0.465241373	0.481932312	24.46514702	0.600000024	0.540000021	0.034238638	-0.01208241	0.309413582	0.12626119	180.9577484	0.131669238	0.114529334	0.018636612	-0.00232411	1753
0.46728211	0.483403772	24.46514702	0.600000024	0.540000021	0.033591437	-0.01377748	0.313694656	0.12626119	180.9577484	0.600000024	0.150000006	0.018636612	-0.00232411	1754
0.472246498	0.485426366	21.53687286	0.600000024	0.540000021	0.03241507	0.016693265	0.317861319	0.12626119	180.9577484	0.600000024	0.150000006	0.066064978	-0.6	1755
0.477250874	0.486897826	21.53687286	0.600000024	0.540000021	0.031206706	0.01944099	0.322979345	0.12626119	177.4701691	0.600000024	0.150000006	0.065023238	-0.6	1756
0.479025811	0.488490671	21.53687286	0.600000024	0.540000021	0.030679539	0.015987169	0.32792145	0.12626119	177.4701691	0.600000024	0.150000006	0.064065737	-0.6	1757
0.482452771	0.488490671	21.53687286	0.600000024	0.540000021	0.029861606	0.022998648	0.330542237	0.12626119	177.4701691	0.600000024	0.150000006	0.063117003	-0.6	1758
0.487072676	0.489883976	21.53687286	0.600000024	0.540000021	0.029311961	0.020158856	0.332178593	0.12765272	177.4701691	0.600000024	0.150000006	0.062708836	-0.6	1759
0.488368998	0.491780788	21.53687286	0.600000024	0.540000021	0.028244885	0.020288354	0.332178593	0.12765272	177.4701691	0.600000024	0.150000006	0.062708836	-0.6	1760
0.492385537	0.493373632	21.53687286	0.600000024	0.540000021	0.027365663	0.019776434	0.332178593	0.12765272	177.4701691	0.600000024	0.150000006	0.062708836	-0.6	1761
0.494099647	0.494966477	21.53687286	0.600000024	0.540000021	0.026851497	0.015711191	0.325186253	0.131061077	173.4327698	0.600000024	0.150000006	0.064275305	-0.6	1762
0.497307211	0.496537477	21.53687286	0.600000024	0.540000021	0.026019342	0.014686954	0.318560964	0.137502677	165.9117126	0.600000024	0.150000006	0.065767136	-0.6	1763
0.499286145	0.496537477	21.53687286	0.600000024	0.540000021	0.02559474	0.018903875	0.314083129	0.13979688	160.966156	0.600000024	0.150000006	0.066165121	-0.6	1764
0.503110409	0.498334318	21.53687286	0.600000024	0.540000021	0.024609343	0.018139025	0.309658378	0.146336198	153.5502319	0.600000024	0.150000006	0.067751778	-0.6	1765
0.505023777	0.499783903	21.53687286	0.600000024	0.540000021	0.024065959	0.014790699	0.304773688	0.152812362	146.8469849	0.600000024	0.150000006	0.06888927	-0.6	1766
0.506925046	0.501316071	21.53687286	0.600000024	0.540000021	0.023518564	0.010805891	0.303381741	0.159535959	142.9207306	0.600000024	0.150000006	0.06923454	-0.6	1767
0.511356354	0.501316071	21.53687286	0.600000024	0.540000021	0.022567264	0.021353125	0.300202221	0.165950939	135.3141174	0.600000024	0.150000006	0.069656897	-0.6	1768
0.51350528	0.502787471	21.53687286	0.600000024	0.540000021	0.021970678	0.01823839	0.300202221	0.173145488	129.6811066	0.600000024	0.150000006	0.07	-0.6	1769
0.51493299	0.504275918	21.53687286	0.600000024	0.540000021	0.021528224	0.013017881	0.298770756	0.181643799	119.112114	0.600000024	0.150000006	0.07	-0.6	1770
0.517016292	0.505808055	21.53687286	0.600000024	0.540000021	0.020942093	0.008967481	0.298770756	0.189072445	114.8827438	0.600000024	0.150000006	0.07	-0.6	1771
0.519264758	0.505808055	21.53687286	0.600000024	0.540000021	0.02045799	0.014828214	0.300486892	0.194242239	106.8597488	0.600000024	0.150000006	0.07	-0.6	1772
0.519264758	0.507213995	21.53687286	0.600000024	0.540000021	0.020332305	0.009519327	0.301990747	0.200131178	106.8597488	0.600000024	0.150000006	0.07	-0.6	1773
0.52067548	0.507213995	21.53687286	0.600000024	0.540000021	0.020027708	0.00962541	0.307101429	0.20973821	90.04727173	0.600000024	0.150000006	0.07	-0.6	1774
0.52067548	0.507213995	21.53687286	0.600000024	0.540000021	0.020027708	0.00962541	0.310727954	0.21421881	83.21324158	0.600000024	0.150000006	0.069414037	-0.6	1775
0.52067548	0.507213995	21.53687286	0.600000024	0.540000021	0.020027708	0.00962541	0.313737243	0.218161985	79.16559497	0.600000024	0.150000006	0.068662049	-0.6	1776
0.52067548	0.507213995	21.53687286	0.600000024	0.540000021	0.020027708	0.00962541	0.320692331	0.223124191	69.35046387	0.600000024	0.150000006	0.067368281	-0.6	1777
0.52210325	0.507213995	21.53687286	0.600000024	0.540000021	0.019720234	0.013496477	0.328251392	0.228557393	61.66412735	0.600000024	0.150000006	0.066489331	-0.6	1778
0.52210325	0.507213995	21.53687286	0.600000024	0.540000021	0.019720234	0.013496477	0.335155904	0.232398851	53.85041428	0.600000024	0.150000006	0.065197892	-0.6	1779
0.523613513	0.508663535	21.53687286	0.600000024	0.540000021	0.019265019	0.008045634	0.350904644	0.236834914	39.96128082	0.600000024	0.150000006	0.062082847	-0.6	1781
0.523613513	0.508663535	21.53687286	0.600000024	0.540000021	0.019265019	0.008045634	0.358717993	0.23784323	31.78671646	0.600000024	0.150000006	0.060378541	-0.59921888	1782
0.523613513	0.508663535	21.53687286	0.600000024	0.540000021	0.019265019	0.008045634	0.364495516	0.23784323	28.14697266	0.600000024	0.150000006	0.059094475	-0.59680763	1783

0.523613513	0.508663535	21.53687286	0.540000024	0.540000021	0.019265019	0.008045634	0.371628881	0.23784323	19.31890106	0.150000006	0.057092719	-0.58292319	1784
0.523613513	0.508663535	21.53687286	0.600000024	0.600000021	0.019265019	0.008045634	0.381268862	0.236390471	10.87256813	0.150000006	0.055418254	-0.55726293	1785
0.525167525	0.508663535	21.53687286	0.600000024	0.540000021	0.018930048	0.012407057	0.387138158	0.232798502	7.119709969	0.150000006	0.0538782	-0.54237222	1786
0.525167525	0.508663535	21.53687286	0.600000024	0.540000021	0.018930048	0.012407057	0.395053893	0.230906677	4.091622353	0.150000006	0.051412193	-0.52851666	1787
0.525167525	0.508663535	21.53687286	0.600000024	0.540000021	0.018930048	0.012407057	0.398762763	0.226838544	356.7689819	0.150000006	0.050261861	-0.48265389	1788
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.404094815	0.222741053	348.0463257	0.150000006	0.049574525	-0.44883411	1789
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.410298318	0.218193963	342.7094727	0.150000006	0.047036871	-0.36851183	1790
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.412281841	0.214444869	338.5648193	0.150000006	0.046310214	-0.3224001	1791
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.416145146	0.209690288	334.8435669	0.150000006	0.045103729	-0.27468682	1792
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.420165509	0.203975081	328.2731018	0.150000006	0.04439249	-0.23090846	1793
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.423496723	0.196761921	321.1936646	0.150000006	0.043088935	-0.12736238	1794
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.426623017	0.192020535	317.9967041	0.150000006	0.042604962	-0.08199988	1795
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.432349672	0.187194034	314.2425537	0.150000006	0.04162585	-0.02207318	1796
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.437712848	0.17414327	310.9055176	0.150000006	0.041068331	-0.007805	1797
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.438949204	0.170761511	310.9055176	0.150000006	0.037996922	0.222416755	1802
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.4438245	0.168944195	310.9055176	0.150000006	0.03670807	0.2364778	1803
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.44785279	0.165557146	310.9055176	0.150000006	0.036133298	0.273338695	1804
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.450492233	0.163378	314.3208618	0.150000006	0.03502453	0.26288887	1805
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.455414534	0.160238385	314.3208618	0.150000006	0.033821092	0.298770156	1806
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.459256619	0.158844188	317.3342285	0.150000006	0.032904903	0.286814326	1807
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.464245439	0.157444671	320.6951294	0.150000006	0.032306395	0.273790404	1808
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.466552258	0.155826971	320.6951294	0.150000006	0.031167482	0.294520181	1809
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.4726120174	0.1542949372	328.9278564	0.150000006	0.028922661	0.236150672	1811
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.481175482	0.152799055	331.9176636	0.150000006	0.027733418	0.231539339	1812
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.483391881	0.152799055	331.9176636	0.150000006	0.027216404	0.23055653	1813
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.488271594	0.152799055	335.2248535	0.150000006	0.026078147	0.195862581	1814
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.490421474	0.152799055	338.1748962	0.150000006	0.025576668	0.165293314	1815
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.494668544	0.152799055	341.9252014	0.150000006	0.024581823	0.124811593	1816
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.498895824	0.152799055	341.9252014	0.150000006	0.023600019	0.122243059	1817
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.502523652	0.152799055	346.3839417	0.150000006	0.023239462	0.075195912	1818
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.504427433	0.152799055	346.3839417	0.150000006	0.022816836	0.073970573	1819
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.507066846	0.152799055	346.3839417	0.150000006	0.021694241	0.070485145	1820
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.152799055	349.3753357	0.150000006	0.021125207	0.037377946	1821
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.510938168	0.15436089	349.3753357	0.150000006	0.020805997	-0.00478495	1822
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1823
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1824
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1825
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1826
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1827
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.512779355	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1828
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.514705715	0.15436089	349.3753357	0.150000006	0.020376911	-0.007172	1829
0.526942492	0.51011312	21.53687286	0.600000024	0.540000021	0.018418012	0.007455528	0.514705715	0.15436089	352.4026489	0.150000006	0.019928	-0.00977533	1830

0.25202021	0.531189322	359.2904663	0.689192414	0.331908017	0.07	-0.24208829	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	92
0.25202021	0.531189322	359.8226013	0.689192414	0.331908017	0.07	-0.25309925	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	93
0.25202021	0.531189322	0.177660748	0.689192414	0.331908017	0.07	-0.24717655	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	94
0.25202021	0.531189322	359.3752747	0.689192414	0.331908017	0.07	-0.24886812	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	95
0.259515822	0.531189322	0.3555318099	0.689192414	0.331908017	0.07	-0.24290064	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	96
0.268422216	0.531189322	359.822876	0.689192414	0.331908017	0.07	-0.25581709	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	97
0.274242431	0.529610991	356.5976868	0.689192414	0.331908017	0.07	-0.25514173	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	98
0.282658809	0.527823925	352.3132629	0.689192414	0.331908017	0.07	-0.22646667	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	99
0.289481342	0.524240017	348.7414856	0.689192414	0.331908017	0.07	-0.18584333	0.280917436	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	100
0.295631796	0.522773385	345.8562427	0.689192414	0.331908017	0.07	-0.14959225	0.287409514	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025866791	101
0.30188179	0.521063983	345.8362427	0.689192414	0.331908017	0.07	-0.12285273	0.295676351	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025769586	102
0.310016006	0.516852021	342.4343567	0.689192414	0.331908017	0.07	-0.12247873	0.303581297	0.098978281	356.9781189	0.854839265	0.093458652	0.07	0.025689162	103
0.31650883	0.515292346	342.4343567	0.689192414	0.331908017	0.07	-0.08801497	0.309921771	0.098978281	0.26102768	0.854839265	0.093458652	0.07	0.025629793	104
0.322433472	0.511664748	342.4343567	0.689192414	0.331908017	0.07	-0.09007856	0.317409009	0.098978281	0.26102768	0.854839265	0.093458652	0.07	-0.00881052	105
0.328462511	0.509481847	338.1330566	0.689192414	0.331908017	0.07	-0.0456571	0.323925078	0.098978281	359.8240967	0.854839265	0.093458652	0.07	-0.00892244	106
0.337186784	0.50542444	338.1330566	0.689192414	0.331908017	0.07	-0.04452111	0.333218902	0.098978281	0.262622058	0.854839265	0.093458652	0.07	-0.00442278	107
0.343028843	0.503263414	338.1330566	0.689192414	0.331908017	0.07	-0.04575396	0.340139419	0.098978281	0.262622058	0.854839265	0.093458652	0.07	-0.0090986	108
0.349361139	0.499504656	338.1330566	0.689192414	0.331908017	0.07	-0.0467524	0.346836418	0.098978281	0.262622058	0.854839265	0.093458652	0.07	-0.00918395	109
0.357468933	0.497465044	335.2124939	0.689192414	0.331908017	0.07	-0.01675143	0.355978608	0.098978281	0.262622058	0.854839265	0.093458652	0.07	-0.00929763	110
0.363619357	0.493220657	335.2124939	0.689192414	0.331908017	0.07	-0.01817055	0.362875164	0.098978281	0.262622058	0.854839265	0.093458652	0.07	-0.0093882	111
0.368462511	0.509481847	338.1330566	0.689192414	0.331908017	0.07	-0.0456571	0.323925078	0.098978281	359.8240967	0.854839265	0.093458652	0.07	-0.00892244	112
0.377186784	0.50542444	338.1330566	0.689192414	0.331908017	0.07	-0.04452111	0.333218902	0.098978281	0.262622058	0.854839265	0.093458652	0.07	-0.00442278	113
0.383766751	0.488460001	335.2124939	0.689192414	0.331908017	0.07	-0.01944351	0.376269162	0.100532137	0.262622058	0.854839265	0.093458652	0.07	-0.01159935	114
0.387441726	0.484435707	335.2124939	0.689192414	0.331908017	0.07	-0.01945945	0.385834426	0.100532137	0.262622058	0.854839265	0.093458652	0.07	-0.01165673	115
0.389804304	0.482155267	335.2124939	0.689192414	0.331908017	0.07	-0.01890052	0.393116772	0.100532137	0.262622058	0.854839265	0.093458652	0.07	-0.01179787	116
0.395524979	0.478411525	331.0363159	0.689192414	0.331908017	0.07	-0.01949404	0.399813771	0.100532137	0.262622058	0.854839265	0.093458652	0.07	-0.01194053	117
0.401857495	0.474966019	331.0363159	0.689192414	0.331908017	0.07	0.026347523	0.406758189	0.100532137	0.262622058	0.854839265	0.093458652	0.07	-0.01212238	118
0.409557134	0.471620053	331.0363159	0.689192414	0.331908017	0.07	0.024574891	0.415852487	0.100532137	0.262622058	0.854839265	0.093458652	0.07	-0.01226588	119
0.415724576	0.468021572	331.0363159	0.689192414	0.331908017	0.07	0.025292228	0.423044398	0.101987563	0.262622058	0.854839265	0.093458652	0.07	-0.01241637	120
0.422156662	0.465756118	331.0363159	0.689192414	0.331908017	0.07	0.026199965	0.430816323	0.101987563	0.262622058	0.854839265	0.093458652	0.07	-0.01459853	121
0.430693954	0.462113947	331.0363159	0.689192414	0.331908017	0.07	0.022577553	0.440134078	0.101987563	0.421594888	0.854839265	0.093458652	0.07	-0.01488938	122
0.437208652	0.459065735	331.0363159	0.689192414	0.331908017	0.07	0.023338693	0.446764588	0.101987563	0.421594888	0.854839265	0.093458652	0.07	-0.01675074	123
0.443580031	0.457235843	334.015625	0.689192414	0.331908017	0.07	0.022896192	0.453485519	0.101987563	0.421594888	0.854839265	0.093458652	0.07	-0.01695107	124
0.449890733	0.45478341	334.015625	0.689192414	0.331908017	0.07	-0.01326649	0.46004948	0.101987563	0.421594888	0.854839265	0.093458652	0.07	-0.0172359	125
0.458058983	0.450551182	334.015625	0.689192414	0.331908017	0.07	0.024574891	0.469542921	0.101987563	359.7507019	0.854839265	0.093458652	0.07	-0.01745022	126
0.463901013	0.448880643	334.015625	0.689192414	0.331908017	0.07	-0.01243479	0.4765113	0.101987563	359.7507019	0.854839265	0.093458652	0.07	-0.01066821	127
0.46980378	0.445755631	334.015625	0.689192414	0.331908017	0.07	0.05222703	0.482942194	0.101987563	359.7507019	0.854839265	0.093458652	0.07	-0.01091268	128
0.47545892	0.443958819	334.015625	0.689192414	0.331908017	0.07	-0.0170252	0.48986271	0.101987563	359.7507019	0.854839265	0.093458652	0.07	-0.01122368	129
0.483075976	0.440833807	334.015625	0.689192414	0.331908017	0.07	-0.0163357	0.500611961	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.00912194	130
0.488488257	0.438706785	334.015625	0.689192414	0.331908017	0.07	-0.01958551	0.507175922	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.00938667	131
0.4938822873	0.436397642	334.015625	0.689192414	0.331908017	0.07	0.052707096	0.51421082	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.00968761	132
0.500684738	0.433051676	334.015625	0.689192414	0.331908017	0.07	-0.0207306	0.523909152	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.0095632	133
0.505094923	0.431150436	334.015625	0.689192414	0.331908017	0.07	0.04991653	0.531077087	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.01023094	134
0.509712517	0.427865207	334.015625	0.689192414	0.331908017	0.07	-0.02453473	0.538963377	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.01051512	135
0.51402247	0.426459312	334.015625	0.689192414	0.331908017	0.07	-0.02481561	0.545998275	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.01094943	136
0.520134032	0.42214939	334.015625	0.689192414	0.331908017	0.07	-0.045907407	0.556606127	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.01124454	137
0.524261832	0.42071863	330.6702881	0.689192414	0.331908017	0.07	-0.02458703	0.565221037	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.05996569	138
0.526959597	0.418781549	330.6702881	0.689192414	0.331908017	0.07	0.010760927	0.570008516	0.100542784	359.7507019	0.854839265	0.093458652	0.07	-0.05542971	139

0.311438888	0.415960073	141.6125641	0.148528919	0.544202328	0.048858368	0.006500258	0.638801038	0.227147222	178.5598907	0.141999319	0.254805237	0.07	-0.01801591	515
0.308622271	0.419823229	141.6125641	0.148528919	0.544202328	0.047601333	0.000889473	0.631295204	0.227147222	178.5598907	0.141999319	0.254805237	0.07	-0.01844534	516
0.303474665	0.421637028	141.6125641	0.148528919	0.544202328	0.046532564	0.004870585	0.623698995	0.227147222	178.5598907	0.141999319	0.254805237	0.07	-0.019194928	517
0.301126659	0.42494902	141.6125641	0.148528919	0.544202328	0.045422241	6.04154E-05	0.616778433	0.227147222	178.5598907	0.141999319	0.254805237	0.07	-0.01965083	518
0.296717167	0.428095847	141.6125641	0.148528919	0.544202328	0.044151045	-0.00064822	0.607279718	0.228623867	178.5598907	0.141999319	0.254805237	0.07	-0.01847503	519
0.291465133	0.430074781	141.6125641	0.148528919	0.544202328	0.043164963	0.003090719	0.600673139	0.228623867	178.5598907	0.141999319	0.254805237	0.07	-0.01895156	520
0.288930178	0.433429205	141.6125641	0.148528919	0.544202328	0.04195506	-0.00287753	0.593457341	0.228623867	178.5598907	0.141999319	0.254805237	0.07	-0.01947104	521
0.284846073	0.436429203	141.6125641	0.148528919	0.544202328	0.040755424	-0.00326006	0.58378553	0.230050012	178.5598907	0.141999319	0.254805237	0.07	-0.01813283	522
0.28065753	0.437878788	141.6125641	0.148528919	0.544202328	0.039960183	0.000105069	0.575667173	0.231457517	178.5598907	0.141999319	0.254805237	0.07	-0.01872457	523
0.278248817	0.441164047	141.6125641	0.148528919	0.544202328	0.038904688	-0.00559658	0.569888771	0.231457517	178.5598907	0.141999319	0.254805237	0.07	-0.01743635	524
0.273999631	0.442778766	141.6125641	0.148528919	0.544202328	0.038050751	-0.00074265	0.559653044	0.233088568	178.5598907	0.141999319	0.254805237	0.07	0.01303943	525
0.272142082	0.444762528	141.6125641	0.148528919	0.544202328	0.037017352	-0.00447437	0.553292938	0.233088568	178.5598907	0.141999319	0.254805237	0.07	0.014829559	526
0.266851217	0.447253793	141.6125641	0.148528919	0.544202328	0.036091907	-0.00444828	0.546072841	0.234677002	178.5598907	0.141999319	0.254805237	0.07	0.016474642	527
0.264993697	0.449232727	141.6125641	0.148528919	0.544202328	0.035064706	-0.00845703	0.54094559	0.234677002	178.5598907	0.141999319	0.254805237	0.07	0.01591236	528
0.262827784	0.452153772	141.6125641	0.148528919	0.544202328	0.034459831	-0.00945971	0.538687234	0.236265421	178.5598907	0.141999319	0.254805237	0.07	0.017946925	529
0.260766327	0.452153772	141.6125641	0.148528919	0.544202328	0.03386966	-0.01014218	0.529164016	0.236265421	178.5598907	0.141999319	0.254805237	0.07	0.017615079	530
0.257107109	0.454380333	141.6125641	0.148528919	0.544202328	0.033193453	-0.00802461	0.522823572	0.236265421	178.5598907	0.141999319	0.254805237	0.07	0.018971371	532
0.255596846	0.456116438	141.6125641	0.148528919	0.544202328	0.032350693	-0.01106875	0.517738938	0.237768725	178.5598907	0.141999319	0.254805237	0.07	0.018425074	533
0.253508657	0.457891484	141.6125641	0.148528919	0.544202328	0.031711293	-0.01087438	0.510323524	0.237768725	178.5598907	0.141999319	0.254805237	0.07	0.017949147	534
0.250223398	0.461016417	141.6125641	0.148528919	0.544202328	0.030656185	-0.00937907	0.499188483	0.237768725	178.5598907	0.141999319	0.254805237	0.07	0.017181329	535
0.245190375	0.464631885	141.6125641	0.148528919	0.544202328	0.029829369	-0.011232915	0.4879444335	0.2394104	178.5598907	0.141999319	0.254805237	0.07	0.018903997	536
0.240032539	0.469041377	141.6125641	0.148528919	0.544202328	0.02784605	-0.01622956	0.46861431	0.2394104	178.5598907	0.141999319	0.254805237	0.07	0.01827949	537
0.237089649	0.470881879	141.6125641	0.148528919	0.544202328	0.027376569	-0.01517728	0.462201983	0.240823224	178.5598907	0.141999319	0.254805237	0.07	0.017539663	538
0.235249117	0.472518474	141.6125641	0.148528919	0.544202328	0.026497752	-0.01912397	0.454677552	0.240823224	178.5598907	0.141999319	0.254805237	0.07	0.019576658	539
0.233209506	0.474497372	141.6125641	0.148528919	0.544202328	0.025888702	-0.01948663	0.447755662	0.240823224	178.5598907	0.141999319	0.254805237	0.07	0.018947977	540
0.231616661	0.477457002	141.6125641	0.148528919	0.544202328	0.025056022	-0.01682543	0.43991523	0.240823224	178.5598907	0.141999319	0.254805237	0.07	0.069981084	541
0.228212562	0.479520022	141.6125641	0.148528919	0.544202328	0.024655121	-0.01512532	0.431712985	0.240823224	178.5598907	0.141999319	0.254805237	0.07	-0.06710821	542
0.226586855	0.478052169	141.6125641	0.148528919	0.544202328	0.024148581	-0.01405717	0.424925476	0.240823224	178.5598907	0.141999319	0.254805237	0.07	0.066613203	543
0.224463373	0.479865968	141.6125641	0.148528919	0.544202328	0.02322248	-0.01974548	0.416121215	0.240823224	178.5598907	0.141999319	0.254805237	0.07	0.065152319	544
0.222727269	0.481546223	141.6125641	0.148528919	0.544202328	0.022659998	-0.01875928	0.408397198	0.24237442	178.5598907	0.141999319	0.254805237	0.07	-0.09337203	545
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629	-0.01875928	0.401447415	0.2437442	178.5598907	0.141999319	0.254805237	0.07	0.0631738	546
0.220765352	0.483403772	141.6125641	0.148528919	0.544202328	0.022080629									

0.432097405	0.560227275	355.6777344	0.540000024	0.040160874	-0.0634932	0.321338862	0.179749325	101.7507324	0.600000024	0.150000006	0.065390419	-0.6	1737
0.43791762	0.560227275	355.6777344	0.600000024	0.038855364	-0.02778876	0.324483842	0.188276932	92.91880035	0.600000024	0.150000006	0.064904547	-0.6	1738
0.442128003	0.560227275	355.6777344	0.600000024	0.038112589	-0.02921832	0.328842044	0.193662167	86.90071106	0.600000024	0.150000006	0.064085175	-0.6	1739
0.447496593	0.55882138	355.6777344	0.600000024	0.037137928	-0.03118135	0.33320564	0.197538853	81.58171082	0.600000024	0.150000006	0.063232552	-0.6	1740
0.451767683	0.55882138	355.6777344	0.600000024	0.03585411	-0.02840374	0.336081833	0.201053619	76.46500397	0.600000024	0.150000006	0.062722536	-0.6	1741
0.45634228	0.55882138	355.6777344	0.600000024	0.03486524	-0.0305019	0.340351453	0.206619859	65.28391266	0.600000024	0.150000006	0.060877432	-0.6	1742
0.458690256	0.557332933	355.6777344	0.600000024	0.033806603	-0.03288482	0.3456601315	0.210911557	56.89035034	0.600000024	0.150000006	0.059903503	-0.6	1743
0.463391444	0.557332933	355.6777344	0.600000024	0.033219391	-0.0275681	0.35120037	0.212805986	49.84739385	0.600000024	0.150000006	0.058536111	-0.6	1744
0.465491444	0.557332933	355.6777344	0.600000024	0.032130958	0.00152576	0.363396674	0.214828089	43.21515656	0.600000024	0.150000006	0.057242248	-0.6	1745
0.469740689	0.558227498	352.6232605	0.600000024	0.031227713	0.006124774	0.371261716	0.216775775	38.85628891	0.600000024	0.150000006	0.055600066	-0.59972986	1746
0.474949002	0.558227498	352.6232605	0.600000024	0.030617391	0.004700173	0.379126728	0.216775775	28.87600708	0.600000024	0.150000006	0.053840885	-0.58954419	1747
0.478299826	0.554322064	352.6232605	0.600000024	0.029411356	0.001709799	0.387420177	0.216775775	20.8773632	0.600000024	0.150000006	0.052369587	-0.58382596	1748
0.480443865	0.554322064	352.6232605	0.600000024	0.028592674	0.006962371	0.393417388	0.216775775	15.85702133	0.600000024	0.150000006	0.050658267	-0.55684993	1749
0.48454985	0.554322064	352.6232605	0.600000024	0.028095887	0.005713778	0.399994671	0.21507287	9.934567451	0.600000024	0.150000006	0.049075841	-0.53170396	1750
0.486511737	0.554322064	352.6232605	0.600000024	0.027144865	0.003195025	0.406759888	0.213409975	5.190643787	0.600000024	0.150000006	0.047454846	-0.5082065	1751
0.490083516	0.554322064	352.6232605	0.600000024	0.026690633	0.001928151	0.412018418	0.209075689	359.2505493	0.600000024	0.150000006	0.046341725	-0.46301285	1752
0.494167626	0.552850604	352.6232605	0.600000024	0.025868987	-0.00049255	0.416879535	0.204603031	351.9811707	0.600000024	0.150000006	0.044587188	-0.40151042	1753
0.495986313	0.552850604	352.6232605	0.600000024	0.024875903	0.00474956	0.420761496	0.202820346	347.1845703	0.600000024	0.150000006	0.043600528	-0.36004509	1754
0.499293447	0.552850604	352.6232605	0.600000024	0.02445439	0.003494556	0.423158765	0.198698923	343.8883362	0.600000024	0.150000006	0.042798966	-0.3148752	1755
0.502848208	0.552850604	352.6232605	0.600000024	0.023688738	0.001098051	0.429153353	0.193702102	340.4443359	0.600000024	0.150000006	0.041147761	-0.26397291	1756
0.504380345	0.552850604	352.6232605	0.600000024	0.022866206	-0.00165683	0.432088137	0.19037886	336.467041	0.600000024	0.150000006	0.040296372	-0.20869826	1757
0.507527232	0.552850604	352.6232605	0.600000024	0.022511843	-0.00290628	0.435353021	0.186941221	339.3587896	0.600000024	0.150000006	0.03978785	-0.15480754	1758
0.508976817	0.551296592	352.6232605	0.600000024	0.02184331	-0.00559994	0.437345684	0.184687539	330.4221497	0.600000024	0.150000006	0.038806124	-0.11333022	1759
0.510994554	0.551296592	352.6232605	0.600000024	0.021401687	0.003163943	0.439186901	0.180018112	327.2168884	0.600000024	0.150000006	0.038171185	-0.04147827	1760
0.510994554	0.551296592	352.6232605	0.600000024	0.020934547	0.001502056	0.442627162	0.176569819	323.6822205	0.600000024	0.150000006	0.037240009	0.023925077	1761
0.510994554	0.551296592	352.6232605	0.600000024	0.020934547	0.001502056	0.44404	0.174201801	323.6822205	0.600000024	0.150000006	0.036826222	0.04696233	1762
0.510994554	0.551296592	352.6232605	0.600000024	0.020934547	0.001502056	0.44700405	0.170820042	319.4638367	0.600000024	0.150000006	0.036028089	0.124495316	1763
0.512427151	0.551296592	352.6232605	0.600000024	0.020602977	0.000276363	0.450476289	0.167510107	319.4638367	0.600000024	0.150000006	0.035127286	0.159703916	1764
0.512427151	0.551296592	352.6232605	0.600000024	0.020602977	0.000276363	0.452269584	0.166102603	319.4638367	0.600000024	0.150000006	0.034674602	0.175212596	1765
0.512427151	0.551296592	352.6232605	0.600000024	0.020602977	0.000276363	0.45250012	0.164599299	319.4638367	0.600000024	0.150000006	0.033714208	0.189659169	1766
0.513854921	0.551296592	352.6232605	0.600000024	0.020272613	-0.00098509	0.460525781	0.16100736	319.4638367	0.600000024	0.150000006	0.033172016	0.20975991	1767
0.513854921	0.551296592	352.6232605	0.600000024	0.020272613	-0.00098509	0.46265167	0.16100736	322.4573364	0.600000024	0.150000006	0.032150702	0.202932387	1768
0.513854921	0.551296592	352.6232605	0.600000024	0.020272613	-0.00098509	0.465762019	0.15911822	322.4573364	0.600000024	0.150000006	0.031394377	0.228082377	1769
0.513854921	0.551296592	352.6232605	0.600000024	0.020272613	-0.00098509	0.470146865	0.157585695	326.0150146	0.600000024	0.150000006	0.030350726	0.213384096	1770
0.513854921	0.551296592	352.6232605	0.600000024	0.019927192	-0.00234915	0.474332184	0.157585695	326.0150146	0.600000024	0.150000006	0.029375869	0.209247391	1771
0.513854921	0.551296592	352.6232605	0.600000024	0.019927192	-0.00234915	0.476367623	0.155853555	329.5896912	0.600000024	0.150000006	0.028879876	0.202122301	1772
0.516819656	0.551296592	352.6232605	0.600000024	0.019927192	-0.00234915	0.481356442	0.155853555	332.8493347	0.600000024	0.150000006	0.027717175	0.165250088	1773
0.516819656	0.551296592	352.6232605	0.600000024	0.019586921	0.00374033	0.48542732	0.154363558	332.8493347	0.600000024	0.150000006	0.027212107	0.191702106	1774
0.516819656	0.551296592	352.6232605	0.600000024	0.019586921	0.00374033	0.487885803	0.154363558	336.9098511	0.600000024	0.150000006	0.026179791	0.147603068	1775
0.516819656	0.551296592	352.6232605	0.600000024	0.019586921	0.00374033	0.492632508	0.154363558	336.9098511	0.600000024	0.150000006	0.025073101	0.143695922	1776
0.518247366	0.551296592	352.6232605	0.600000024	0.019586921	-0.00374033	0.494130492	0.154363558	340.1887512	0.600000024	0.150000006	0.024723864	0.108820055	1777
0.518247366	0.551296592	352.6232605	0.600000024	0.019256871	0.00513711	0.499281615	0.154363558	340.1887512	0.600000024	0.150000006	0.023935486	0.105696393	1778
0.518247366	0.551296592	352.6232605	0.600000024	0.019256871	0.00513711	0.500779569	0.154363558	340.1887512	0.600000024	0.150000006	0.023173817	0.102478933	1779
0.519779563	0.551296592	352.6232605	0.600000024	0.019256871	-0.00513711	0.502838969	0.154363558	340.1887512	0.600000024	0.150000006	0.022693764	0.100341508	1780
0.519779563	0.549825191	352.6232605	0.600000024	0.018902787	-0.00669033	0.508080542	0.154363558	343.8286133	0.600000024	0.150000006	0.022109548	0.059836509	1781
0.521190286	0.549825191	352.6232605	0.600000024	0.018857977	0.004126913	0.508080542	0.154363558	347.121582	0.600000024	0.150000006	0.021472032	0.022266313	1782
0.521190286	0.549825191	352.6232605	0.600000024	0.018531293	0.00283129	0.509692907	0.154363558	347.121582	0.600000024	0.150000006	0.021096244	0.020305407	1783

XBEE ZIGBEE NETWORKS

Introduction to ZigBee

ZigBee is an open global standard built on the IEEE 802.15.4 MAC/PHY. ZigBee defines a network layer above the 802.15.4 layers to support advanced mesh routing capabilities. The ZigBee specification is developed by a growing consortium of companies that make up the ZigBee Alliance. The Alliance is made up of over 300 members, including semiconductor, module, stack, and software developers.

ZigBee Stack Layers

The ZigBee stack consists of several layers including the PHY, MAC, Network, Application Support Sublayer (APS), and ZigBee Device Objects (ZDO) layers. Technically, an Application Framework (AF) layer also exists, but will be grouped with the APS layer in remaining discussions. The ZigBee layers are shown in the figure below.

A description of each layer appears in the following table:

TECHNOLOGY OVERVIEW

- **ZigBee: A High Level Communication Protocol**
 - Uses small, ultra low-power digital radios based on the IEEE 802.15.4 standard for wireless networks
- **Targeted at RF Applications Requiring:**
 - Low data rate
 - Long battery life
 - Secure networking
- **IEEE 802.15.4**
 - Serves as the standard upon which ZigBee is built

ZigBee Technology Facts	
Attribute	ZigBee
Number of Channels	27
Radio Frequency Band[s]	<ul style="list-style-type: none">● 2.4 GHz with 16 channels for global use● 915 MHz with 10 channels for N. America, Australia and a few additional countries● 868 MHz with 1 channel for EU countries
Network Capabilities	Self-organizing and self-healing dynamic mesh network based on ZigBee public standard
Network Size	Thousands of devices per network

ZIGBEE & OTHER WIRELESS STANDARDS

Market Name	ZigBee®	---	WiFi-Fi™	Bluetooth™
Standard	802.15.4	GSM/GPRS CDMA1xRTT	802.11b	802.15.1
Application Focus	Monitoring & Control	Wide Area Voice & Data	Web, Email, Video	Cable Replacement
System Resources	4KB - 32KB	16MB+	1MB+	250KB+
Battery Life (days)	100 - 1,000+	1-7	.5 - 5	1 - 7
Network Size	Unlimited (2 ⁶⁴)	1	32	7
Bandwidth (KB/s)	20 - 250	64 - 128+	11,000+	720
Transmission Range (meters)	1 - 100+	1,000+	1 - 100	1 - 10+
Success Metrics	Reliability, Power, Cost	Reach, Quality	Speed, Flexibility	Cost, Convenience

Device Types

ZigBee defines three different device types: coordinator, router, and end device.

Node Types / Sample of a Basic ZigBee Network Topology

A **coordinator** has the following characteristics: it

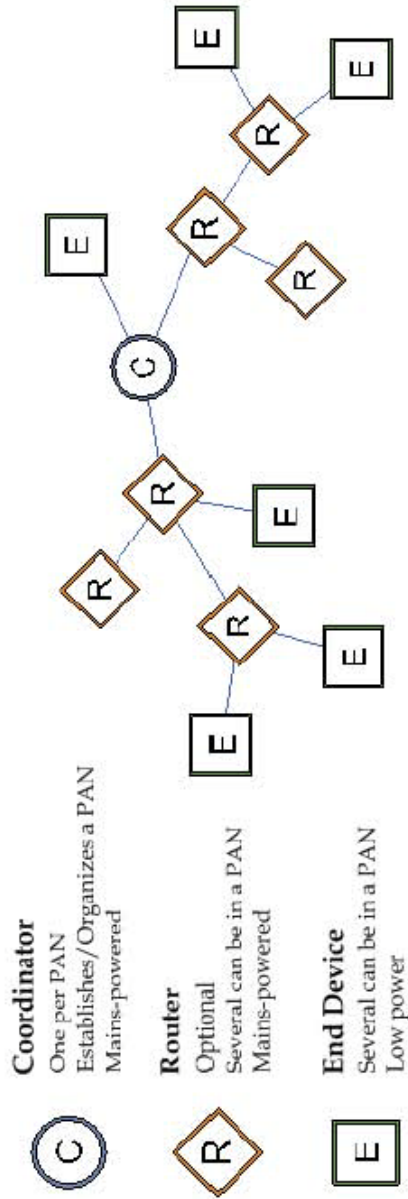
- Selects a channel and PAN ID (both 64-bit and 16-bit) to start the network
- Can allow routers and end devices to join the network
- Can assist in routing data
- Cannot sleep--should be mains powered
- Can buffer RF data packets for sleeping end device children.

A **router** has the following characteristics: it

- Must join a ZigBee PAN before it can transmit, receive, or route data
- After joining, can allow routers and end devices to join the network
- After joining, can assist in routing data
- Cannot sleep--should be mains powered.
- Can buffer RF data packets for sleeping end device children.

An **end device** has the following characteristics: it

- Must join a ZigBee PAN before it can transmit or receive data
- Cannot allow devices to join the network
- Must always transmit and receive RF data through its parent. Cannot route data.
- Can enter low power modes to conserve power and can be battery-powered.



In ZigBee networks, the coordinator must select a PAN ID (64-bit and 16-bit) and channel to start a network. After that, it behaves essentially like a router. The coordinator and routers can allow other devices to join the network and can route data.

After an end device joins a router or coordinator, it must be able to transmit or receive RF data through that router or coordinator. The router or coordinator that allowed an end device to join becomes the "parent" of the end device. Since the end device can sleep, the parent must be able to buffer or retain incoming data packets destined for the end device until the end device is able to wake and receive the data.

A Comparison of Transparent and API Operation

The following table compares the advantages of transparent and API modes of operation:

Transparent Operation Features	
Simple Interface	All received serial data is transmitted unless the module is in command mode.
Easy to support	It is easier for an application to support transparent operation and command mode
API Operation Features	
Easy to manage data transmissions to multiple destinations	Transmitting RF data to multiple remotes only requires changing the address in the API frame. This process is much faster than in transparent operation where the application must enter AT command mode, change the address, exit command mode, and then transmit data. Each API transmission can return a transmit status frame indicating the success or reason for failure.
Received data frames indicate the sender's address	All received RF data API frames indicate the source address.
Advanced ZigBee addressing support	API transmit and receive frames can expose ZigBee addressing fields including source and destination endpoints, cluster ID and profile ID. This makes it easy to support ZDO commands and public profile traffic.
Advanced networking diagnostics	API frames can provide indication of IO samples from remote devices, and node identification messages.
Remote Configuration	Set / read configuration commands can be sent to remote devices to configure them as needed using the API.

Ejemplo de transmit request.

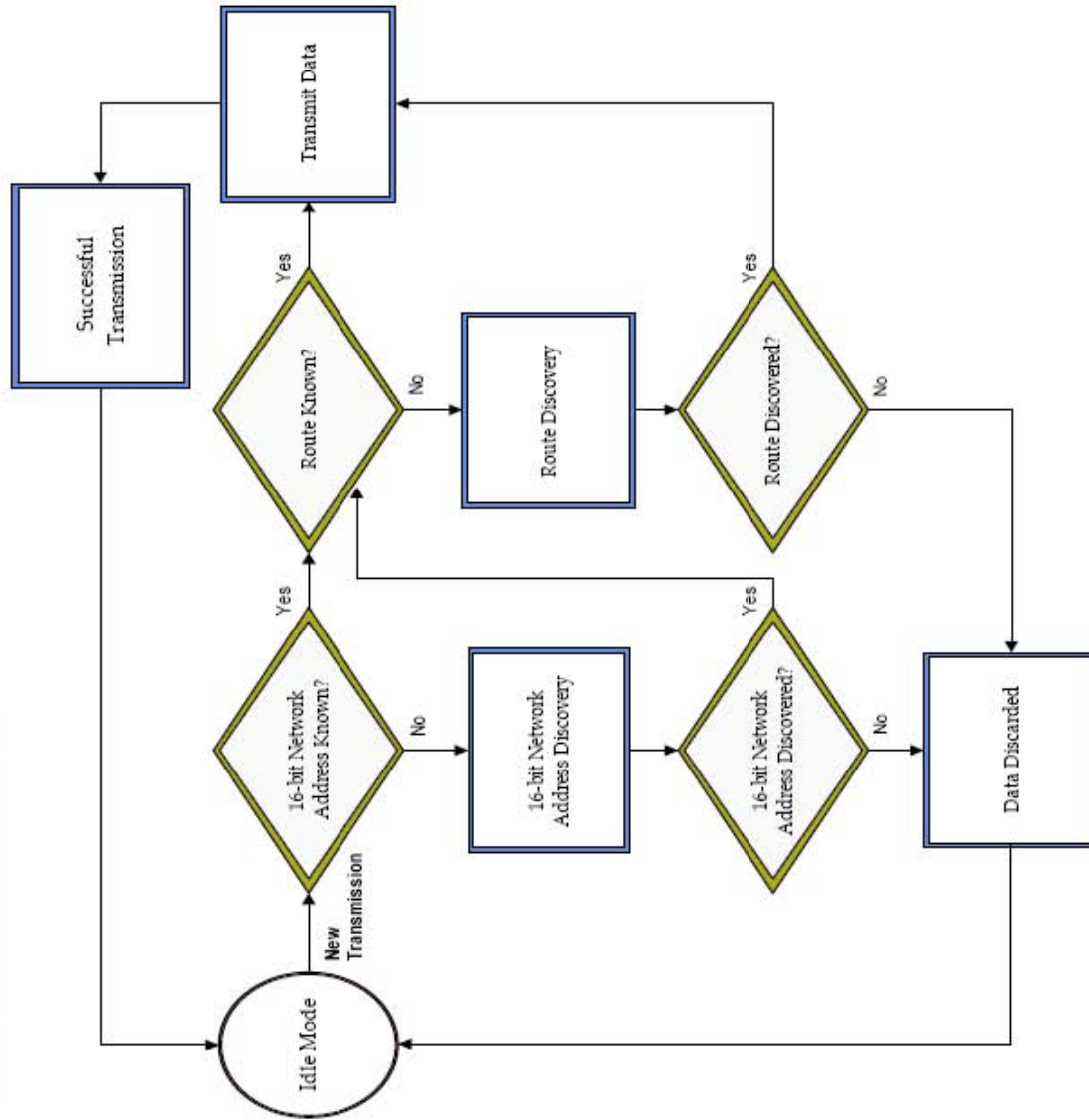
Frame Fields	Offset	Example	Description	
Start Delimiter	0	0x7E		
Length	MSB 1	0x00	Number of bytes between the length and the checksum	
	LSB 2	0x16		
Frame-specific Data	3	0x10		
Frame Type				
Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.	
64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address	
	6	0x13		
	7	0xA2		
	8	0x00		
	9	0x40		
	10	0x0A		
	11	0x01		
	LSB 12	0x27		
	MSB 13	0xFF		Set to the 16-bit address of the destination device, if known. Set to 0xFFE if the address is unknown, or if sending a broadcast.
	LSB 14	0xFE		
16-bit Destination Network Address				

Leer capítulo 9 de "Xbee_API.pdf" para mayor información.

A P I P a c k e t	Broadcast Radius	15	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.	
		Options			<p>Bitfield of supported transmission options. Supported values include the following:</p> <ul style="list-style-type: none"> 0x20 - Enable APS encryption (if EE=1) 0x40 - Use the extended transmission timeout for this destination <p>Enabling APS encryption decreases the maximum number of RF payload bytes by 4 (below the value reported by NP).</p> <p>Setting the extended timeout bit causes the stack to set the extended transmission timeout for the destination address. (See chapter 4.)</p> <p>All unused and unsupported bits must be set to 0.</p>
	RF Data	17	0x54		
		18	0x78		
		19	0x44		
		20	0x61		
		21	0x74		
		22	0x61		
		23	0x30		
		24	0x41		
25	0x13		0xFF - the 8 bit sum of bytes from offset 3 to this byte.		
Checksum					

Leer capítulo 9 de "Xbee_API.pdf" para mayor información.

Transmit Mode Sequence



Apéndice G – ReacTIVision

3.2. Optical marker tracking systems

The present work stemmed from our experience with d-touch, a system developed specifically for tangible user interfaces [3]. D-touch makes use of a topological recognition approach which we have not encountered elsewhere. Below we review some other systems and approaches.

Within our field we observe that the ARToolKit [6] is the most commonly employed optical marker tracking system. ARToolKit markers can be arbitrary images framed in a black square. Markers are recognised with a simplified template matching algorithm. The use of arbitrary images offers a significant benefit to many applications as the markers can be made readable by users. In addition to marker tracking ARToolKit provides tools for overlaying aligned computer graphics on a real-time video stream. We note that the success of ARToolKit is not necessarily due to the quality or speed of its marker tracking subsystem. Recent projects such as ARTag [14] have specifically aimed to address shortcomings in ARToolKit’s marker tracking.

Zhang et al. [15] evaluated the strengths and weaknesses of four publicly available square-shaped marker tracking systems including ARToolKit. They considered criteria of usability, efficiency, accuracy, and reliability. They also provide an insightful discussion of the diversity of requirements placed on such systems in different application areas.

A range of other marker recognition techniques have been explored in the literature, including: projective invariant line pencils [16], circular bar codes [17], Discrete Cosine Transform basis functions [18], and non-symmetric patterns of equally spaced dots [19].

4. Topology Based Fiducials

This section introduces the representations and data structures used by reacTIVision to recognise fiducials in binary images. First the topological region adjacency graph is introduced. Following subsections discuss the application of this representation in d-touch and reacTIVision. We then describe our approach to determining the location and orientation of fiducials. Next we outline the method used to generate the topology and geometry of individual reacTIVision fiducials. A final subsection introduces left heavy depth sequences, a method for computing canonical names for unordered trees which we use to recognise subgraphs corresponding to fiducials.

4.1. Topological Recognition

ReacTIVision employs the topological fiducial recognition approach introduced by Costanza and Robinson in the d-touch system [3]. In this approach, a region adjacency graph is derived from a binary image of the scene through the process of *segmentation*. The graph can be understood as

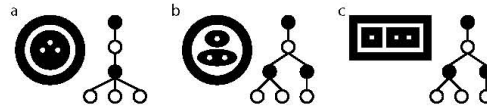


Figure 3: Some simple topologies and their corresponding region adjacency graphs.

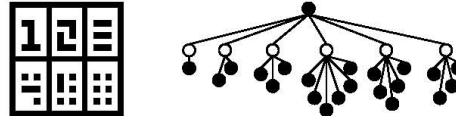


Figure 4: A d-touch fiducial and its region adjacency graph.

a tree representing the containment hierarchy of the image, that is, which black regions are contained inside which white regions and vice-versa. Regions of the image containing no other regions (unbroken blobs for example) appear as leaf nodes in the region adjacency graph. Figure 3 illustrates some simple images and their corresponding region adjacency graphs. Observe that figures 3b and 3c have identical region adjacency representations even though their geometries are different.

One important property of region adjacency graphs derived from binary images is that they belong to a class of graphs known as rooted trees [20]. Since the children of any node in the graph are not ordered the graphs are more specifically known as unordered rooted trees.

4.2. D-touch Fiducials

D-touch employs a single topology for all fiducials in a set. Figure 4 shows a d-touch fiducial and its region adjacency graph. The fiducial belongs to a set with a black grid containing six white regions each of which contains a different number of black regions from one to six. The set contains 120 unique fiducials which are differentiated using a permutational code expressed by the number of black leaf regions contained within the spatially ordered set of white regions. For example: reading clockwise, the code in Figure 4 is (1,2,3,6,5,4).

We identified a number of aspects of the d-touch approach which offered opportunities for improvement in our context: Firstly, d-touch binds geometric extraction of the permutational code to specific fiducial sets. Secondly, the original d-touch publication does not prescribe a specific method for computing location and orientation, leaving some steps to traditional computer vision techniques. Finally, the simple geometries of d-touch fiducials (which incidentally have the advantage of being easily drawable by hand) were not designed to minimise fiducial size.

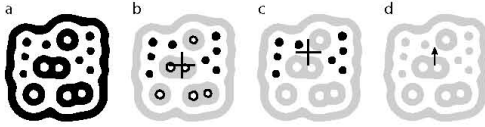


Figure 5: (a) a reactIVision fiducial (b) black and white leaves and their average centroid (c) black leaves and their average centroid, and (d) the vector used to compute the orientation of the fiducial.

4.3. ReactIVision Fiducials

Following our evaluation and re-implementation of a d-touch fiducial recogniser, we set out to produce smaller fiducials, and eventually arrived at a scalable technique which allows us to vary the size of the fiducials depending on the number of unique fiducials required. We also aimed to explore whether we could accurately compute the location and orientation of fiducials without resorting to additional image processing techniques such as corner or edge detection. Given that the region adjacency graph already existed and was relatively expensive to compute, it seemed wise to make maximum use of it.

ReactIVision fiducials are identified purely by their topological structure. Each fiducial in a set has a unique topology. In addition to expressing the correct topology, the geometry of each fiducial is constrained by the method used to compute its location and orientation. In the next subsection we describe the method used to compute the location and orientation of fiducials; after that we discuss the generation of trees describing the topology of each fiducial; following which we give an overview of how we generate geometries which conform to the requirements of our location and orientation calculation method while minimising size. Note that the separation of this discussion is somewhat artificial as the methods were co-designed, with the requirements of each effecting the design of the others to some extent.

4.4. Fiducial Location and Orientation

Our method for computing fiducial location and orientation was influenced significantly by the design of our segmentation algorithm, which only retains axis aligned bounding boxes for each region. The center of a region’s axis aligned bounding box provides a good approximation for the center of the region if the region is square, circular, and/or relatively small. We reasoned that as leaf regions will always be the smallest regions in a rendering of a tree, their centers are likely to be the most accurate spatial information we have about a fiducial. Consequently we choose to compute a fiducial’s location and orientation as a combination of the bounding box centers of its leaf nodes.

As illustrated in figure 5, we compute the center point of

the fiducial by taking a weighted average of all leaf centers. The vector from this centroid to a point given by the weighted average of all black (or white) leaf centers is used to compute the orientation of the fiducial. Each leaf is weighted by a function of its depth in the tree to account for the total area it consumes.

We selected this method because it can be applied to any fiducial with at least one black and one white leaf region. Thus allowing us to vary the topological structure of fiducials without changing the method used to track them. In practice we found that a minimum of four black and four white leaf nodes was necessary to achieve the tracking stability which we previously achieved with our d-touch fiducial tracker.

4.5. Fiducial Tree Generation

Before generating the geometry for a set of fiducials we generate a set of unique trees. Given a set size it is possible to calculate the number of tree nodes required to accommodate the set, however other constraints are also important such as ensuring a certain number of black and white leaf nodes. The number of nodes in a tree and its depth also impacts the minimum size of the geometry which can be generated for a tree. Additionally, smaller trees are less unique, leading to a greater potential for encountering false detections when trying to recognise them in a scene.

Rather than enumerating all possible trees (which is possible, see [21]), we randomly generate trees with the desired number of nodes and select those which fulfill criteria including maximum depth and number of black and white leaves. The generation process slows when many of the candidates have been found in the search space. By observing this slowing we experimentally reduce the number of nodes to the minimum required to generate a set fulfilling our criteria.

4.6. Fiducial Geometry Generation

Given a tree representing a fiducial’s topology we create a compact geometry which conforms to the constraints implied by the location and orientation method described above: that the computed centroid of the fiducial is the same as, or lies very close to the real center of the fiducial and that the centroid of all black leaves is sufficiently distant from the centroid of all leaves to allow the fiducial’s orientation to be computed with reasonable accuracy.

Each tree can be drawn in a huge number of ways making an exhaustive search for ‘optimally’ rendered fiducials impractical. We chose to employ a genetic algorithm to optimise parameters such as fiducial area, aspect ratio, symmetry and centroid locations for black and white leaves. We experimented with many combinations of these parameters before achieving satisfactory results.

The method used to draw a single fiducial involves aggregating circular leaves at angles relative to a fixed starting point while maintaining a fixed spacing between regions. A list of these angles forms the genotype for the genetic algorithm. Circles are easy to pack together at arbitrary angles, which we consider important to optimise the location of the centroids used to compute location and orientation.

We implemented a simple but effective distributed computing system which allows us to compute a number of different versions of a fiducial set within a reasonable period of time. When producing the currently used set of 128 fiducials with 19 tree nodes with a maximum depth of 3, we obtained usable results after 12 hours of computation time on a cluster of 11 dual processor 1Ghz Pentium 3 machines. In this time we computed 20 pools of 500 fiducials for each of 128 trees and selected the best result for each tree according to size and orientation vector length.

4.7. Left Heavy Depth Sequences

In order to recognise subgraphs representing fiducials in the region adjacency graph of the whole scene it is useful to be able to compute a canonical name for any subgraph. This name can then be compared against a dictionary of recognisable fiducial names². One canonical form for unordered rooted trees is the left heavy depth sequence discussed by Nakano and Uno [21] and Asai et al. [22] which we describe below.

The *depth sequence* of a graph is defined as the sequence of node depth values encountered in a preorder (depth first) traversal, where the root has depth 0 and the depth of any other node is the number of edges between it and the root. Depth sequences can be used to uniquely describe ordered trees, however some unordered trees can be described by multiple depth sequences, depending on the order in which children are visited. For example, the depth sequences (0,1,2,3,3,2,3) and (0,1,2,3,2,3,3) are both possible for the tree in figure 3c. To resolve this ambiguity we can order traversal according to the ‘weight’ of the depth sequence of each child, where heavier sequences are those which compare as lexically greater. Following from this, the *left heavy depth sequence* can be understood recursively as the concatenation of the left heavy depth sequences of the children of a node in lexically decreasing order, where the depth sequence of a leaf is simply its depth.

Figure 6 shows a fiducial and its corresponding region adjacency graph annotated with the left heavy depth sequence of each node. As in the reacTIVision system, the outermost black region (the top node in the graph) serves only to provide a boundary for the white region it contains.

²In practice we first identify candidate subgraphs using simple metrics such as their depth and number of descendants. Only then is the canonical name of the subgraph considered.

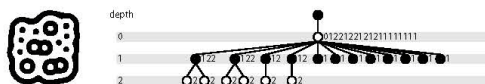


Figure 6: A fiducial and its left heavy depth sequence.

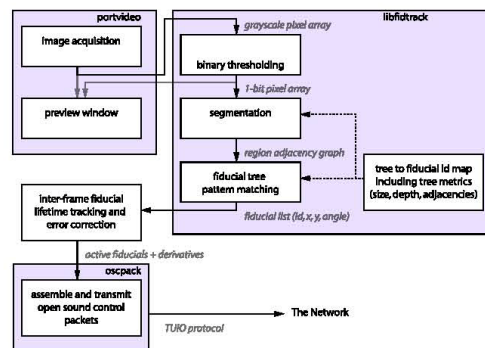


Figure 7: reacTIVision system diagram

5. Implementation

Figure 7 shows the module level design of the reacTIVision system, which aggregates three open source libraries: (1) PortVideo, a cross-platform real-time video acquisition and previewing library; (2) oспенpack, a library for packing and unpacking Open Sound Control messages; and (3) libfidtrack, a library implementing the fiducial recognition algorithms described in this paper. The system communicates with clients using the TUJO protocol [4]. Glue code connects the libraries and handles fiducial object lifetime tracking, error correction and formatting of TUJO messages.

Figure 7 also illustrates the flow of data between modules: PortVideo acquires grayscale video frames from the operating system and passes them to libfidtrack. A thresholding algorithm produces a binary image which is fed to the segmentation module to construct a region adjacency graph. This graph is passed to the fiducial recognition module which recognises subgraphs whose topology matches those in a dictionary loaded from a fiducial tree description file. The recognised fiducial identifiers along with their location and orientation are then passed to a module which tracks fiducials as they become visible and disappear. This module also calculates second order information such as velocity and acceleration vectors. Finally this information is formatted into Open Sound Control packets conforming to the TUJO protocol and passed to clients via a network socket.

5.1. Thresholding

As noted by Pintaric [23], thresholding (binarisation) is an important and sometimes poorly handled step in optical marker tracking systems. Some earlier systems simply used a global threshold, while more recently, locally adaptive thresholding has been employed [23, 3]. Although the lighting conditions in the reacTable* are not as unpredictable as in some augmented reality applications, we found an adaptive scheme to be useful.

Unlike many applications where quality is of greatest importance, we were equally concerned with speed. One property of our system which allowed us to select an efficient algorithm was that correct thresholding is only required on image areas containing clean, well lit fiducial images within a known scale range. The segmentation process tends to take time proportional to the number of black and white regions in the image, so ideally our thresholder should output non-fiducial areas in a single colour. Having experimented with a variety of schemes based on locally computed statistics we currently employ a tile-based variation of Berensen's method [24]. This method computes the threshold for each tile (e.g. 32×32 pixels) as the mean of the minimum and maximum intensities in a larger surrounding tile. Areas of low dynamic range are clamped to black.

5.2. Segmentation

Segmentation constructs a region adjacency graph from the binary image produced by thresholding. This section presents some aspects of the technique used to build the graph. A key feature of our approach is that in general we construct an *incomplete* region adjacency graph.

The main data structure is the Region which represents a node in the graph. An axis aligned bounding box for each Region is stored for later use by the fiducial recogniser. During segmentation, the auxiliary RegionReference structure provides an additional level of indirection to support efficient region merging. Large pre-allocated arrays of Regions and RegionReferences are reused for each frame. Individual structures are allocated by taking the next free block from the end of the appropriate array. This mechanism is efficient and allows Regions to be enumerated by stepping through the array.

```
struct Region{
    int flags;
    short left, top, right, bottom;
    ...
    int adjacent_regions_count;
    Region *adjacent_regions[ MAX_ADJACENT ];
};

struct RegionReference{
    RegionReference *redirect;
    Region *region;
};
```

During segmentation, arrays of pointers to RegionReferences corresponding to pixels in the current and previous

scan line are maintained. The algorithm proceeds through the source image line-by-line, extending existing Regions from the previous line and allocating new Regions when necessary. Adjacent Regions are linked together as discussed below. Some inputs require Regions to be merged. During merging the redirect pointer of one of the corresponding RegionReferences is set to point to the other RegionReference. This allows pointers to merged RegionReferences to be resolved lazily rather than requiring all pointers to be updated after every merge operation.

Bidirectional adjacency relations between Regions are represented using fixed size arrays of pointers to facilitate the allocation scheme mentioned earlier. This places an upper bound on the number of adjacency relations which can be maintained for a Region. When an adjacency array becomes full, new adjacencies are discarded and the participating Regions are marked as *saturated* (for the region whose array is full) or *fragmented* (for the other region) using the flags field. As a result of this mechanism, the segmentation algorithm may construct incomplete region adjacency graphs, however, note that subgraphs containing no saturated or fragmented nodes are complete. These complete subtrees are sufficient to perform fiducial recognition provided that the capacity of each region adjacency array is sufficient to represent the region adjacency graphs of the target fiducials.

5.3. Fiducial Recognition

Fiducial recognition proceeds in two steps: First, candidate subgraphs are identified in the incomplete region adjacency graph produced by the segmenter. Candidates are selected according to properties precomputed from the dictionary of recognisable fiducials, including total node count and maximum child depth. Second, the left heavy depth sequence of each candidate is computed according to the definition in section 4.7 and used as a key against a dictionary of depth sequences. If the sequence is found in the dictionary a fiducial has been identified and we compute its location and orientation according to the method described in section 4.4.

The incomplete region adjacency graph produced by the segmenter precludes top down traversal. As the graph is undirected, the parent of each node must be deduced during traversal. We employ a progressive bottom up traversal in which we iterate through each leaf attempting to traverse upwards. A counter in each parent is used to ensure that we only progress upwards once all of the children of the parent have been visited. Upwards traversal is terminated if a fragmented or saturated node is encountered, or if a criterion for candidate selection such as descendant count is exceeded.

Observe that in the worst case each node will be visited three times: once during the iteration to find leaf nodes, once during the upwards traversal to identify candidate sub-

Library	CPU Usage	Frame Rate
libfidtrack	18%	30 fps
libdtouch	82%	30 fps
dtouch_old	100%	10 fps

Table 1: CPU utilisation and frame rate of d-touch and reacTIVision fiducial trackers tracking 12 fiducials in a 640×480 frame.

trees, and (assuming each node can be in at most one candidate subtree) once during the recursive calculation of the left heavy depth sequence. Ignoring constant factors, and assuming that depth sequence dictionary look-up is a constant time operation, the time complexity for recognising all fiducials in a scene is linear with respect to the number of nodes in the region adjacency graph. This is the same time complexity offered by Costanza and Robinson’s method [11]. While their method is constrained to three level trees, our method works with trees of any depth, allowing reacTIVision to recognise fiducials generated from a larger class of trees.

6. Results

This section summarises improvements achieved by reacTIVision over the d-touch topological fiducial tracker in three areas: performance, fiducial size, and scalability to different fiducial topologies and set sizes.

Table 1 compares the CPU utilisation and frame rates³ obtained with reacTIVision (libfidtrack) and two versions of d-touch: a recent version available online (libdtouch) [25] and the original version obtained from the author (dtouch_old). The table indicates that the current reacTIVision implementation is over 4 times faster than the current d-touch system, and over 16 times faster than the version in use when development of reacTIVision began. These gains arise from the techniques described in the implementation section and are equally applicable to tracking d-touch fiducials. In fact, reacTIVision can track the type of d-touch fiducials shown in figure 4 with similar performance to its native fiducials. We note that these results are not comprehensive as other factors such as noise or non-fiducial inputs may also effect performance in the field. We also note that d-touch supports features not provided by reacTIVision, such as reporting the coordinates of the corners of rectangular fiducials.

The d-touch fiducial shown in figure 4 is representative of the set containing 120 unique fiducials. Decomposing the fiducial on a grid of pixel-like squares, it has a size of 21×21 units (including the required outer white boundary).

³These results were obtained while tracking 12 fiducials filling a 640×480 frame. The system used was a 2.6 GHz Pentium IV with 512 Mb RAM running Fedora Linux 3, Kernel 2.6.11 with a fire-i camera in grayscale mode.

Taking the diameter of the smallest circles on a reacTIVision fiducial (such as the one in figure 5) as an equivalent unit, the maximum size of the reacTIVision fiducials generated for a unique set of 128 is 15×15, an almost 50% reduction in area.

The generalised methods for recognising and tracking fiducials described earlier allow the reacTIVision system to support fiducial sets of varying sizes and topologies without any code changes. We consider this a significant improvement over d-touch in our context, where we wish to experiment with fewer smaller fiducials (at one extreme) and a larger number of uniquely identifiable fiducials (at the other).

7. Future Work

The current system partitions leaf nodes in the region adjacency graph into two classes based on their colour. The left heavy depth sequence provides a partial ordering of nodes, which could be used to construct alternate partitionings of the graph. Alternate dual partitionings might create opportunities for more efficient spatial packing. Partitioning into more than two classes could facilitate an implementation of three dimensional tracking using a generalised method similar to the one we currently use to compute location and orientation.

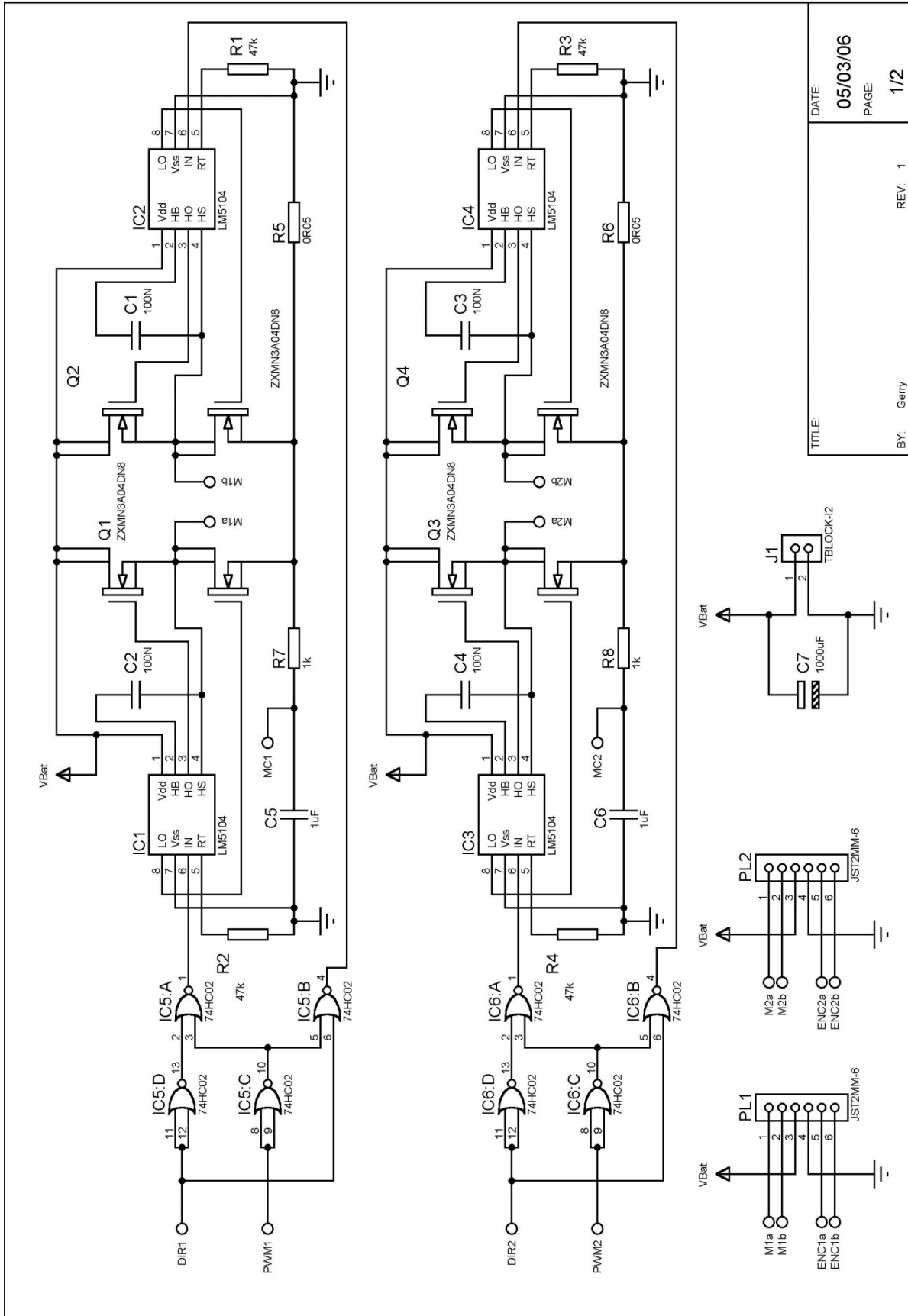
Some recent marker tracking systems are able to recognise markers which are partially occluded, by a finger for example [16, 14]. ReacTIVision currently only recognises fiducials if their topology has not been modified by occlusion or thresholding errors. By using fiducials with sufficiently complex topology it may be possible to apply the method described by Lladós et al. [26] to implement occlusion or error tolerant tracking.

8 Conclusion

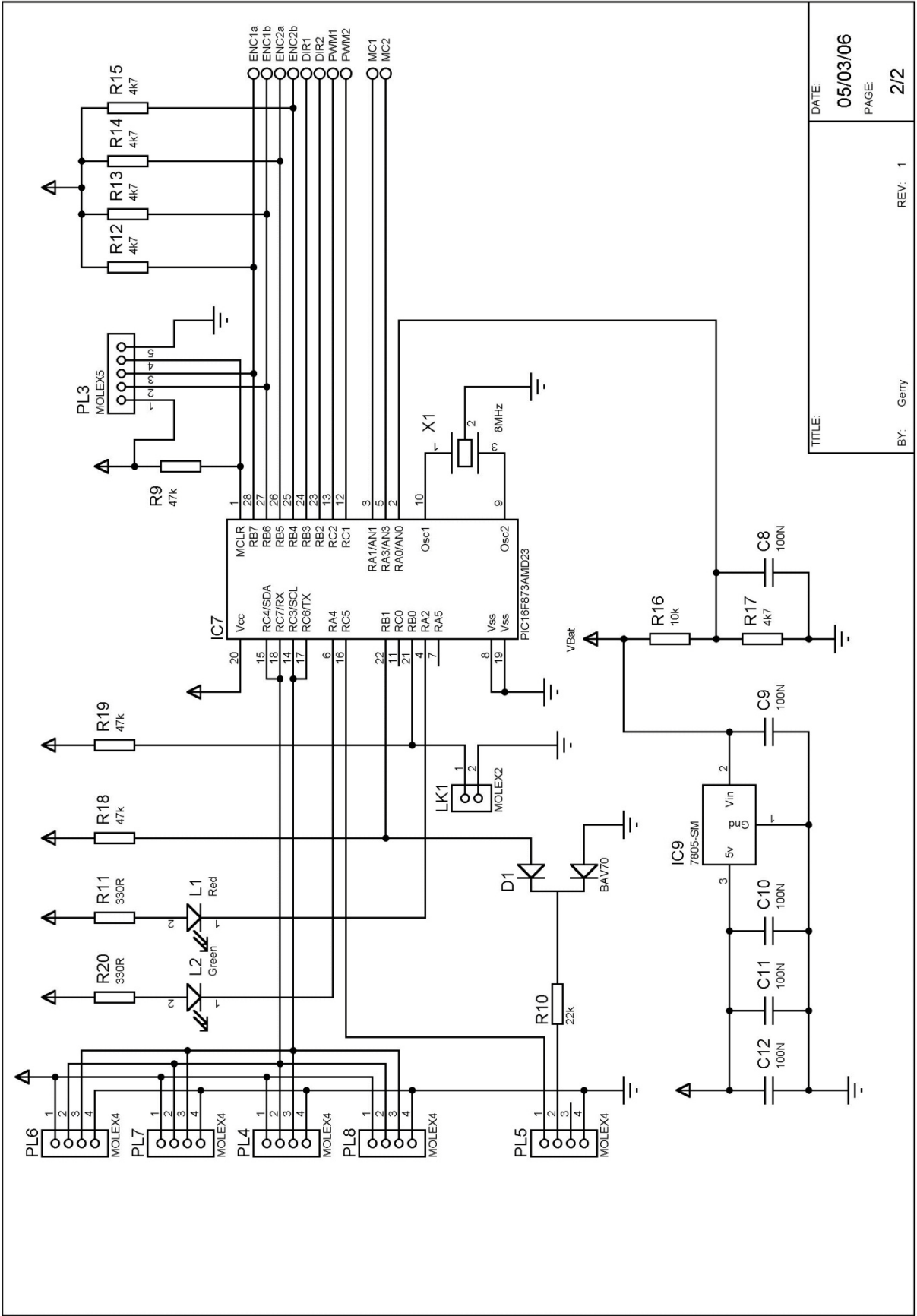
In this paper we have described reacTIVision, a fiducial tracking system for tangible user interfaces. We provided an explanation of the topological technique used to recognise fiducials, and presented advances beyond the earlier work of Costanza and Robinson in the d-touch system. Our enhancements include: the application of left heavy depth sequences for fiducial recognition and identification; a scheme for computing fiducial location and orientation which is independent of the fiducial’s specific topology; and an incomplete region adjacency graph representation for efficient image segmentation. We also improved performance by employing an efficient locally adaptive thresholding scheme. In addition, we outlined our approach to designing fiducials using evolutionary computation.

Our implementation techniques yielded a four-fold increase in performance over the current d-touch system. Our

Apéndice H - Esquemas electrónicos tarjeta MD25



TITLE:	DATE: 05/03/06
BY: Gerry	PAGE: 1/2
REV: 1	



TITLE:	DATE: 05/03/06
BY: Gerry	PAGE: 2/2
REV: 1	

REFERENCIAS

- [1] Theraulaz, G., Goss, S., Gervet, J., & Deneubourg, J.-L. *Task differentiation in Polistes wasp colonies: a model for self-organizing groups of robots*. In Proceedings of the First International Conference on Simulation of Adaptive Behavior 346-355. 1990.
- [2] Steels, L. *Cooperation between distributed agents through self-organization.*, Decentralized A.I. Elsevier Science. Yves Demazeau and Jean-Pierre Muller editors 1990.
- [3] Matriac, M.J. *Interaction and Intelligent Behavior*. PhD thesis, Massachusetts Institute of Technology, 1994.
- [4] Cai, A., Fudaka, T., Arai, F., Ueyama, T., & Sakai, A. *Hierarchical Control Architecture for Celular Robotic System – Simulation and Experiments*. Proc. Of the IEEE Int. Conference on Robotics and Automation, 1191-1196, 1995.
- [5] Mitsumoto, N., Fuduka, T., Shimojima, K., & Ogawa, A. *Micro Autonomous Robotic System and Biologically Inspired Immune Swarm Strategy as a Multi Agent Robotic System*. Proc. Of the IEEE Int. Conf. on Robotics and Automation, 2187-2192. 1995.
- [6] Sugawara, K., Kazama, T., & Watanabe, T. *Foraging behavior of interacting robots with virtual pheromone*. IEEE/RSJ International Conference, Intelligent Robots and Systems. 2004.
- [7] Purnamadjaja, A., Iskandar, J., & Russell, R. *Pheromone Communication Simulation for Mobile Robots Using Java 3D*. ICIS 2007 6th IEEE/ACIS International Conference. 261-266. 2007.
- [8] T.Fukunda, S. Nakagawa, Y. Kawauchi, and M. Buss. *Self organizing robots based on cell structures*. CEBOT. In Proceedings of 1988 IEEE International Workshop on Intelligent Robots and Systems (IROS' 88), pages 145-150, 1988.
- [9] H. Asama, K. Ozasi, A. Matsumoto, Y. Ishida, and I. Endo. *Development of tasks assignement system using communication for multiple autonomous robots*. Journal of Robotics and Mechatronics, 4(2): 122-127,1992.
- [10] J. Deneubourg, S. Goss, G. Sandini, F. Ferrari, and P. Dario. *Self-organizing collection and transport of objects in unpredictable environments*. In *Japan-U.S.A. Symposium on Flexible Automation*, pages 1093-1098, 1990
- [11] Jin, K., Liang, P., & Beni, G. *Stability of synchronized distributed control of discrete swarm structure*. IEEE ICRA. 1033-1038. 1994.
- [12] Lynne E. Parker, *Alliance: An Architecture for Fault Tolerant Multi-Robot Cooperation*, IEEE Transactions on Robotics and Automation, Vol. 14 No. 2 p. 220-240, 1998.

- [13] Arkin, R. C., & Balch, T. AuRA: Principles and Practice in Review. *Journal of Experimental & Theoretical Artificial Intelligence*. 1997
- [14] Caloud, P., Choi, W., Latomb, J.-C., Le Pape, C., & Yim, M. *Indoor automation with many mobile robots*. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, 67-72. 1990
- [15] Le Pape, C. *A combination of centralized and distributed methods for multi-agent planning and scheduling*. IEEE ICRA, 488, 493 . 1990.
- [16] Lynne E. Parker. *Path Planning and Motion Coordination in Multiple Mobile Robot Teams*. Encyclopedia of Complexity and System Science, Robert A. Meyers, Editor-in-Chief, Springer, 2009.
- [17] Ollero, Anibal. *Robótica, manipuladores y robots móviles* Editorial Alfaomega, España, pag. 8-11, 2007
- [18] Cao, Y. U., Fukunaga, A. S., & Kahng, A. B., *Cooperative Mobile Robotics: Antecedents and Directions*. In *Autonomous Robots*. 7-27. 1997
- [19] Zebrowski, P. (2004). *Communication In Multi-Robot System*.
- [20] Jordà, S. & Kaltenbrunner, M. & Geiger, G. & Bencina, R. *The Reactable*. Proceedings of the International Computer Music Conference (ICMC 2005) Barcelona, Spain. 2005.
- [21] Kaltenbrunner, M. & Bovermann, T., Bencina, R., & Costanza, E. *TUIO- A Protocol for Table Based Tangible User Interfaces*. Proceedings of the 6th International Workshop on Gesture in Human- Computer Interaction and Simulation (GW 2005), Vannes (France), 2005.
- [25] Borenstein, J., Everett, B., & Feng, L. *Navigating Mobile Robots: Systems and Techniques*. Wellesly, MA: A. K. Peters. 1996.
- [26] Negenborn, R. *Robot Localization and Kalman Filters*. Utrecht University. "BT. Localización local". 2003.
- [27] Martín, Matellán, Barrera & María. *Localización basada en Lógica Difusa y Filtros de Kalman para robots con patas*. Madrid: Grupo de robótica, Universidad Rey Juan Carlos. 2006
- [28] Arkin, R. C., Balch, T. & Nitz, E. *Communication behavioral state in multi-agent retrieval tasks*. International Conference on Robotics and Automation, 588-594. 1993.
- [29] Latombe, J. C., *Robot Motion Planning*. Boston: Kluwer Academic Publishers. 1991.
- [30] J. A. Peña Casimiro, *Localization and control of a mobile robot for the pursuit of trajectories through visual feedback*. Thesis, Department of Mechatronics, UNAM. p. 46-53, 2009

[31] Ollero, Anibal. *Robótica, manipuladores y robots móviles* Editorial Alfaomega, España, pag. 97-102. 2007

[32] González-Villela, V. J., Parkin, R. M., Lopez-Parra, M., Dorador-González, J. M., et al. (2004). "A wheeled mobile robot with obstacle avoidance capability." *Ingeniería Mecánica Tecnología y Desarrollo. Revista de la Sociedad Mexicana de Ingeniería Mecánica (SOMIM)*. ISSN 1665-7381. Vol: 1(5): 159-166.

[33] Lochmatter, Thomas., Roduit, Pierre., Cianci, Chris., Correll, Nikolaus., Jacot, Jaques and Martinoli *SwisTrack-A Flexible Open Source Tracking Software for Multy-Agent Systems*. 2008.

INTERNET

[22] http://buscon.rae.es/drael/SrvltConsulta?TIPO_BUS=3&LEMA=comunicar

[23] ZigBee Productos. Junio 2010. <http://www.zigbee.org/>

[24] Digi. Soporte. Junio 2010 <http://www.digi.com/support/eservice/login.jsp>