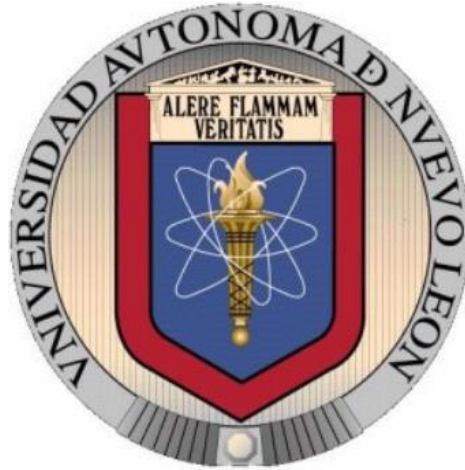


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



EFFECTO DE LOS PARÁMETROS DE REPULSIÓN, ATRACCIÓN Y
ORIENTACIÓN EN UN ENJAMBRE DE ROBOTS

POR

ING. MARIO AGUILERA RUIZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA

NOVIEMBRE, 2015

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO**



**EFFECTO DE LOS PARÁMETROS DE REPULSIÓN, ATRACCIÓN Y
ORIENTACIÓN EN UN ENJAMBRE DE ROBOTS**

**POR
ING. MARIO AGUILERA RUIZ**

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA**

NOVIEMBRE, 2015


Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Efecto de los parámetros de repulsión, atracción y orientación en un enjambre de robots», realizada por el alumno Ing. Mario Aguilera Ruiz, con número de matrícula 1500709, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica.

El Comité de Tesis



Dr. Luis Martín Torres Treviño

Asesor



Dr. Juan Ángel Rodríguez Liñán

Revisor

Dr. Oscar Salvador Salas Peña

Revisor

Vo. Bo.

Dr. Simón Martínez Martínez

Subdirección de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, noviembre 2015

AGRADECIMIENTOS

A mi asesor de tesis, el Dr. Luis Martín Torres Treviño, por su apoyo y enseñanza durante el desarrollo de esta tesis.

Al Dr. Juan Ángel Rodríguez Liñán y el Dr. Oscar Salvador Salas Peña por sus comentarios y observaciones que ayudaron a mejorar el contenido de esta tesis.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), la Universidad Autónoma de Nuevo León (UANL) y la Facultad de Ingeniería Mecánica y Eléctrica (FIME), por otorgarme la beca y los medios para realizar este proyecto.

A los profesores del Posgrado en Ingeniería Eléctrica por sus enseñanzas durante mis estudios.

A mis compañeros Angel Martínez, Efraín González, Manuel Ortiz y Luis Mercado por su apoyo y amistad durante el desarrollo de esta tesis.

A mis padres por su apoyo, guía y enseñanzas.

ÍNDICE GENERAL

Agradecimientos	IV
Índice de figuras	IX
Índice de tablas	XII
Nomenclaturas	XIII
Resumen	XV
1. Introducción	1
1.1. Motivación	1
1.2. Antecedentes	2
1.3. Planteamiento del problema	4
1.4. Hipótesis	4
1.5. Objetivos	5
1.5.1. Objetivo general	5
1.5.2. Objetivos particulares	5

1.6. Metodología	5
1.7. Contribuciones	6
1.8. Organización de la tesis	6
2. Preliminares	8
2.1. Enjambre de robots	8
2.1.1. Características y Ventajas	9
2.1.2. Aplicaciones	9
2.1.3. Arquitectura de un enjambre de robots	10
2.1.4. Comportamientos colectivos básicos	11
2.2. Modelo de comportamiento de Couzin	13
2.2.1. Reglas de comportamiento	13
2.2.2. Comportamiento exhibido por el modelo	15
2.3. Robots móviles	17
2.3.1. Configuraciones de robots con ruedas	17
2.3.2. Modelo cinemático	20
2.4. Sistemas embebidos	23
2.4.1. Definición	23
2.4.2. Arquitectura de un sistema embebido	24
3. Modelo de comportamiento implementado	26
3.1. Capacidad de percepción de los miembros del enjambre	26

3.2. Políticas de comportamiento	28
3.2.1. Política de repulsión	30
3.2.2. Política de atracción	32
3.3. Plataforma de Simulación	33
3.3.1. Representación de los robots	33
3.4. Plataforma de experimentación	35
3.4.1. Sistema embebido	36
3.4.2. Capacidad de percepción	36
4. Simulaciones	38
4.1. Descripción de los experimentos	38
4.1.1. Parámetros explorados	39
4.2. Resultados de los experimentos	41
4.2.1. Enjambre de 10 robots	41
4.2.2. Enjambre de 25 robots	45
4.2.3. Enjambre de 50 robots	49
5. Implementación en robots	52
5.1. Descripción de los experimentos	52
5.2. Resultados	54
6. Conclusiones y Trabajo Futuro	60

A. Reporte técnico de los robots	62
A.1. Diseño	62
A.2. Hardware	63
A.2.1. Microcontrolador	63
A.2.2. Sensores y Actuadores	63
A.2.3. Alimentación	65
A.3. Software	66
A.4. Funcionamiento	67
B. Código en Arduino	68

ÍNDICE DE FIGURAS

2.1. Representación de las zonas de repulsión, orientación y atracción propuestas por Couzin.	14
2.2. Comportamiento colectivo observado en el modelo de Couzin [6].	15
2.3. Configuración Ackerman	18
2.4. Configuración Triciclo	18
2.5. Configuración Diferencial	19
2.6. Configuración Síncrona	20
2.7. Sistemas de referencia local y global	21
2.8. Desplazamiento del vehículo	21
3.1. Zonas de percepción de cada robot.	27
3.2. Limitaciones de los sensores de proximidad.	28
3.3. Vectores de dirección de cada sensor.	29
3.4. Diagrama de flujo del comportamiento de cada robot.	30
3.5. Zonas de comportamiento y de percepción combinadas.	31
3.6. Configuración diferencial de un robot móvil.	33

3.7. Robot construido para realizar experimentos	35
3.8. Posición de los sensores en el robot	37
4.1. Posiciones y orientaciones iniciales usadas en las simulaciones	40
4.2. Enjambre de 10 robots con $r_r = 0.1$ y $r_a = 0.2$	41
4.3. Enjambre de 10 robots con $r_r = 0.05$ y $r_a = 0.2$	42
4.4. Enjambre de 10 robots con $r_r = 0.075$ y $r_a = 0.6$	43
4.5. Enjambre de 10 robots con $r_r = 0.075$ y $r_a = 0.6$	43
4.6. Enjambre de 10 robots con $r_r = 0.05$ y $r_a = 1.0$	44
4.7. Enjambre de 10 robots con $r_r = 0.1$ y $r_a = 1.0$	44
4.8. Enjambre de 25 robots con $r_r = 0.05$ y $r_a = 0.2$	45
4.9. Enjambre de 25 robots con $r_r = 0.1$ y $r_a = 0.2$	46
4.10. Enjambre de 25 robots con $r_r = 0.05$ y $r_a = 1.0$	47
4.11. Enjambre de 25 robots con $r_r = 0.1$ y $r_a = 1.0$	47
4.12. Enjambre de 25 robots con $r_r = 0.075$ y $r_a = 0.6$	48
4.13. Enjambre de 25 robots con $r_r = 0.075$ y $r_a = 0.6$	48
4.14. Enjambre de 50 robots con $r_r = 0.1$ y $r_a = 0.2$	49
4.15. Enjambre de 50 robots con $r_r = 0.05$ y $r_a = 0.2$	50
4.16. Enjambre de 50 robots con $r_r = 0.05$ y $r_a = 1.0$	50
4.17. Enjambre de 50 robots con $r_r = 0.1$ y $r_a = 1.0$	51
5.1. Posiciones y orientaciones iniciales en los experimentos	53

5.2. Experimento 1	54
5.3. Experimento 2	55
5.4. Experimento 3	55
5.5. Experimento 4	56
5.6. Experimento 5	56
5.7. Experimento 6	57
5.8. Experimento 7	57
5.9. Experimento 8	58
5.10. Experimento 9	58
5.11. Trayectoria de 2 robots en el experimento 2	59
5.12. Trayectoria de 2 robots en el experimento 8	59
A.1. Vista explosionada de la estructura del robot	62
A.2. Asignación de pines del sensor ultrasónico HC-SR04	64
A.3. Asignación de pines del sensor infrarrojo TCRT5000.	64
A.4. Módulo convertidor DC-DC y cargador de batería.	65
A.5. Diagrama esquemático	66
A.6. Alimentación del arduino y los motores	67

ÍNDICE DE TABLAS

3.1. Características del Arduino Micro	36
4.1. Parámetros de prueba empleados en las simulaciones	39
4.2. Radios de las zonas de comportamiento en las simulaciones	39
5.1. Radios de las zonas de comportamiento en los experimentos	53
A.1. Características del Arduino Micro	63
A.2. Características del sensor ultrasónico	64
A.3. Características de los sensores infrarrojos	64
A.4. Características de los motores	65

NOMENCLATURAS

ALU	Aritmetic and Logic Unit.
DSP	Digital Signal Processor.
EEPROM	Electrically Erasable Programmable Read-Only Memory.
FPGA	Field Programable Gate Array.
MAC	Multiplier Accumulator Unit.
MCU	Microcontrolador.
MPU	Microprocesador.
PWM	Pulse Width Modulation.
RAM	Random Access Memory.
ROM	Read Only Memory.
SRAM	Static Random Access Memory.
UART	Universal Asynchronous Receiver-Transmitter.
USB	Universal Serial Bus.
x	Posición del robot en el eje horizontal.
y	Posición del robot en el eje vertical.
ϕ	Orientación del robot.
v	Velocidad lineal del robot.
ω	Velocidad angular del robot.
c	Radio de la rueda.
b	Distancia entre las ruedas.
n	Vecinos detectados.
n_a	Vecinos en la zona de atracción.

n_o	Vecinos en la zona de orientación.
n_r	Vecinos en la zona de repulsión.
d_a	Dirección deseada de atracción.
d_r	Dirección deseada de repulsión.
r_{ij}	Vector en la dirección del vecino j.

RESUMEN

Ing. Mario Aguilera Ruiz.

Candidato para obtener el grado de Maestría en Ciencias de la Ingeniería Eléctrica.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: EFECTO DE LOS PARÁMETROS DE REPULSIÓN, ATRACCIÓN Y ORIENTACIÓN EN UN ENJAMBRE DE ROBOTS.

Número de páginas: 80.

OBJETIVOS Y MÉTODO DE ESTUDIO: El estudio de los enjambres de robots plantea un nuevo enfoque para regular la actividad de grandes cantidades de robots simples, a partir de políticas de comportamiento observadas en la naturaleza. Los enjambres de robots presentan ventajas sobre otros sistemas robóticos para la realización de ciertas tareas. En el presente trabajo se estudia el efecto que tienen las políticas de comportamiento basadas en parámetros de repulsión y atracción, implementadas en un pequeño enjambre de robots con capacidades muy simples. Se implementó una plataforma de simulación para probar estas políticas y se construyeron prototipos de robots móviles para realizar experimentos.

CONTRIBUCIONES Y CONCLUSIONES: En las simulaciones realizadas se pudo apreciar que emergen distintos comportamientos al cambiar los parámetros de repulsión

y atracción. Uno de los comportamientos más notorios es la orientación emergente en enjambres de pequeños robots. La implementación de los prototipos de experimentación demostró que es posible aplicar políticas de comportamiento en un enjambre de robots muy simples que no requiera información global o comunicación entre los robots.

Firma del asesor: _____

Dr. Luis Martín Torres Treviño

CAPÍTULO 1

INTRODUCCIÓN

1.1 MOTIVACIÓN

Actualmente los robots se utilizan para facilitar muchas de las tareas de los seres humanos, principalmente aquellas que son repetitivas y necesitan mucho tiempo o que pueden ser peligrosas. Algunas de estas tareas como monitoreo, exploración, limpieza, búsqueda y rescate, entre otras, requieren que los robots cubran grandes espacios. Para este tipo de tareas es conveniente el uso de sistemas multi-robot, donde la tarea es dividida entre varios robots para minimizar el tiempo de realización.

La coordinación de sistemas multi-robot representa un nuevo problema, para el cual un enfoque que ha tomado gran atención en los años recientes toma su inspiración en la naturaleza, específicamente en ciertos seres sociales que viven en grupos de grandes cantidades de individuos. Las observaciones de ciertos insectos, como hormigas, abejas y termitas, han demostrado que existe cierto comportamiento grupal emergente para la realización de tareas complejas fuera de las habilidades individuales de cada miembro y sin mecanismos centralizados para la coordinación del grupo [1]. El vivir en grupo brinda grandes ventajas a los individuos como mayor eficiencia para recolectar alimento [2] o menor riesgo de depredación [3].

Estos estudios tuvieron algunas de sus primeras aplicaciones en una disciplina

de la Inteligencia Artificial, conocida como Inteligencia Colectiva o Inteligencia de Enjambre, orientada hacia el desarrollo de algoritmos para resolver problemas de optimización principalmente [4].

En años recientes una disciplina conocida como Robótica de Enjambre busca emular el comportamiento de estos seres sociales pero en robots móviles relativamente sencillos. El objetivo de la robótica de enjambre es obtener comportamientos robustos, escalables y flexibles como los observados en la naturaleza.

Los enjambres de robots presentan ventajas sobre otros sistemas robóticos, para la realización de ciertas tareas. Además, existe un creciente interés en el estudio de esta área, gracias a los avances tecnológicos que hacen posible la construcción de estos sistemas.

1.2 ANTECEDENTES

Uno de los primeros modelos del movimiento de grupos de animales fue propuesto por Reynolds en 1987 [5], para animaciones computacionales del comportamiento de una parvada de aves. En su trabajo se implementan tres reglas básicas de comportamiento: 1) Evitar colisiones con miembros cercanos, 2) tratar de igualar la velocidad de miembros cercanos y 3) tratar de permanecer cerca de los otros miembros. En este modelo los individuos simulados tienen acceso directo a la posición exacta, orientación y velocidad de todos los objetos a su alrededor, lo cual representa una percepción alejada de las capacidades reales de los animales sociales en la naturaleza.

Iain Couzin et al. [6] proponen un modelo para simular el comportamiento de individuos moviéndose en grupo, como resultado de tendencias de repulsión, alineación y atracción, basadas en la posición y orientación de los individuos en relación a los demás. Establecen dos reglas de comportamiento: 1) Los individuos tratan de mantener una distancia mínima con otros, en todo momento. 2) Si los individuos

no están evadiendo a otros, entonces tienden a atraerse hacia otros individuos y alinearse con sus vecinos.

Payton et al. [7, 8] implementan un sistema de feromonas virtuales para crear un gradiente entre un punto de origen y un objetivo. Por medio de transceptores infrarrojos montados en cada robot para intercambiar mensajes, el enjambre se vuelve una malla distribuida en el ambiente, para señalar un camino hacia determinado objetivo, que otros robots o humanos puedan seguir.

Dorigo et al. [9, 10] presentaron el proyecto Swarm-bots donde exploran el diseño e implementación de robots auto-organizados y auto-ensamblables. Utilizando controles adaptativos, robots relativamente simples (s-bots) logran ensamblarse en una estructura (swarm-bot), para realizar varias tareas que requieren cooperación y coordinación. Cada s-bot está equipado con sensores para percibir el ambiente, sensores y dispositivos de comunicación para detectar y comunicarse con otros s-bots, mecanismos para realizar conexiones con otros robots y objetos, y sensores para obtener información de contactos físicos y esfuerzos en los puntos de conexión.

Fine y Shell [11] unifican varios enfoques de modelos de agrupamiento, junto con los requerimientos sensoriales de cada modelo. La mayoría de los modelos que existen requieren que cada individuo tenga información de la posición, dirección o velocidad de los otros miembros del grupo, lo que en muchos casos puede implicar diseños complejos y altos costos que dificultan la implementación física.

Finalmente, Moeslinger et al. [12] proponen un enfoque similar al planteado en este trabajo, donde no se hace uso de comunicaciones o información global. Se presenta un algoritmo para congregarse individuos inicialmente dispersos, utilizando cada individuo 4 sensores de proximidad para detectar vecinos a su alrededor.

1.3 PLANTEAMIENTO DEL PROBLEMA

El estudio de los enjambres de robots es un nuevo enfoque para regular el comportamiento de grandes cantidades de robots simples. La mayoría de los modelos propuestos utilizan información global como posición, velocidad y orientación de los vecinos cercanos, lo cual requiere que cada robot pueda conocer y transmitir esta información a los demás, mediante sistemas de comunicación que resultan demasiado complejos para grandes grupos de robots.

El utilizar pequeños robots simples conlleva limitaciones en poder computacional, alcance de los sensores y transmisión de información. Con el fin de facilitar la implementación de estos sistemas, es necesario desarrollar algoritmos basados en políticas de comportamientos existentes en la naturaleza, que requieran sólo la información local percibida por cada robot.

1.4 HIPÓTESIS

Es posible obtener distintos comportamientos colectivos para regular el funcionamiento de un enjambre de robots simples, a partir de políticas de repulsión, atracción y orientación, implementadas en cada miembro del enjambre, sin necesidad de sistemas de comunicación y utilizando sólo la información local percibida por cada robot.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Realizar un estudio sobre el efecto de los parámetros de interacción entre los miembros de un enjambre de robots al aplicar modelos matemáticos de repulsión, atracción y orientación, derivados del estudio de la inteligencia colectiva y utilizando la información local percibida por cada robot.

1.5.2 OBJETIVOS PARTICULARES

- Objetivo 1.1: Desarrollo de algoritmos para la interacción local de softbots basados en ecuaciones de repulsión, atracción y orientación.
- Objetivo 1.2: Desarrollo de una plataforma de simulación para la investigación de enjambres de robots, siguiendo reglas de interacción basadas en el comportamiento entre animales sociales.
- Objetivo 1.3: Implementación en pequeños robots simples que permitan un comportamiento similar a un enjambre, basado en ecuaciones de repulsión, atracción y orientación.

1.6 METODOLOGÍA

- Revisión bibliográfica de los trabajos existentes.
- Revisión de modelos de diferentes configuraciones de robots móviles.

- Desarrollar un algoritmo computacional para simular un enjambre de robots con las funciones de atracción, repulsión y orientación.
- Realizar experimentos con diferentes configuraciones y parámetros.
- Implementación en robots físicos para experimentación.

1.7 CONTRIBUCIONES

Las contribuciones de esta tesis son las siguientes:

- Desarrollo de una plataforma para simulación de enjambres de robots.
- Desarrollo de políticas de repulsión y atracción basadas sólo en información local percibida por cada robot.
- Estudio de los efectos de los parámetros de repulsión/atraccción en un enjambre de robots.
- Desarrollo de un pequeño enjambre de robots para implementar las políticas de comportamiento.

1.8 ORGANIZACIÓN DE LA TESIS

En el capítulo 2 se presentan algunas temas preliminares que sirven como base para el trabajo desarrollado. Se da una descripción de los enjambres de robots, sus aplicaciones y esquema general. Se presenta el modelo propuesto por Iain Couzin para el movimiento de seres en grupo. Además, se muestran diferentes configuraciones de robots móviles con su modelo cinemático. Por último se describen los sistemas embebidos y las arquitecturas más comunes.

En el capítulo 3 se describe el modelo de comportamiento implementado. Se explica la manera en que opera cada miembro del enjambre, primero en una plataforma de simulación y posteriormente en un sistema embebido.

El capítulo 4 presenta los resultados obtenidos a partir de las simulaciones. Se describen las condiciones del experimento planteado así como los parámetros utilizados. Finalmente se ilustran los resultados de simulaciones en el dominio del tiempo, mostrando las posiciones de los miembros del enjambre en diferentes instantes.

El capítulo 5 contiene los resultados obtenidos de los experimentos con robots físicos, mostrando secuencias de imágenes de los videos tomados.

Por último, el capítulo 6 muestra las conclusiones de este trabajo, así como los trabajos futuros.

CAPÍTULO 2

PRELIMINARES

2.1 ENJAMBRE DE ROBOTS

La inteligencia colectiva es una disciplina de la inteligencia artificial inspirada en la observación del comportamiento emergente en seres sociales como enjambres de abejas, colonias de hormigas, cardúmenes de peces o parvadas de aves, entre otros. Estos seres poseen habilidades individuales muy simples, pero al vivir en grandes grupos logran realizar tareas más complejas como la búsqueda y recolección de comida o escapar de algún depredador.

Los enjambres de robots se basan en la aplicación de la inteligencia colectiva al diseño y construcción de sistemas multi-robot para que exhiban características similares a las observadas en animales sociales. Utilizando grandes cantidades de robots simples, con reglas de comportamiento sencillas y sin el uso de una unidad central para controlar el enjambre, se espera que un comportamiento deseado emerja de la interacción entre robots y la interacción de robots con el ambiente.

2.1.1 CARACTERÍSTICAS Y VENTAJAS

El objetivo es que estos enjambres logren comportamientos robustos, escalables y flexibles como los observados en animales sociales [13, 14]. Algunas ventajas en comparación con otros sistemas robóticos son:

- Escalable: Debido a que las interacciones entre los miembros son locales, el enjambre es adaptable a cambios en el número de individuos.
- Económico: Gracias a la simplicidad de diseño y la producción en masa, el costo total de un enjambre de robots sería inferior al de un sólo robot complejo.
- Descentralizado: Las reglas de comportamiento de cada robot le permiten al enjambre completar su tarea sin el uso de un control centralizado. Esto es especialmente útil en ambientes donde las comunicaciones pueden presentar interrupciones o retardos de tiempo.
- Flexibilidad: Un enjambre debe ser capaz de realizar diversas tareas con la misma arquitectura y mínimos cambios de programación.

2.1.2 APLICACIONES

Un enjambre de robots puede ser útil en actividades que involucran grandes cantidades de tiempo y espacio o que pueden resultar peligrosas para los robots [15]. Algunos ejemplos de aplicaciones potenciales se encuentran en:

- Búsqueda de sobrevivientes en desastres [16].
- Detección de derrames de petróleo en el mar [17].
- Búsqueda de objetivos [18].
- Vigilancia y monitoreo de recursos [19].

- Aplicaciones militares [20].
- Exploración espacial [21].

2.1.3 ARQUITECTURA DE UN ENJAMBRE DE ROBOTS

El trabajo presentado en [11] muestra un diagrama general de las etapas necesarias para la producción de movimiento en grupo, tomando como referencia diferentes modelos encontrados en la literatura. Este diagrama consta de 5 etapas que se describen a continuación:

Percepción. Es la forma en que cada miembro del enjambre obtiene información del ambiente que lo rodea. Por lo general esta información es obtenida por medio de sensores que miden diversas magnitudes físicas, sistemas de visión computacional e inclusive sistemas de comunicación entre los miembros del enjambre para intercambiar información. La etapa de percepción es la primera, ya que convierte el mundo real en información que puede ser usada por las demás etapas del modelo del enjambre.

Detección de miembros del enjambre. En esta etapa se utiliza la información de la etapa de percepción para determinar el conjunto de todos los robots en el rango de detección. Este conjunto de individuos detectados puede incluir información como posición y velocidad entre otros, dependiendo de las necesidades de cada algoritmo.

Selección de vecinos. Tomando el conjunto de individuos detectados anteriormente, ahora se seleccionan sólo aquellos que influyen en el cálculo del movimiento del robot. Por ejemplo, en esta etapa se separarían los vecinos que están en la zona de repulsión o atracción, de acuerdo a las políticas establecidas por el modelo de comportamiento.

Cálculo de movimiento. Esta etapa se encarga de calcular el siguiente movi-

miento del robot, tomando como entradas los conjuntos de vecinos seleccionados en la etapa previa. Sólo se calcula la dirección que debe tomar el vehículo. El resultado es sólo una representación, como puede ser un vector. El movimiento real del robot se ejecuta en la etapa final.

Movimiento físico. Tomando como entrada la dirección calculada previamente, se traduce para que pueda ser ejecutada dependiendo del sistema, por ejemplo en velocidades de los motores del vehículo.

2.1.4 COMPORTAMIENTOS COLECTIVOS BÁSICOS

Existen varios comportamientos colectivos considerados como básicos, los cuales son fundamentales para la para la realización de tareas más complejas. Estos comportamientos se pueden dividir en organización espacial, navegación y toma de decisiones colectivas [13].

2.1.4.1 COMPORTAMIENTOS DE ORGANIZACIÓN ESPACIAL

Estos comportamientos están orientados hacia la organización y distribución del enjambre en un determinado ambiente.

- **Agregación.** Es el comportamiento más simple y consiste en agrupar un conjunto de robots muy cerca entre sí. Es de gran importancia al permitir la interacción entre los robots y ser la base para otros comportamientos como el movimiento en grupo.
- **Formación.** Consiste en que los robots mantengan cierta distancia entre ellos para formar un determinado patrón o figura repetible. Este comportamiento es útil en tareas en las que se puede requerir transportar objetos grandes.

- Auto ensamblaje. Se trata de las acciones que permiten a los robots conectarse entre sí físicamente, con el fin de obtener una estructura de mayor tamaño que permita realizar acciones más complejas. El contar con una estructura de mayor tamaño ayuda al enjambre a completar tareas en terrenos complicados, o para superar obstáculos.
- Agrupación de objetos. Consiste en que los robots muevan objetos distribuidos en el ambiente, para agruparlos en algún punto. Esta es una tarea muy común en los seres vivos como las hormigas y termitas, que puede ser aplicable a la recolección de alimentos o la construcción.

2.1.4.2 COMPORTAMIENTOS DE NAVEGACIÓN

Los comportamientos de navegación están orientados hacia la coordinación del movimiento de un enjambre de robots.

- Exploración colectiva. El objetivo de este comportamiento es desplegar los robots en un ambiente desconocido con el fin de crear una malla o arreglo para realizar comunicaciones entre los robots y guiar a otros hacia determinado objetivo.
- Movimiento coordinado. Basado en el movimiento de las aves y peces, este comportamiento busca que los robots se muevan en formación, lo cual es de gran importancia para que los robots puedan navegar por un ambiente sin colisiones entre ellos.
- Transportación colectiva. Cuando un objeto es demasiado grande o pesado para ser movido por un sólo robot, la transportación colectiva busca que un grupo de robots trabajen en conjunto para mover ese objeto. Es importante que exista coordinación entre los robots para que se muevan en la misma dirección.

2.1.4.3 COMPORTAMIENTOS DE DECISIÓN COLECTIVA

Este comportamiento trata el problema de la influencia que tienen los robots entre sí al momento de tomar una decisión. Ejemplos de toma de decisiones pueden ser el logro de consenso o la distribución de tareas.

- Consenso. Cuando hay varias alternativas, el enjambre debe llegar a un consenso sobre la mejor opción para maximizar su desempeño. Esta es una tarea muy compleja ya que la solución óptima puede cambiar o no ser perceptible por las capacidades del robot.
- Distribución de tareas. En este comportamiento los robots se distribuyen para realizar diferentes tareas, con el objetivo de maximizar el desempeño del sistema.

2.2 MODELO DE COMPORTAMIENTO DE COUZIN

El modelo propuesto por Iain Couzin en 2002, simula el comportamiento de individuos moviéndose en grupo, como resultado de tendencias de repulsión, alineación y atracción, basadas en la posición y orientación de los individuos en relación a los demás [6].

2.2.1 REGLAS DE COMPORTAMIENTO

Se establecen dos reglas de comportamiento para cada miembro del grupo: 1) Los individuos tratan de mantener una distancia mínima con otros, en todo momento. 2) Si los individuos no están evadiendo a otros, entonces tienden a atraerse hacia

otros individuos y alinearse con sus vecinos.

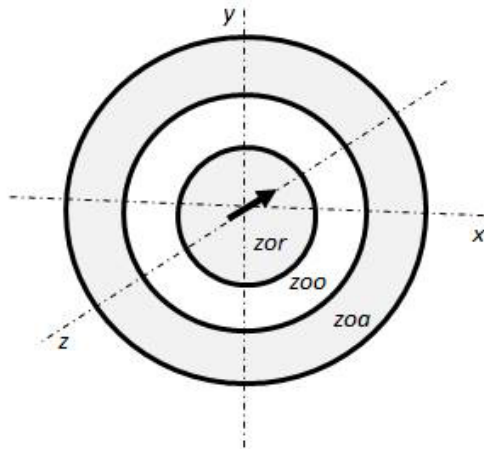


Figura 2.1: Representación de las zonas de repulsión, orientación y atracción propuestas por Couzin.

La Figura 2.1 muestra las zonas esféricas de repulsión, orientación y atracción que rodean a cada individuo en el modelo propuesto por Couzin. Si un individuo i detecta a otro individuo j en su zona de repulsión, entonces calculará su siguiente dirección en base a la ecuación:

$$d_r(t + \tau) = - \sum_{j \neq i}^{n_r} \frac{r_{ij}(t)}{|r_{ij}(t)|} \quad (2.1)$$

Si la cantidad de vecinos en su zona de repulsión es cero, entonces la dirección deseada se calcula en base a los vecinos en la zona de orientación y atracción, utilizando las ecuaciones:

$$d_o(t + \tau) = \sum_{j \neq i}^{n_o} \frac{v_j(t)}{|v_j(t)|} \quad (2.2)$$

$$d_a(t + \tau) = \sum_{j \neq i}^{n_a} \frac{r_{ij}(t)}{|r_{ij}(t)|} \quad (2.3)$$

donde $d(t + \tau)$ es dirección deseada para el individuo i , n_r es la cantidad de vecinos en la zona de repulsión, n_o es la cantidad de vecinos en la zona de orientación, n_a es la cantidad de vecinos en la zona de atracción y v_j es el vector unitario de dirección del individuo j . El término r_{ij} es el vector en la dirección del vecino j :

$$r_{ij} = c_j - c_i \quad (2.4)$$

siendo c_j la posición del individuo j con respecto al origen y c_i la posición del individuo i con respecto al origen.

2.2.2 COMPORTAMIENTO EXHIBIDO POR EL MODELO

El comportamiento colectivo exhibido por el modelo de Couzin se muestra en la Figura 2.2, donde se representan los individuos del enjambre en un espacio tridimensional.

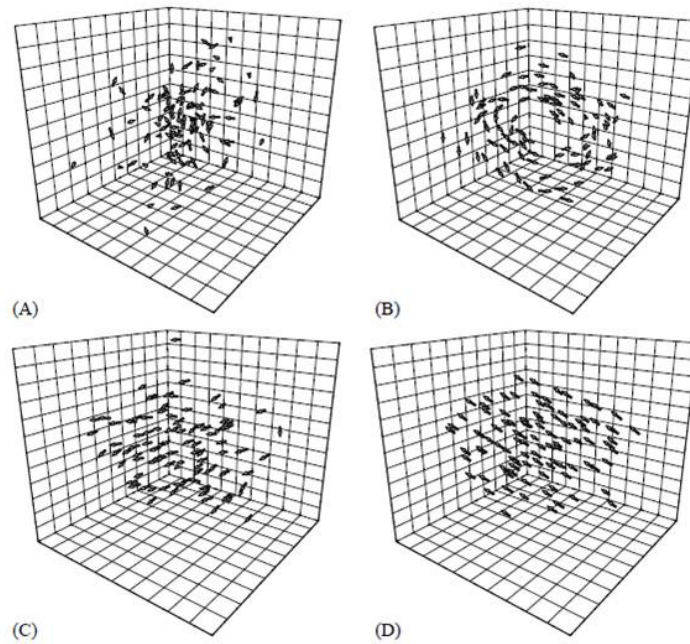


Figura 2.2: Comportamiento colectivo observado en el modelo de Couzin [6].

(A) Enjambre: Repulsión y atracción con poca o ninguna orientación. Los individuos se mueven en todas direcciones buscando evitar a otros, pero al mismo sin alejarse demasiado.

(B) Toroide: Cuando la zona de orientación es pequeña y la zona de atracción es grande, el enjambre comienza a girar alrededor de un punto.

(C) Grupo paralelo: Valores intermedios de orientación y valores intermedios o grandes de atracción, que permiten que el enjambre realice un movimiento rectilíneo

(D) Grupo altamente paralelo: Similar al comportamiento anterior, al incrementar la zona de orientación. todos los individuos se mueven en la misma dirección.

2.3 ROBOTS MÓVILES

Los robots móviles son máquinas automáticas capaces de desplazarse. Según el medio en que se mueven, podemos clasificar a los robots móviles en terrestres, acuáticos y aéreos.

En este trabajo se consideran sólo robots móviles terrestres, específicamente del tipo que utiliza ruedas para su locomoción. En general, los robots móviles terrestres se desplazan usando ruedas o piernas articuladas. La locomoción con piernas articuladas implica mecanismos más complejos que la locomoción con ruedas. Además, la locomoción con ruedas es más adecuada para superficies planas, mientras que la locomoción con piernas articuladas tiene ventajas en terrenos irregulares [22].

2.3.1 CONFIGURACIONES DE ROBOTS CON RUEDAS

La locomoción con ruedas es el mecanismo más popular en robots móviles terrestres, gracias a su implementación mecánica simple, además de eficiencia y balance en terrenos rígidos. Existen diversas configuraciones de robots con ruedas, con diferentes características de maniobrabilidad. Algunas de estas configuraciones son: Ackerman, triciclo, diferencial, pistas de deslizamiento y síncrona [23].

2.3.1.1 CONFIGURACIÓN ACKERMAN

Esta configuración es la más usada en vehículos convencionales como automóviles y vehículos de carga. Como se ilustra en la Figura 2.3, esta configuración utiliza dos ruedas delanteras para la dirección, mientras que la tracción puede aplicarse en las ruedas delanteras o traseras. La desventaja principal de esta configuración es su poca maniobrabilidad ya que sus giros están descritos por circunferencias concéntricas con el punto de intersección de la prolongación del eje de la ruedas traseras con

la prolongación de los ejes de las ruedas delanteras.

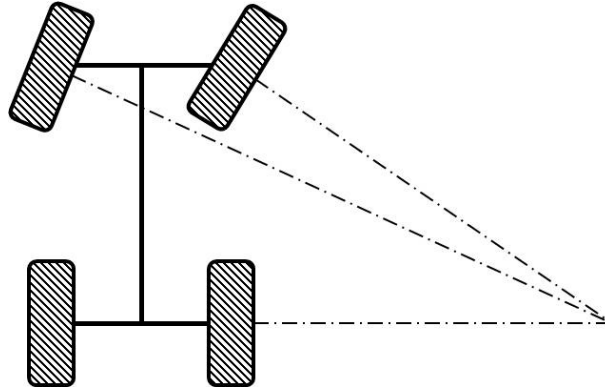


Figura 2.3: Configuración Ackerman

2.3.1.2 CONFIGURACIÓN TRICICLO

En la configuración de triciclo mostrada en la Figura 2.4, la rueda delantera se utiliza tanto para la tracción como para la dirección, mientras que las ruedas traseras se mueven libremente. Presenta mayor maniobrabilidad que la configuración Ackerman, pero la estabilidad puede verse comprometida en terrenos difíciles o pendientes.

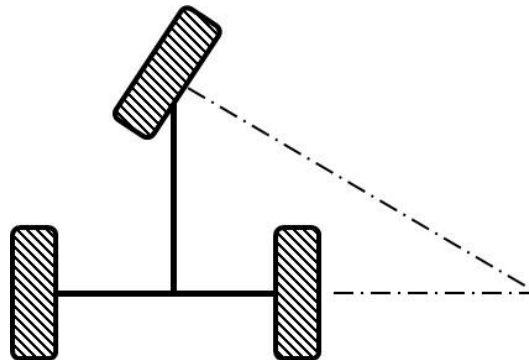


Figura 2.4: Configuración Triciclo

2.3.1.3 CONFIGURACIÓN DIFERENCIAL

El sistema de tracción diferencial de la Figura 2.5 es el más común para robots en interiores, gracias a su gran capacidad de maniobra en espacios reducidos. Esta configuración utiliza dos ruedas motoras a cada lado de la plataforma y una o más ruedas de soporte. La dirección del vehículo se obtiene a partir de la diferencia de velocidad de las dos ruedas motoras, logrando con esto realizar giros sobre su propio eje.

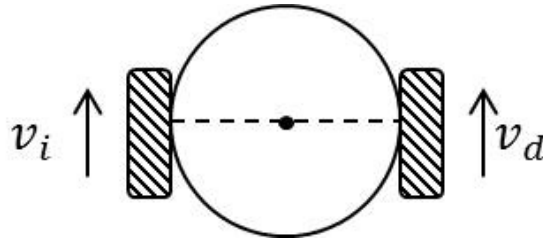


Figura 2.5: Configuración Diferencial

2.3.1.4 CONFIGURACIÓN CON ORUGAS

Este sistema de tracción utiliza cintas o bandas articuladas a cada lado del vehículo, tanto para la tracción como para el direccionamiento. Esta configuración es de gran utilidad en terrenos irregulares o muy blandos, ya que distribuye el peso de vehículos grandes en una superficie de contacto mayor. El movimiento es el resultado de la diferencia en la velocidad de las orugas, similar a la configuración diferencial.

2.3.1.5 CONFIGURACIÓN SÍNCRONA

Esta configuración consiste en la acción sincronizada de todas las ruedas. Como se ilustra en la Figura 2.6, se utiliza un motor de tracción para ajustar la velocidad de todas las ruedas y un motor de dirección para orientar las ruedas en la dirección deseada. Debido a que las ruedas giran con respecto a la plataforma del robot, no

hay forma directa de reorientar la plataforma. Es de gran utilidad cuando se busca omnidireccionalidad.

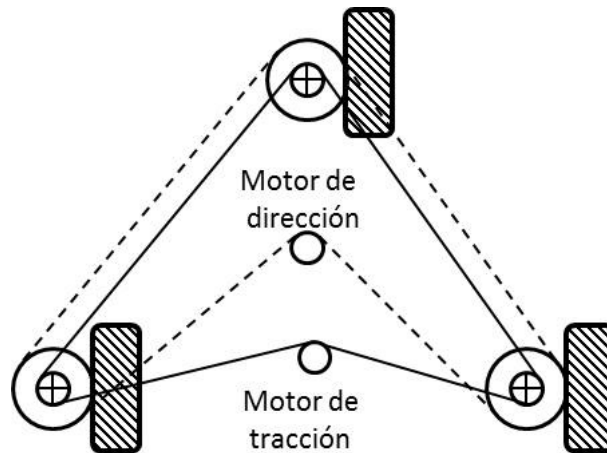


Figura 2.6: Configuración Síncrona

2.3.2 MODELO CINEMÁTICO

La cinemática es el estudio del movimiento de un cuerpo, sin considerar las fuerzas que lo producen. El objetivo del modelo cinemático directo es obtener la velocidad del robot $[\dot{x} \ \dot{y} \ \dot{\phi}]^T$ como función de las velocidades de las ruedas ω_k , las velocidades de giro α_k y los parámetros geométricos del robot.

Es necesario considerar algunas restricciones, bajo las cuales el modelo cinemático es válido. Se supone que el robot se mueve sobre una superficie plana y horizontal, que el deslizamiento de las ruedas es despreciable y que la fricción entre las ruedas y el punto de contacto es despreciable. Además, se supone que el robot se mueve con velocidad baja.

Considerando el sistema de referencia global $\{X_G, Y_G\}$ y el local $\{X_L, Y_L\}$ solidario al robot, como se muestra en la Figura 2.7 y suponiendo que el robot se desplaza en una circunferencia de radio R como se ve en la Figura 2.8, la orientación del robot ϕ se toma con respecto al eje paralelo al Y de la trama global.

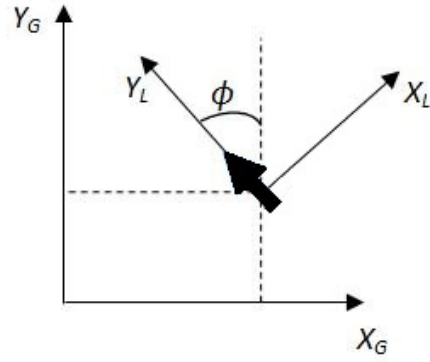


Figura 2.7: Sistemas de referencia local y global

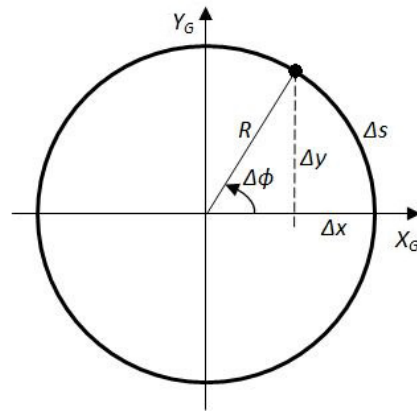


Figura 2.8: Desplazamiento del vehículo

La velocidad lineal y angular del robot están dadas respectivamente por:

$$v = \frac{\Delta s}{\Delta t} \quad (2.5)$$

$$\omega = \frac{\Delta \phi}{\Delta t} \quad (2.6)$$

con $\Delta \phi$ siendo el cambio de orientación y $\Delta s = R \Delta \phi$ el arco recorrido en un intervalo de tiempo Δt .

El desplazamiento en el marco de referencia local está dado por las ecuaciones:

$$\Delta x_L = R(\cos(\Delta\phi) - 1) \quad (2.7)$$

$$\Delta y_L = R \sin(\Delta\phi) \quad (2.8)$$

Para obtener el desplazamiento en el marco de referencia global $[\Delta x \ \Delta y]^T$, es necesario rotar un ángulo ϕ las coordenadas locales:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \Delta x_L \\ \Delta y_L \end{bmatrix} \quad (2.9)$$

Suponiendo un intervalo de tiempo muy pequeño, $\Delta\phi$ será pequeño y la ecuación (2.9) se reduce a:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -R\Delta\phi \sin \phi \\ R\Delta\phi \cos \phi \end{bmatrix} \quad (2.10)$$

Reemplazando $R\Delta\phi = \Delta s$ y dividiendo entre Δt , obtenemos las expresiones:

$$\dot{x} = -v \sin \phi \quad (2.11)$$

$$\dot{y} = v \cos \phi \quad (2.12)$$

a las cuales podemos agregar la ecuación de la variación de la orientación que se obtiene de (2.6):

$$\dot{\phi} = \omega \quad (2.13)$$

A partir de las ecuaciones (2.11), (2.12) y (2.13), el modelo cinemático directo de velocidad de un robot móvil puede representarse como:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin \phi & 0 \\ \cos \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.14)$$

En el siguiente capítulo se utiliza este modelo para representar el movimiento de un robot móvil diferencial, con el fin de poder realizar las simulaciones que se plantean en esta tesis.

2.4 SISTEMAS EMBEBIDOS

La regulación del funcionamiento del robot móvil requiere de un sistema embebido para obtener información del entorno, procesarla y desplazar al robot en la dirección adecuada. A continuación se define lo que es un sistema embebido y se describen en forma breve algunas arquitecturas de hardware usadas comúnmente.

2.4.1 DEFINICIÓN

Un sistema embebido consta de software integrado en hardware computacional, dedicado para una aplicación, o una parte específica de un sistema más grande [24]. Un sistema embebido cuenta con tres componentes principales:

- Hardware similar al de una computadora.
- El software principal de la aplicación.
- Un sistema operativo en tiempo real que supervisa la ejecución del software y la asignación de los recursos.

En cuanto a las unidades principales de hardware embebido que se pueden encontrar en un sistema, podemos mencionar que es común encontrar una unidad

de procesamiento, una fuente de energía, circuitos de reloj para control del tiempo de ejecución, unidades de memoria y puertos de entrada/salida.

2.4.2 ARQUITECTURA DE UN SISTEMA EMBEBIDO

Existen diversas arquitecturas de hardware para sistemas embebidos. Dependiendo de las necesidades particulares de cada aplicación, al momento de escoger una arquitectura se deben considerar algunas restricciones, como la cantidad de memoria del sistema, la velocidad de procesamiento, el consumo de energía y el aspecto económico. Algunas de las arquitecturas de hardware empleadas más comúnmente son las siguientes:

- Microprocesador (MPU). Es un chip que contiene varios registros, una unidad de control para la secuencia de procesamiento y una unidad lógica aritmética (ALU) para realizar operaciones básicas. Son procesadores de uso general que requieren unidades de memoria externa (ROM¹, RAM², etc.) para su funcionamiento. Se utilizan cuando es necesario un gran procesamiento computacional o cuando el software a embeber es muy extenso.
- Microcontrolador (MCU). Es un chip que integra un procesador, unidades de memoria y otras unidades de hardware. Un microcontrolador tiene menos poder computacional que un microprocesador, pero integra unidades funcionales que mejoran su capacidad de entrada-salida. Algunas de estas unidades funcionales son módulos de comunicación UART³, interrupciones digitales, temporizadores, convertidores analógico-digital y salidas moduladas por ancho de pulso (PWM), entre otras. Los microcontroladores se utilizan principalmente en sistemas de control donde se requieren algunas de las funcionalidades mencionadas, además de poder embeber el software en la memoria interna.

¹Read Only Memory.

²Random Access Memory.

³Universal Asynchronous Receiver-Transmitter.

- Digital Signal Processor (DSP). Es un chip diseñado para el procesamiento digital de señales en aplicaciones de audio y video en tiempo real. Cuenta con unidades de acumuladores multiplicativos (MAC), un amplio conjunto de instrucciones específicas para procesamiento digital, además de la capacidad para trabajar con varios datos en paralelo.
- Field Programmable Gate Array (FPGA). Es un chip que contiene una gran cantidad de arreglos de bloques de lógica programable con interconexiones reconfigurables. Los arreglos están formados por compuertas lógicas AND, OR, XOR, multiplexores, demultiplexores y bloques de memoria. La realización de cálculos complejos se logra programando adecuadamente las conexiones entre los bloques.

En este trabajo se utiliza la arquitectura de microcontrolador como sistema embebido para el robot móvil. Un microcontrolador ofrece ventajas para su utilización en un robot móvil, principalmente por la necesidad de leer información de diferentes sensores y operar motores. Además los cálculos a realizar no son muy extensos, por lo que no se requiere tanto procesamiento computacional.

CAPÍTULO 3

MODELO DE COMPORTAMIENTO IMPLEMENTADO

Este capítulo explica las reglas de comportamiento implementadas en esta tesis. Estas reglas o políticas de comportamiento se programan en cada robot para cambiar su dirección en base a los parámetros de repulsión y atracción. Se toma como base el modelo propuesto por Iain Couzin, pero se hacen algunas modificaciones para adaptarlo a robots con movimiento en dos dimensiones y con limitaciones sensoriales. Se toman consideraciones para que las políticas sean lo más sencillas posible, debido a las capacidades muy simples de los robots propuestos. Se describe el procedimiento realizado para probar estas políticas primero en una plataforma de simulación y posteriormente en una plataforma de experimentación.

3.1 CAPACIDAD DE PERCEPCIÓN DE LOS MIEMBROS DEL ENJAMBRE

En el Capítulo 2 se definió que la percepción es la forma en que cada miembro del enjambre obtiene información del medio que lo rodea. En este caso la información necesaria por cada individuo es la posición de sus vecinos, por lo que se

decidió utilizar sensores de proximidad del tipo fotoeléctrico y ultrasónico.

Los sensores ultrasónicos emiten pulsos de sonido que rebotan contra los objetos. El eco es recibido por el sensor y se registra el tiempo que transcurrió entre la emisión del pulso y la recepción del eco. La distancia hacia un objeto es calculada en base al tiempo registrado y la velocidad del sonido en el aire.

Los sensores fotoeléctricos de reflexión emiten un haz de luz infrarroja que rebota con los objetos y es captada por el receptor del sensor. Es posible calcular la distancia aproximada al objeto detectado, en base a la intensidad del haz infrarrojo que se recibe de regreso.

Para los prototipos de robot que se construirán, cada uno contará con 3 sensores de proximidad para detectar y estimar la distancia hacia otros robots u obstáculos. Estos sensores se ubican al frente, a la izquierda y derecha del robot, con lo cual podemos dividir el área de percepción de cada robot en 3 zonas Q_1 , Q_2 y Q_3 , como se ve en la Figura 3.1.

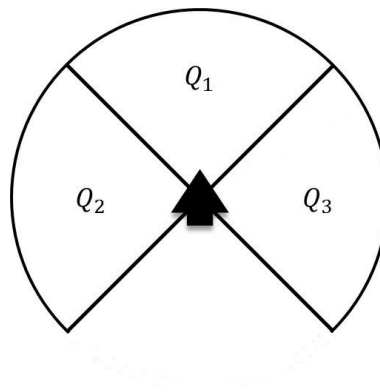


Figura 3.1: Zonas de percepción de cada robot.

Detrás de cada robot no hay ningún sensor, lo cual no afecta al momento de evitar colisiones, ya que los robots giran sobre su centro y avanzan hacia el frente, nunca se mueven en reversa. Además, este punto ciego detrás del robot concuerda con las capacidades sensoriales de los seres vivos, que cuentan con determinado rango

de visión que no cubre la totalidad del espacio a su alrededor.

El utilizar sensores de proximidad limita la capacidad sensorial del robot, ya que los individuos detrás de los vecinos más cercanos no pueden ser detectados, aunque se encuentren dentro del área de percepción, como se ve en la Figura 3.2.

Otra limitación es que el número de objetos visibles para el robot depende de la cantidad de sensores con los que cuenta. Sin importar la cantidad de robots vecinos que se encuentren en la zona de percepción de un sensor, este los considerará como uno, ya que no puede determinar la cantidad de vecinos frente a él.

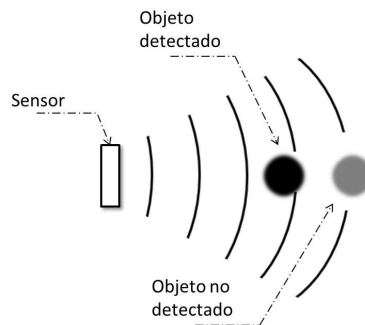


Figura 3.2: Limitaciones de los sensores de proximidad.

En resumen, la información disponible para cada robot es la distancia hacia los vecinos en la dirección de los tres sensores de proximidad. No se tiene información de la orientación ni de la cantidad exacta de vecinos.

3.2 POLÍTICAS DE COMPORTAMIENTO

En el modelo propuesto por Couzin, el cálculo de la dirección deseada para cada miembro del enjambre se hace en base a una suma de todos los vectores en la dirección de los vecinos detectados.

Como se pretende implementar estas políticas en un conjunto de robots reales, es necesario considerar la capacidad sensorial de cada robot, con el fin de que las

políticas de comportamiento funcionen tanto en las simulaciones como en la plataforma de experimentación.

Las limitaciones sensoriales presentadas en la sección anterior nos hacen considerar sólo 3 vectores en la dirección de los 3 sensores con los que cuenta cada robot. La Figura 3.3 muestra los tres vectores unitarios de dirección en el sistema de coordenadas local del robot en cuestión.

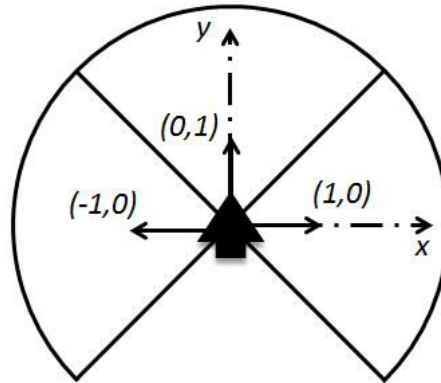


Figura 3.3: Vectores de dirección de cada sensor.

El espacio alrededor de cada robot se divide en 3 zonas de comportamiento: repulsión, orientación y atracción, definidas por los radios r_r , r_o y r_a , respectivamente.

Debido a las limitaciones sensoriales de los robots propuestos, no es posible conocer la orientación de los individuos vecinos. Esto implica que no se establece una política de orientación, sino que los robots mantienen su misma dirección, buscando que la alineación emerja de las políticas de repulsión y atracción. La Figura 3.4 muestra el diagrama de flujo que sigue cada robot para tomar la decisión de su siguiente movimiento.

Después de tomar las lecturas de los sensores, se determina la cantidad de vecinos en cada una de las zonas de comportamiento y en base a esto se calcula una dirección de repulsión d_r , una dirección de atracción d_a , o se mantiene la dirección anterior.

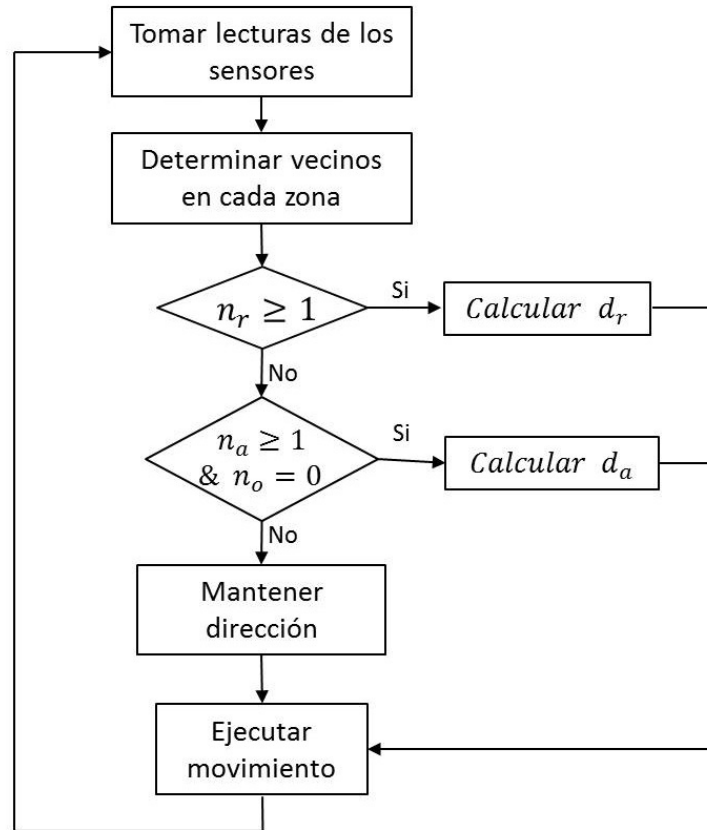


Figura 3.4: Diagrama de flujo del comportamiento de cada robot.

3.2.1 POLÍTICA DE REPULSIÓN

Si un robot detecta a otros en su zona de repulsión, cambiará su dirección en base a la ecuación:

$$d_r = - \sum_{j=1}^{n_r} r_{ij} \quad (3.1)$$

donde r_{ij} es el vector en la dirección del vecino j , que en este caso sólo puede tener el valor de los vectores unitarios en la dirección de los sensores como se ve en la Figura 3.3. La cantidad de vecinos en la zona de repulsión, n_r , está limitada por la cantidad de sensores, $0 \leq n_r \leq 3$.

La ecuación (3.1) puede expresarse como:

$$d_r = -(0, 1)q_{1r} - (-1, 0)q_{2r} - (1, 0)q_{3r} \quad (3.2)$$

donde el valor de q_{kr} será de cero si no se detecta ningún individuo en la sección Q_{kr} de la Figura 3.5, o será de uno si se detecta al menos un individuo.

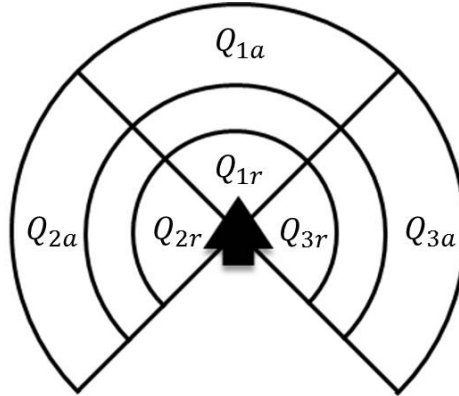


Figura 3.5: Zonas de comportamiento y de percepción combinadas.

$$q_{kr} = \begin{cases} 1, & \text{si } n_{Q_{kr}} \geq 1 \\ 0, & \text{si } n_{Q_{kr}} = 0 \end{cases} \quad (3.3)$$

donde:

$n_{Q_{kr}}$ es la cantidad de vecinos en la zona Q_{kr} .

$k = 1, 2, 3$.

El resultado será un vector que apunta en la dirección promedio opuesta a las secciones de repulsión en las que se detectaron robots.

3.2.2 POLÍTICA DE ATRACCIÓN

Cuando no se detecta ningún robot en la zona de repulsión, la dirección se calcula en base a los robots detectados en la zona de atracción con la ecuación:

$$d_a = \sum_{j=1}^{n_a} r_{ij} \quad (3.4)$$

siendo n_a la cantidad de vecinos en la zona de atracción, limitado por la cantidad de sensores, $0 \leq n_a \leq 3$.

Al igual que con la dirección de repulsión, la ecuación (3.4) se puede expresar como:

$$d_a = (0, 1)q_{1a} + (-1, 0)q_{2a} + (1, 0)q_{3a} \quad (3.5)$$

donde el valor de q_{ka} será de cero si no se detecta ningún individuo en la sección Q_{ka} de la Figura 3.5, o será de uno si se detecta al menos un individuo.

$$q_{ka} = \begin{cases} 1, & \text{si } n_{Q_{ka}} \geq 1 \\ 0, & \text{si } n_{Q_{ka}} = 0 \end{cases} \quad (3.6)$$

donde:

$n_{Q_{ka}}$ es la cantidad de vecinos en la zona Q_{ka} .

$k = 1, 2, 3$.

El resultado es un vector que apunta en la dirección promedio de las secciones de atracción en las que se detectaron robots.

3.3 PLATAFORMA DE SIMULACIÓN

El objetivo de esta plataforma de simulación es poder realizar tantos experimentos de comportamiento de enjambres como sean necesarios, con un número muy elevado de robots y sin costo de hardware. Además, las simulaciones nos permiten explorar el desempeño del modelo propuesto en condiciones que actualmente serían difíciles de implementar en un sistema físico. La plataforma de simulación está implementada en la versión 5.5.2 de Scilab, un software de código abierto para cómputo numérico [25].

Para las simulaciones es necesario un modelo que represente a cada robot en el espacio, la capacidad de percepción de cada robot y las políticas de comportamiento de cada robot.

3.3.1 REPRESENTACIÓN DE LOS ROBOTS

Para la representación de cada miembro del enjambre, se utiliza el modelo cinemático de un robot móvil con configuración diferencial como en la Figura 3.6. Como se presentó en el Capítulo 2, esta configuración es la más común para robots en interiores, gracias a su gran capacidad de maniobra en espacios reducidos.

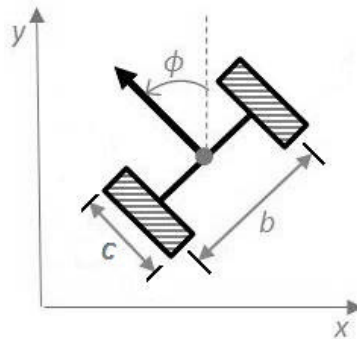


Figura 3.6: Configuración diferencial de un robot móvil.

El modelo cinemático directo de velocidad para un robot móvil está dado por:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin \phi & 0 \\ \cos \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.7)$$

Para la configuración diferencial, la velocidad lineal (v) y angular (ω) están determinadas por la velocidad de cada rueda, como se ve muestra a continuación:

$$v = \frac{(\omega_d + \omega_i)c}{2} \quad (3.8)$$

$$\omega = \frac{(\omega_d - \omega_i)c}{b} \quad (3.9)$$

donde ω_d y ω_i son respectivamente las velocidades de giro de la rueda derecha e izquierda, b es la distancia entre las dos ruedas y c es el radio de la rueda.

Sustituyendo las ecuaciones (3.8) y (3.9) en (3.7), podemos expresar el modelo en función de las variables de control ω_d y ω_i :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -c(\sin \phi)/2 & -c(\sin \phi)/2 \\ c(\cos \phi)/2 & c(\cos \phi)/2 \\ c/b & -c/b \end{bmatrix} \begin{bmatrix} \omega_d \\ \omega_i \end{bmatrix} \quad (3.10)$$

Para estimar la posición y orientación del robot $[x \ y \ \phi]^T$, se necesita integrar el modelo cinemático directo de velocidad.

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \phi_0 \end{bmatrix} + \begin{bmatrix} \int_0^t -v \sin \phi \, d\tau \\ \int_0^t v \cos \phi \, d\tau \\ \int_0^t \omega \, d\tau \end{bmatrix} \quad (3.11)$$

Utilizando estas expresiones es posible estimar la posición y orientación de cada robot, a partir de la posición previa, la orientación previa y la velocidad actual de cada rueda.

Como se sabe, el modelo cinemático no considera las fuerzas que causan el movimiento. Los resultados son aceptables bajo las condiciones de que no exista deslizamiento de las ruedas, que la carga del vehículo sea ligera y que el vehículo se mueva con baja velocidad y aceleración. Estas consideraciones son razonables para el robot propuesto para los experimentos físicos que se realizarán en este trabajo, razón por la cual se opta por utilizar un modelo cinemático.

3.4 PLATAFORMA DE EXPERIMENTACIÓN

Para la plataforma experimental se construyeron 5 robots móviles con un diámetro de 12.5cm, como el que se ilustra en la Figura 3.7.

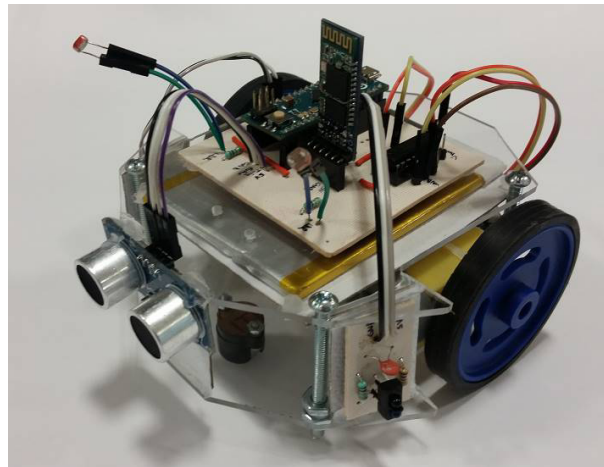


Figura 3.7: Robot construido para realizar experimentos

Este robot cuenta con diversos sensores para percibir su entorno, una tarjeta electrónica para procesar la información y dos motores de corriente directa para desplazarse. A continuación se detalla de manera breve algunas de las características de esta plataforma. Una descripción más detallada del diseño y los componentes del robot se anexa en el Apéndice A.

3.4.1 SISTEMA EMBEBIDO

La parte del procesamiento se realiza en una placa Arduino Micro [26], la cual incorpora un microcontrolador de 8 bits, comunicación USB y todos los elementos necesarios para su funcionamiento. La Tabla 3.1 muestra un resumen de las características de esta placa.

Tabla 3.1: Características del Arduino Micro

Microcontrolador	ATmega32u4
Voltaje de operación	5 V
Pines digitales I/O	20
Canales PWM	7
Canales de entrada analógica	12
Corriente DC por pin I/O	40 mA
Memoria Flash	32 KB
SRAM	2.5 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

La programación del microcontrolador se realizó en el entorno de desarrollo integrado de Arduino (Arduino IDE 1.6.3).

3.4.2 CAPACIDAD DE PERCEPCIÓN

Cada robot cuenta con un sensor de ultrasonido al frente, un sensor infrarrojo a la izquierda y otro a la derecha. El rango máximo de detección del sensor de ultrasonido es de 400cm, mientras que los infrarrojos tienen un alcance aproximado de 35cm. Una limitación que tienen estos sensores es el no poder distinguir entre un robot y un objeto, como la pared que delimita el área de trabajo.

Además cada robot tiene dos fotorresistencias ubicadas en la parte frontal

superior. Esto con el fin de futuros experimentos donde pueda ser requerido que el enjambre localice cierto objetivo, el cual podría ser representado por una fuente de luz.

La Figura 3.8 muestra la ubicación de los sensores.

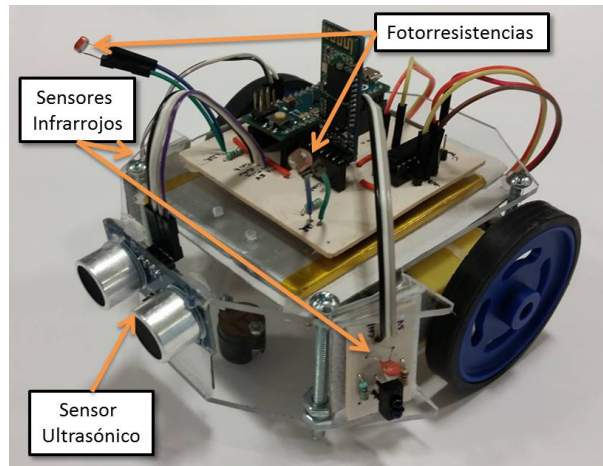


Figura 3.8: Posición de los sensores en el robot

CAPÍTULO 4

SIMULACIONES

En este capítulo se muestran los resultados obtenidos en la simulaciones, a partir del modelo de comportamiento propuesto en el capítulo 3. Primeramente se hace una descripción de los experimentos y los parámetros utilizados. El objetivo es observar los cambios que surgen en el comportamiento del enjambre al cambiar los parámetros de repulsión y atracción. Se incluyen algunos de los resultados obtenidos más representativos, utilizando enjambres de 10, 25 y 50 robots simulados.

4.1 DESCRIPCIÓN DE LOS EXPERIMENTOS

Las pruebas realizadas en la plataforma de simulación consisten en colocar los robots en una determinada posición y orientación inicial, para observar la evolución del enjambre con diferentes parámetros.

Para las pruebas con 10 robots, el área de prueba es un cuadrado de 2 x 2 m, mientras que para 25 y 50 robots se utilizó un área de 3 x 3 m. Cuando un robot alcanza el límite de esta área realiza un giro de 180 grados.

Cada paso de simulación k , corresponde a un intervalo de tiempo de 0.1 s.

4.1.1 PARÁMETROS EXPLORADOS

Con el fin de estudiar el efecto que tienen los parámetros de repulsión y atracción en un enjambre, se realizan pruebas empleando diferentes valores, así como también cambiando la cantidad de robots en el enjambre. La Tabla 4.1 resume los parámetros considerados para las simulaciones, mientras que la Tabla 4.2 muestra las combinaciones de los radios de repulsión y atracción empleados. Los 9 experimentos mostrados en la Tabla 4.2 se repiten con 5 diferentes formaciones iniciales, con 10, 25 y 50 robots, lo que da un total de 135 simulaciones.

Parámetro	Valores empleados
Cantidad de robots	10 - 50
Radio de repulsión r_r	0.05 - 0.1 m
Radio de orientación r_o	0.15 m
Radio de atracción r_a	0.2 - 1 m
Velocidad	0.1 m/s
Área de prueba	2x2 - 3x3 m

Tabla 4.1: Parámetros de prueba empleados en las simulaciones

Experimento	r_r	r_o	r_a
1	0.05	0.15	0.2
2	0.05	0.15	0.6
3	0.05	0.15	1.0
4	0.075	0.15	0.2
5	0.075	0.15	0.6
6	0.075	0.15	1.0
7	0.1	0.15	0.2
8	0.1	0.15	0.6
9	0.1	0.15	1.0

Tabla 4.2: Radios de las zonas de comportamiento en las simulaciones

La Figura 4.1 muestra las 5 configuraciones iniciales de posición y orientación empleadas, incluyendo una con posición y orientación aleatorias.

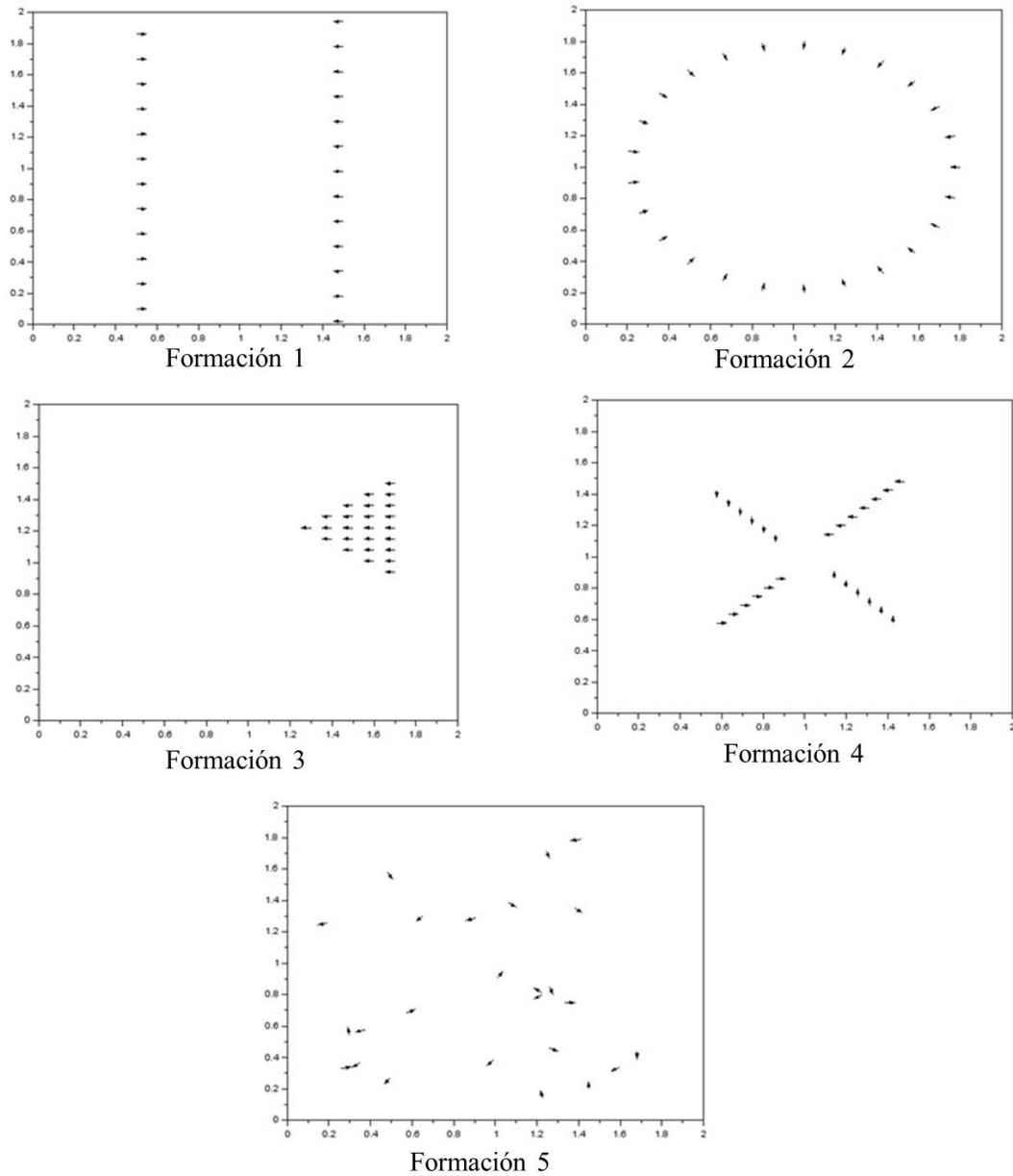


Figura 4.1: Posiciones y orientaciones iniciales usadas en las simulaciones

4.2 RESULTADOS DE LOS EXPERIMENTOS

4.2.1 ENJAMBRE DE 10 ROBOTS

En esta sección se muestran los resultados obtenidos en las simulaciones con 10 robots. Al igual que en las simulaciones con 25 y 50 robots, por cuestiones de espacio no se incluyen todas las imágenes, sólo las más representativas de los comportamientos obtenidos.

Después de realizar todas las simulaciones con 10 robots, con los parámetros antes mencionados, se observa que utilizando valores bajos de atracción como en las Figuras 4.2 y 4.3 no se muestran resultados notorios, sin importar los valores del radio de repulsión.

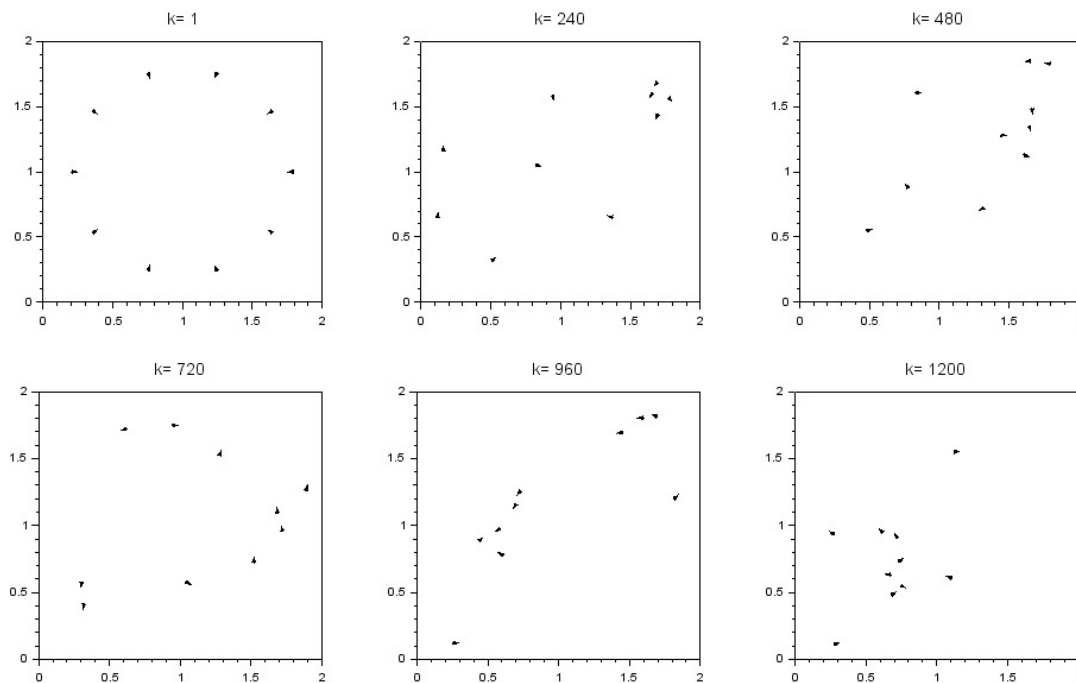


Figura 4.2: Enjambre de 10 robots con $r_r = 0.1$ y $r_a = 0.2$

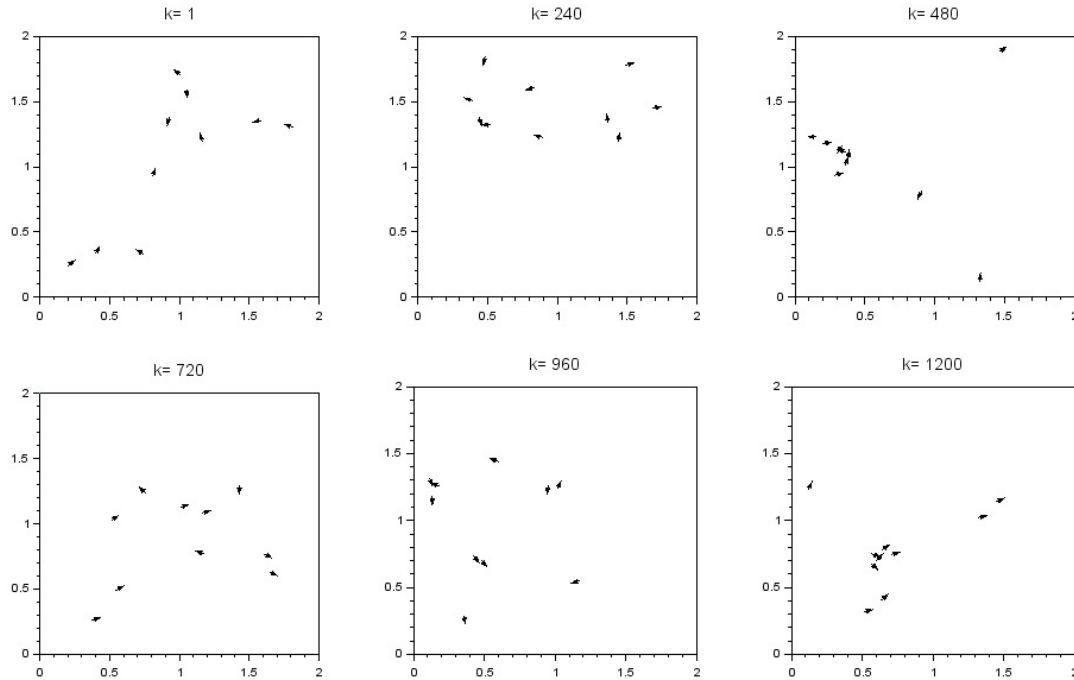


Figura 4.3: Enjambre de 10 robots con $r_r = 0.05$ y $r_a = 0.2$

Sin embargo, utilizando valores medios de atracción como en las Figuras 4.4 y 4.5 o valores altos como en las Figuras 4.6 y 4.7, se observa que existe cierta alineación entre algunos grupos de robots. A pesar de no existir una política explícita de orientación, esta alineación emerge como resultado de las interacciones de repulsión y atracción. Estos grupos se mueven en la misma dirección hasta que llegan al límite del área de prueba o más robots tratan de unirse al grupo.

Mientras más robots tratan de unirse a un grupo, más difícil es que todos se muevan en la misma dirección, lo que causa que el grupo se comporte como enjambre, atrayéndose y repeliéndose.

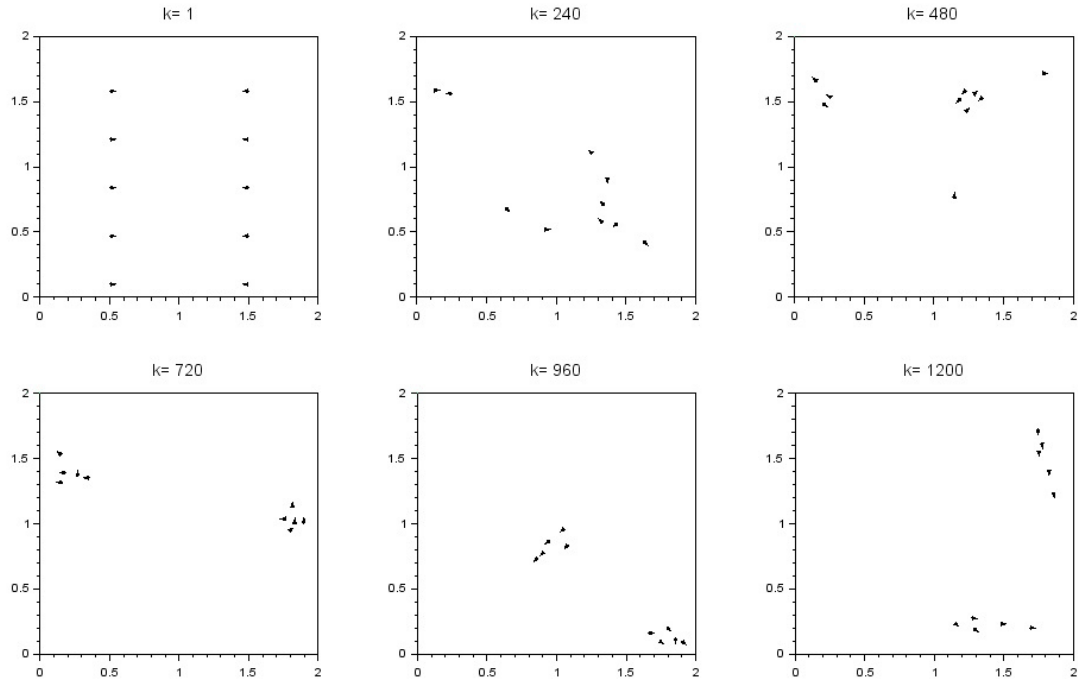


Figura 4.4: Enjambre de 10 robots con $r_r = 0.075$ y $r_a = 0.6$

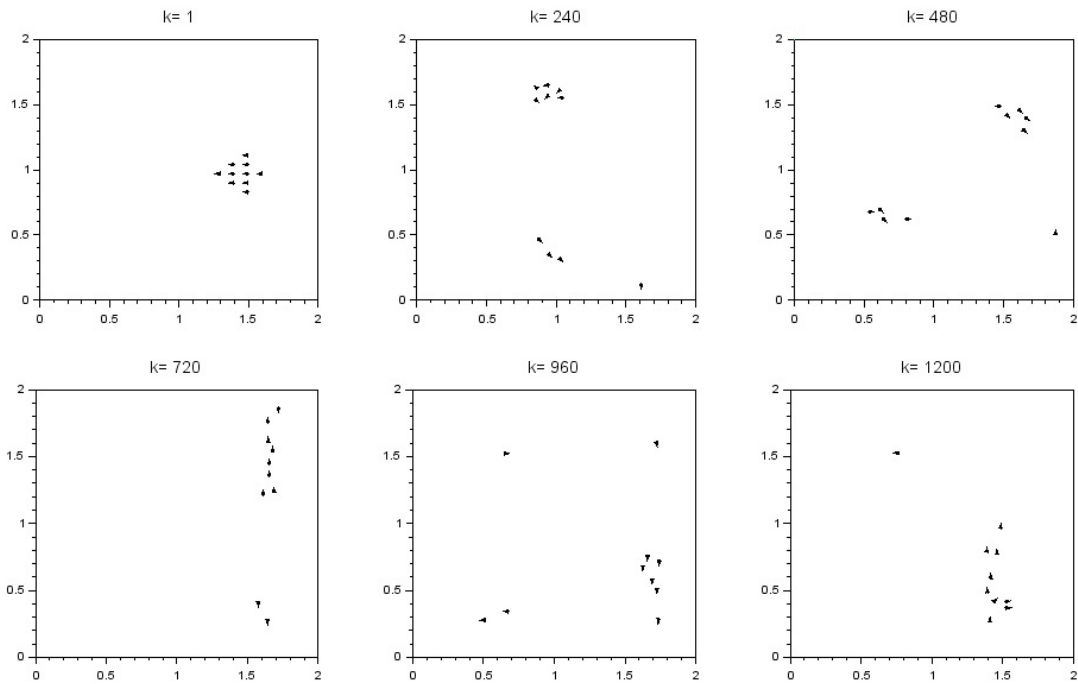


Figura 4.5: Enjambre de 10 robots con $r_r = 0.075$ y $r_a = 0.6$

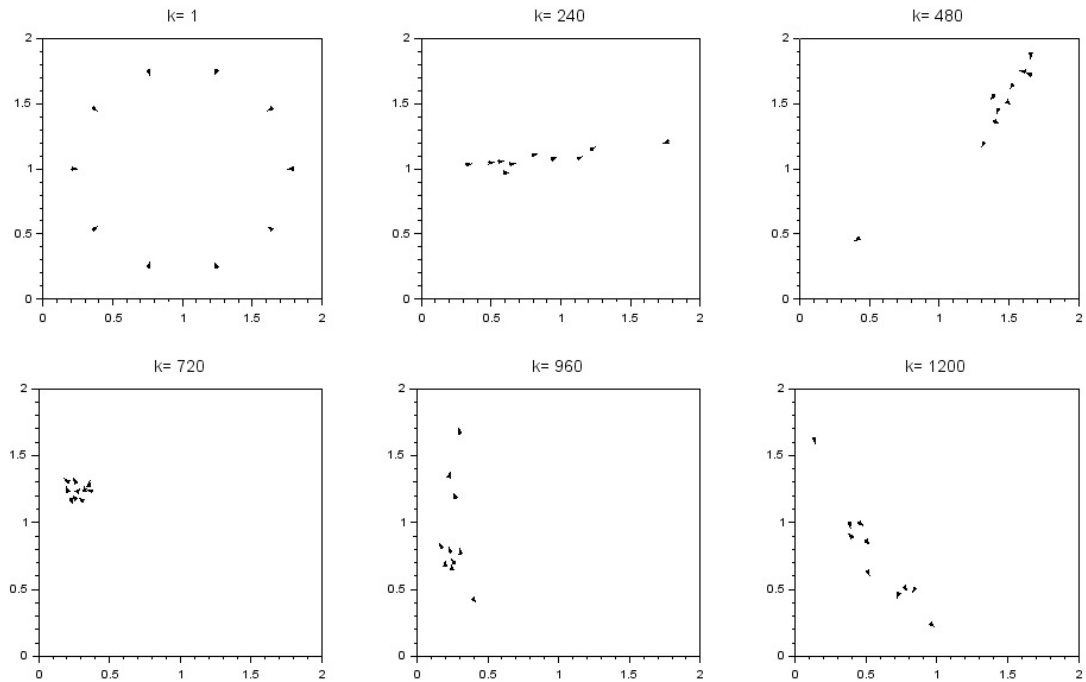


Figura 4.6: Enjambre de 10 robots con $r_r = 0.05$ y $r_a = 1.0$

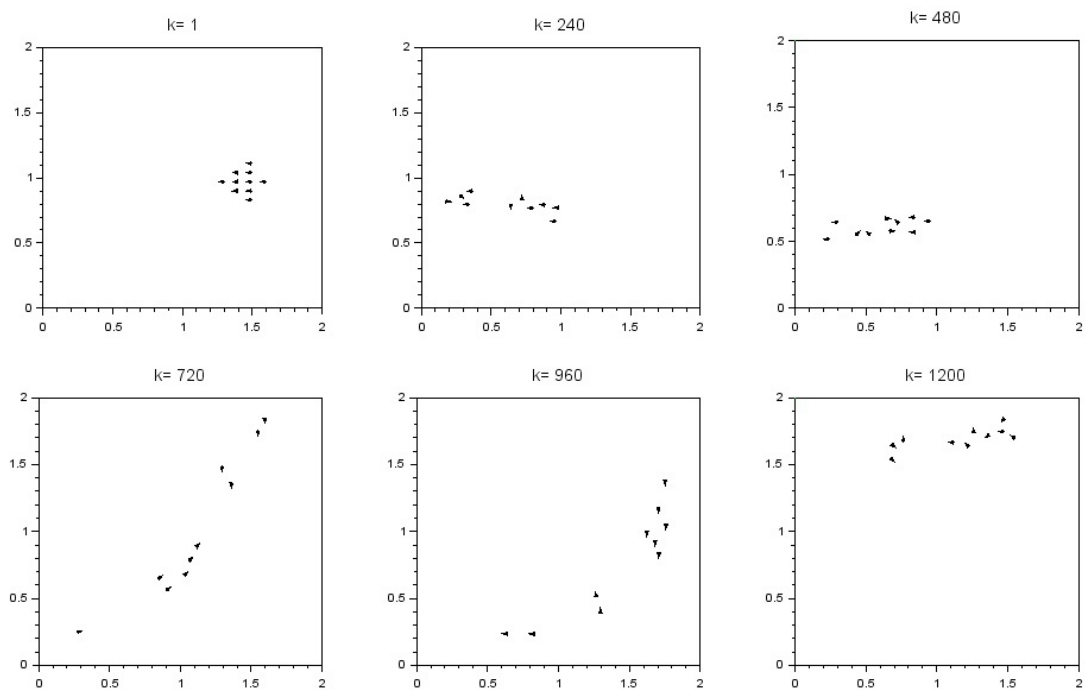


Figura 4.7: Enjambre de 10 robots con $r_r = 0.1$ y $r_a = 1.0$

4.2.2 ENJAMBRE DE 25 ROBOTS

Utilizando 25 robots es más notorio el efecto que tienen el radio de repulsión y el radio de atracción en la formación del enjambre. Si el valor de atracción es bajo, sólo los vecinos más cercanos pueden ser detectados por un robot y como resultado se forman pequeños grupos que permanecen dispersos como en las Figuras 4.8 y 4.9. Este es un comportamiento que podría resultar provechoso para realizar tareas de exploración o búsqueda, ya que permitiría a los robots partir de un punto y distribuirse en el área para agilizar la tarea.

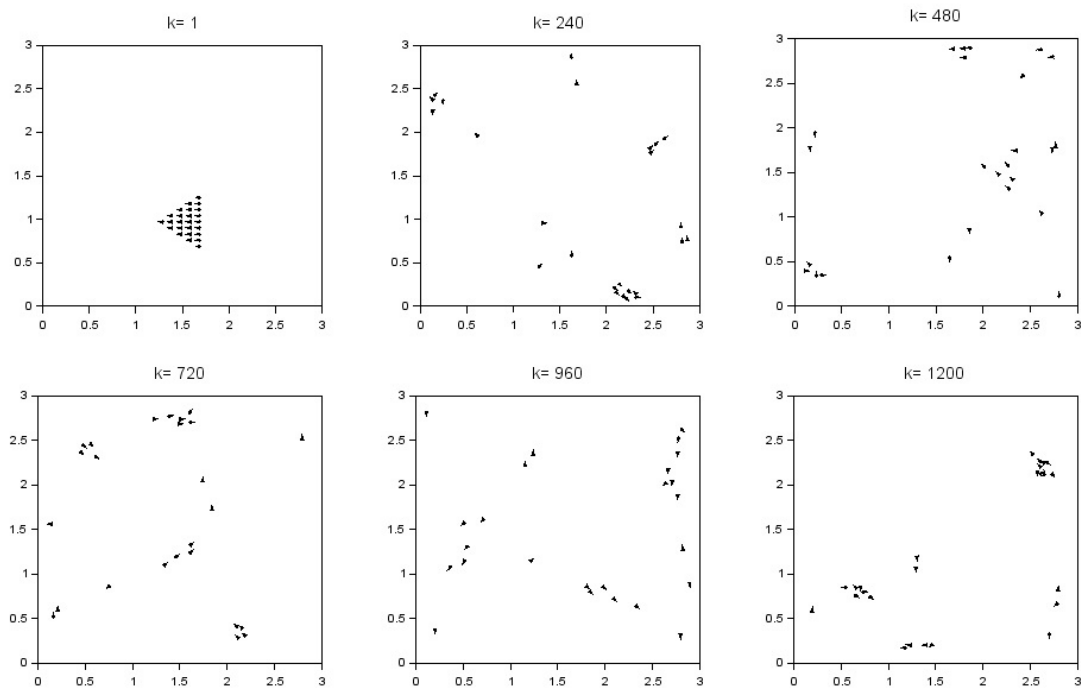


Figura 4.8: Enjambre de 25 robots con $r_r = 0.05$ y $r_a = 0.2$

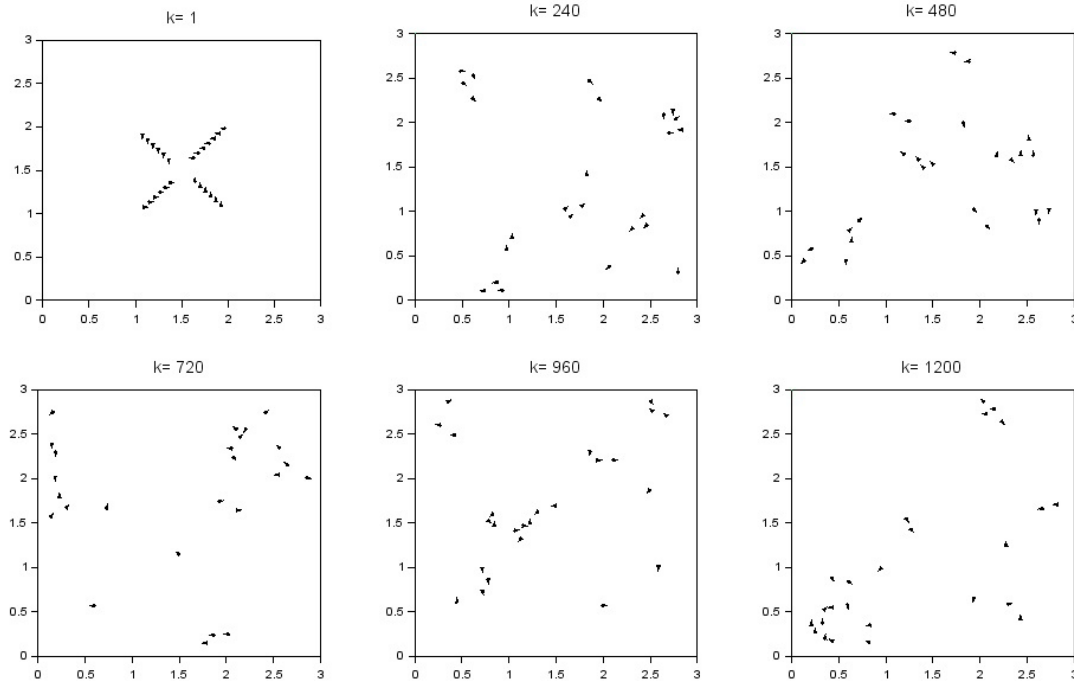


Figura 4.9: Enjambre de 25 robots con $r_r = 0.1$ y $r_a = 0.2$

En cambio, si se utilizan valores altos de atracción como en las Figuras 4.10 y 4.11, más vecinos pueden ser detectados por cada robot, ocasionando que casi el total de miembros se junten en un solo grupo. En este caso, el efecto que tiene el parámetro de repulsión es mantener el espacio individual entre los miembros del enjambre para evitar colisiones. En la Figura 4.10 el bajo valor de repulsión hace que todos los miembros estén muy cerca entre sí, mientras que en la Figura 4.11 hay más repulsión, por lo que el enjambre se distribuye más.

En el caso de valores medios tanto de repulsión como de atracción, ocurre algo similar a lo que se presentó en los enjambres de 10 robots, donde se forman pequeños grupos que logran alinearse por ciertos momentos y se mueven en la misma dirección. Este comportamiento es ejemplificado en las Figuras 4.12 y 4.13.

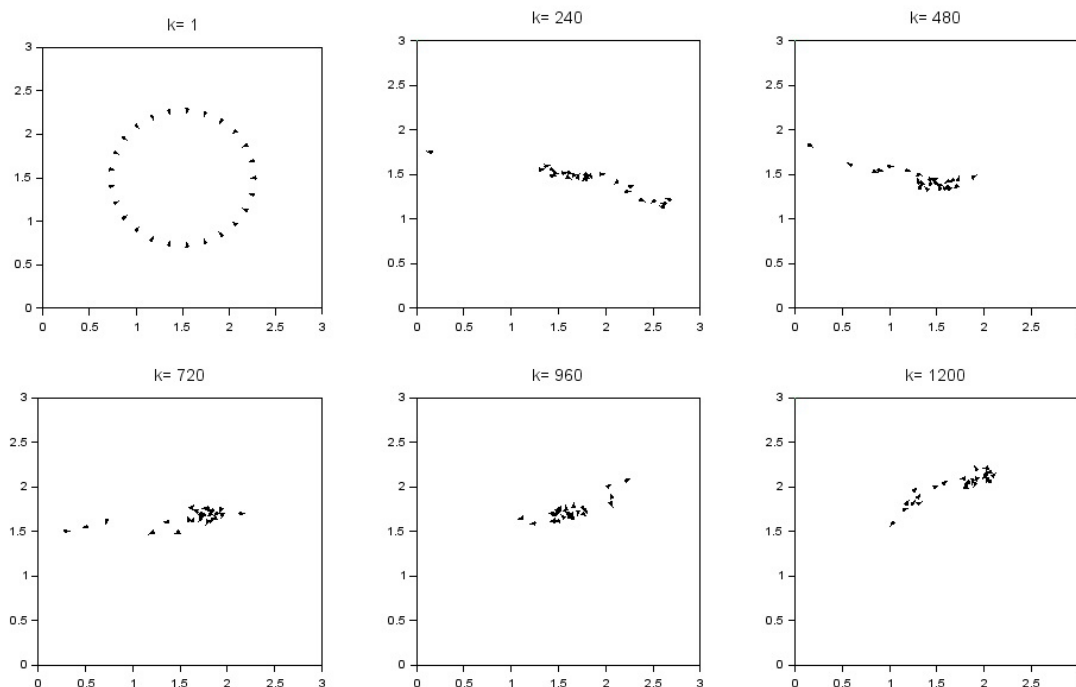


Figura 4.10: Enjambre de 25 robots con $r_r = 0.05$ y $r_a = 1.0$

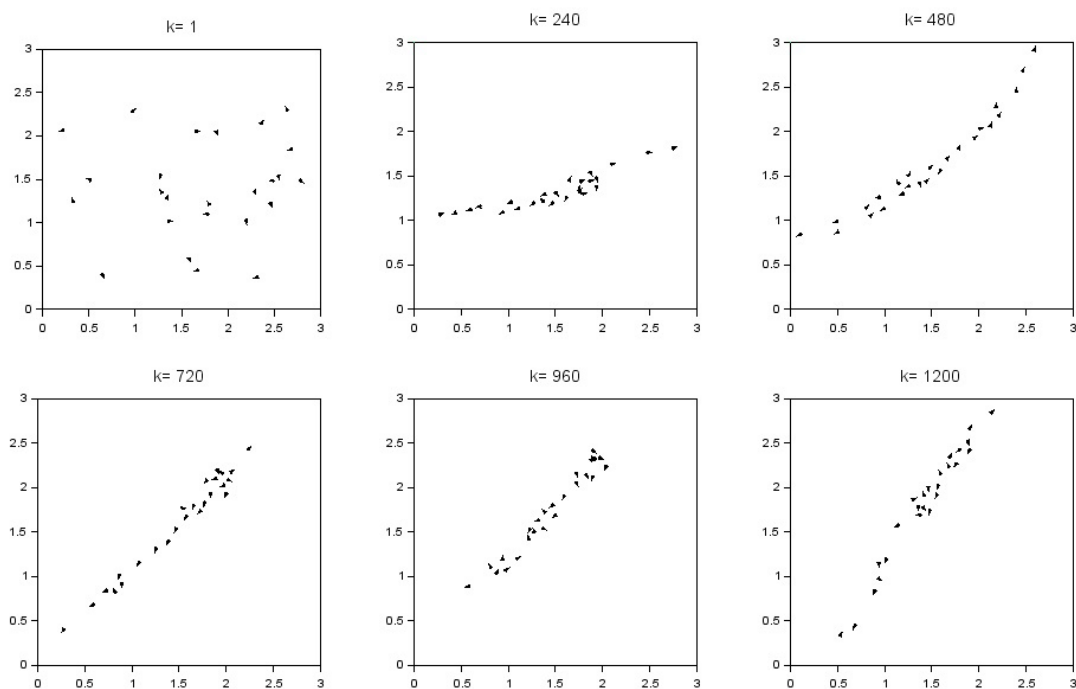


Figura 4.11: Enjambre de 25 robots con $r_r = 0.1$ y $r_a = 1.0$

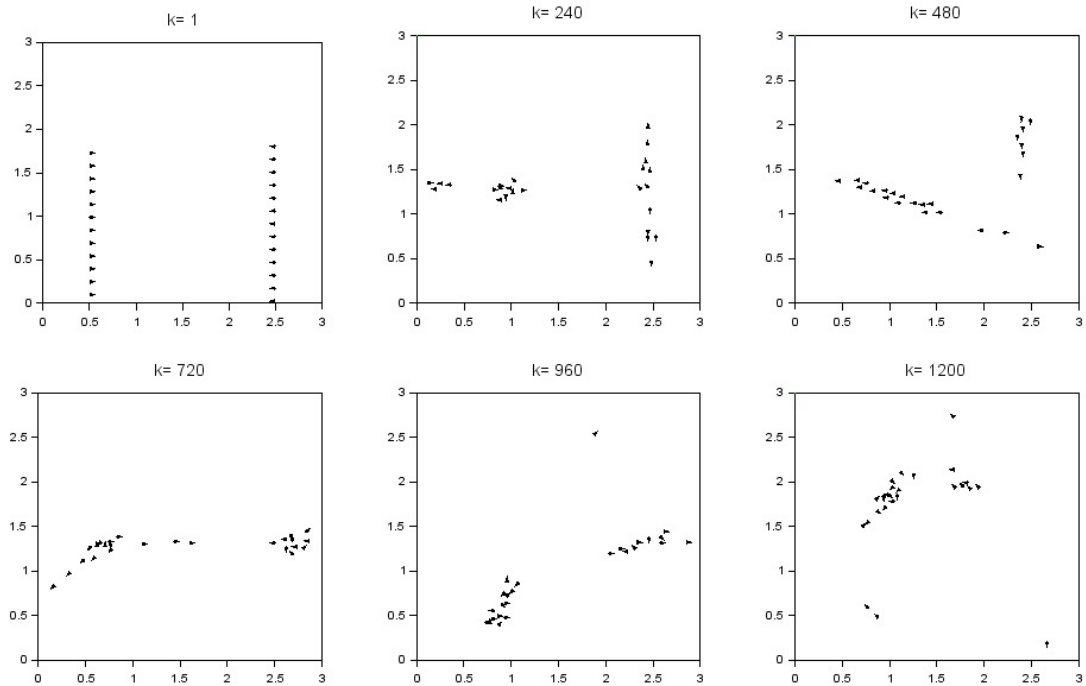


Figura 4.12: Enjambre de 25 robots con $r_r = 0.075$ y $r_a = 0.6$

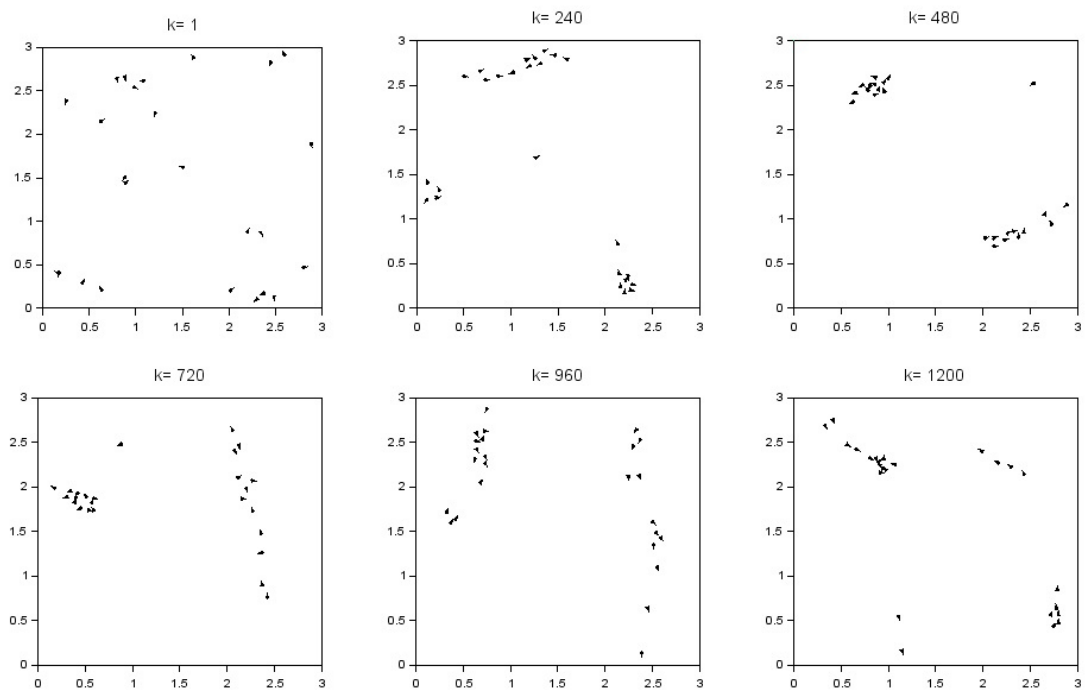


Figura 4.13: Enjambre de 25 robots con $r_r = 0.075$ y $r_a = 0.6$

4.2.3 ENJAMBRE DE 50 ROBOTS

Similar a los resultados con 25 robots, utilizando 50 también es notorio que valores bajos de atracción dispersan al enjambre en el área, sin importar la posición inicial en la que se encontraban (Figuras 4.14 y 4.15).

Si se amplía el radio de atracción, entonces los robots tienden a formar un sólo grupo unido como en las Figuras 4.16 y 4.17.

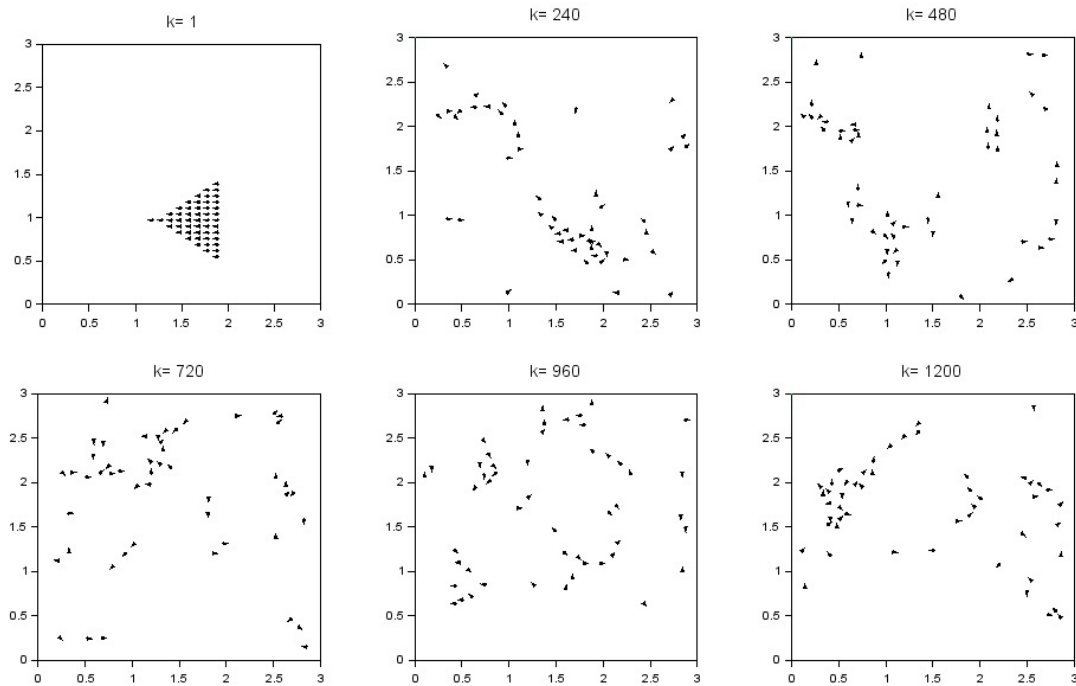


Figura 4.14: Enjambre de 50 robots con $r_r = 0.1$ y $r_a = 0.2$

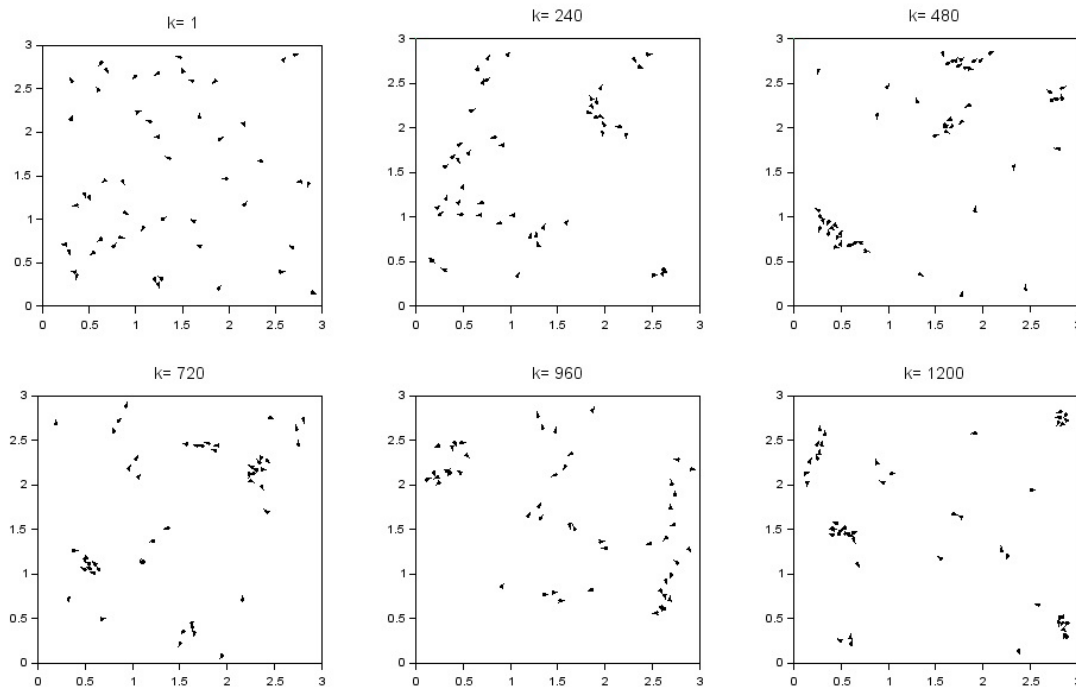


Figura 4.15: Enjambre de 50 robots con $r_r = 0.05$ y $r_a = 0.2$

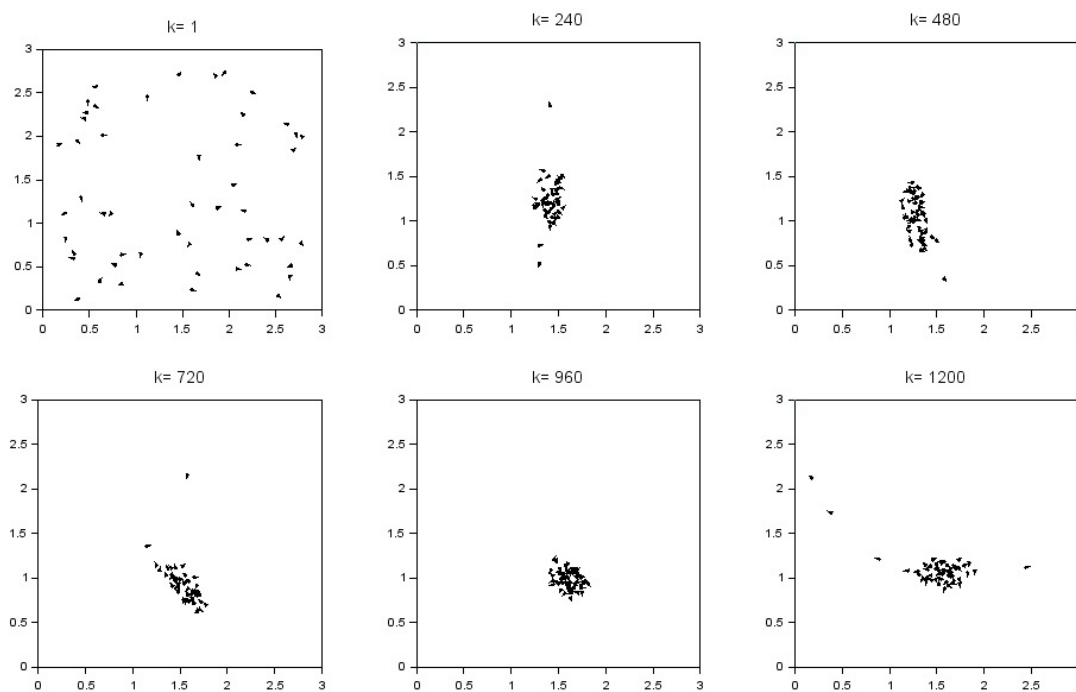


Figura 4.16: Enjambre de 50 robots con $r_r = 0.05$ y $r_a = 1.0$

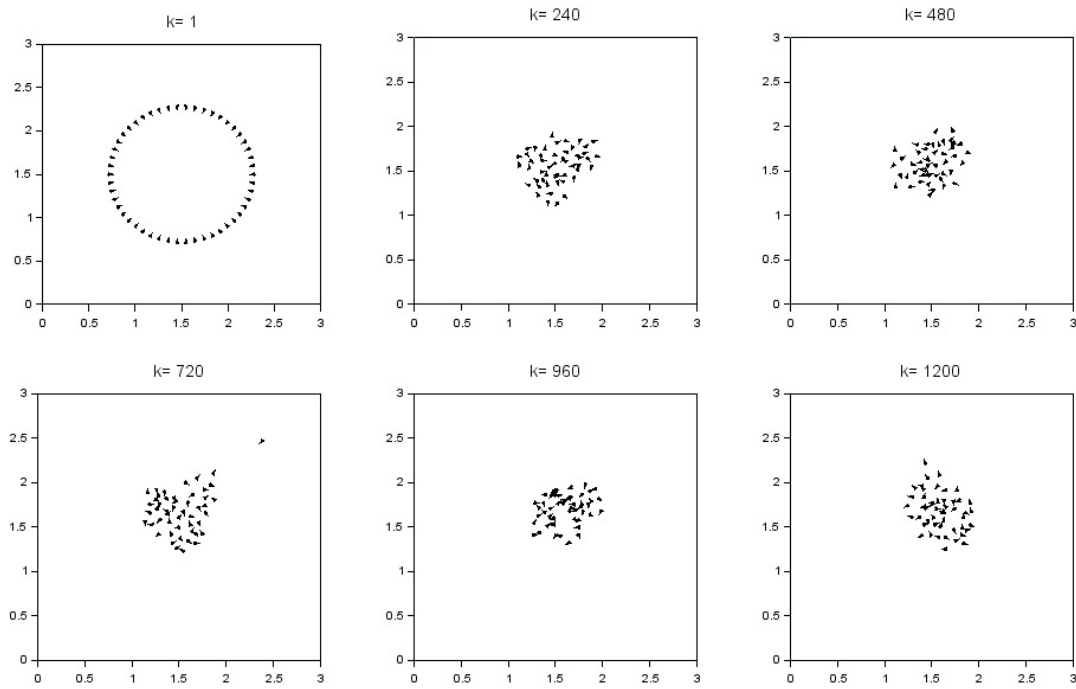


Figura 4.17: Enjambre de 50 robots con $r_r = 0.1$ y $r_a = 1.0$

CAPÍTULO 5

IMPLEMENTACIÓN EN ROBOTS

En este capítulo se detallan los resultados obtenidos al implementar las políticas de comportamiento en los prototipos experimentales diseñados. En primer lugar se da una descripción de la configuración de los experimentos, la posición y orientación inicial de los robots, junto con los parámetros explorados. Posteriormente se muestran secuencias de imágenes de los resultados obtenidos con diferentes parámetros.

5.1 DESCRIPCIÓN DE LOS EXPERIMENTOS

Los experimentos con robots reales se realizaron en un área cuadrada de 1.8 x 1.8 m, delimitada por paredes de 10 cm de altura. Los robots se colocan en una posición inicial similar a un círculo, con todos los robots orientados hacia el centro como se muestra en la Figura 5.1.

Se activa cada robot por medio de una comunicación bluetooth que se establece con la computadora. Esta comunicación también permite realizar cambios en los parámetros de repulsión y atracción, además de poder detener los robots. El vínculo se establece con un robot a la vez.

Considerando las capacidades sensoriales de los robots, presentadas en el Capítu-

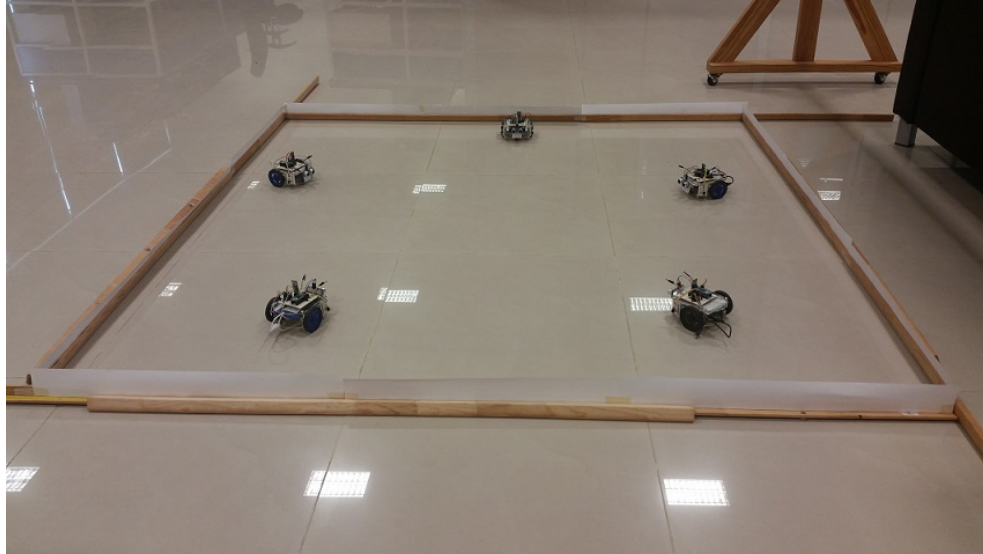


Figura 5.1: Posiciones y orientaciones iniciales en los experimentos

lo 3, se establecen las combinaciones de parámetros presentadas en la Tabla 5.1 para la realización de los experimentos.

Experimento	r_r (m)	r_o (m)	r_a (m)
1	0.05	0.1	0.2
2	0.05	0.1	0.27
3	0.05	0.1	0.35
4	0.1	0.15	0.2
5	0.1	0.15	0.27
6	0.1	0.15	0.35
7	0.15	0.15	0.2
8	0.15	0.15	0.27
9	0.15	0.15	0.35

Tabla 5.1: Radios de las zonas de comportamiento en los experimentos

5.2 RESULTADOS

A continuación se muestran secuencias de imágenes correspondientes a cada experimento, con un intervalo de 20 segundos entre cada imagen de la secuencia.

En las Figuras 5.2, 5.3 y 5.4 correspondientes a los experimentos con menor radio de repulsión, se presentan colisiones entre los robots. Con estos parámetros, los robots mantienen poco espacio entre ellos y al momento de llegar a un muro, comienzan a moverse a lo largo de este.

La principal causa de que se muevan a lo largo del muro, se debe a los sensores empleados que no pueden diferenciar entre obstáculos y robots. Esto provoca que los robots se muevan paralelo al muro, como si se tratara de otro robot que se mueve junto a ellos.

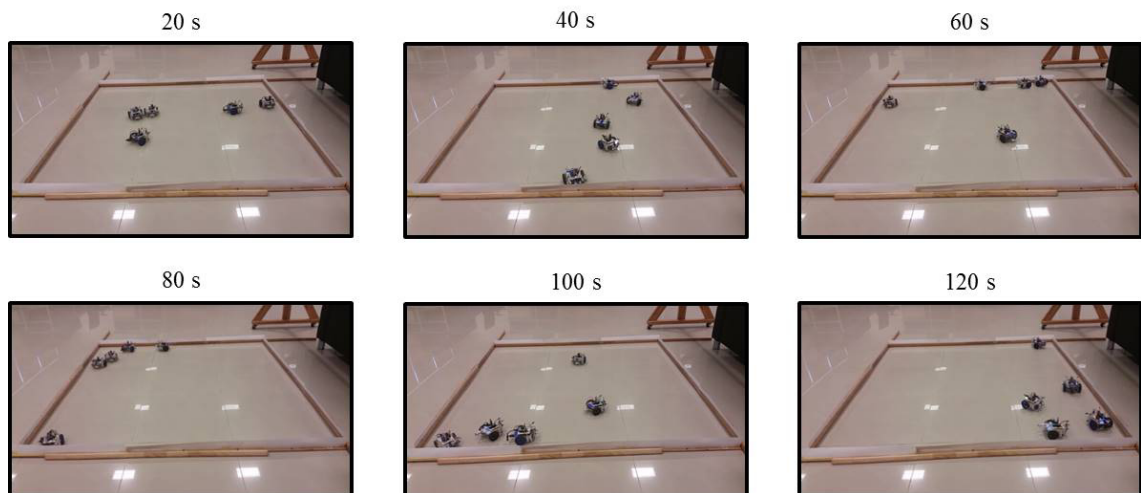


Figura 5.2: Experimento 1

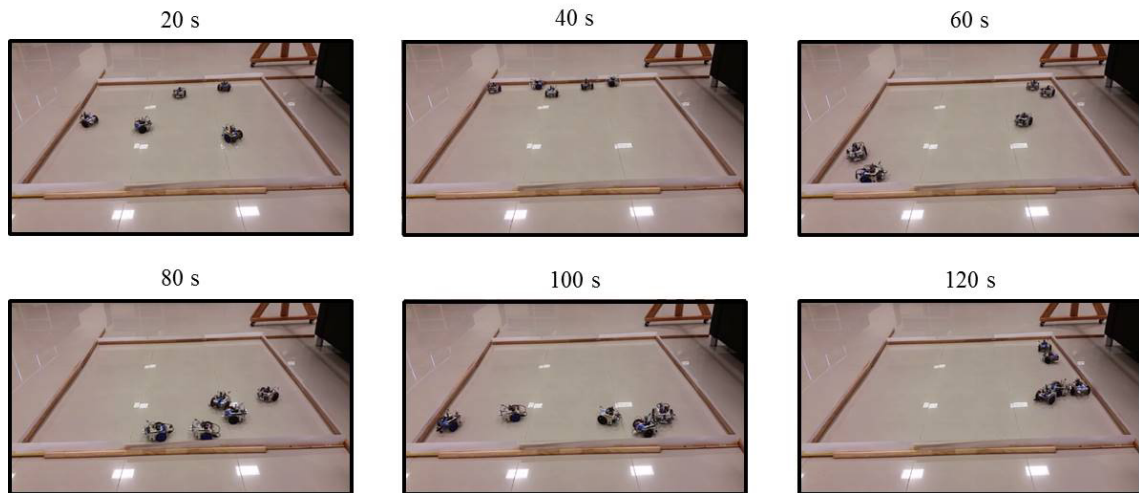


Figura 5.3: Experimento 2

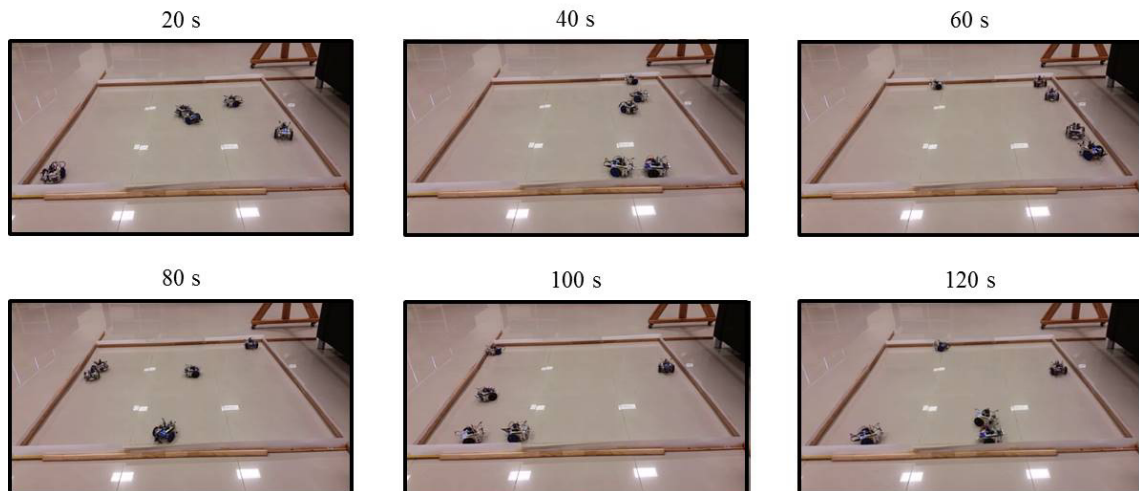


Figura 5.4: Experimento 3

Las Figuras 5.5, 5.6 y 5.7 corresponden a los experimentos con un radio de repulsión de 0.1 m. En estos casos, los robots mantienen más espacio entre ellos y se generan menos colisiones. De igual forma se mantienen más alejados de los muros y sus movimientos cubren más el área de prueba.

En algunos casos los robots se mueven también siguiendo los muros durante algunos momentos. Este comportamiento se observa principalmente cuando el radio de atracción es grande, porque los robots se atraen hacia otros o hacia los muros. Cuando el radio de atracción es pequeño, los robots tienden a esquivarse y moverse

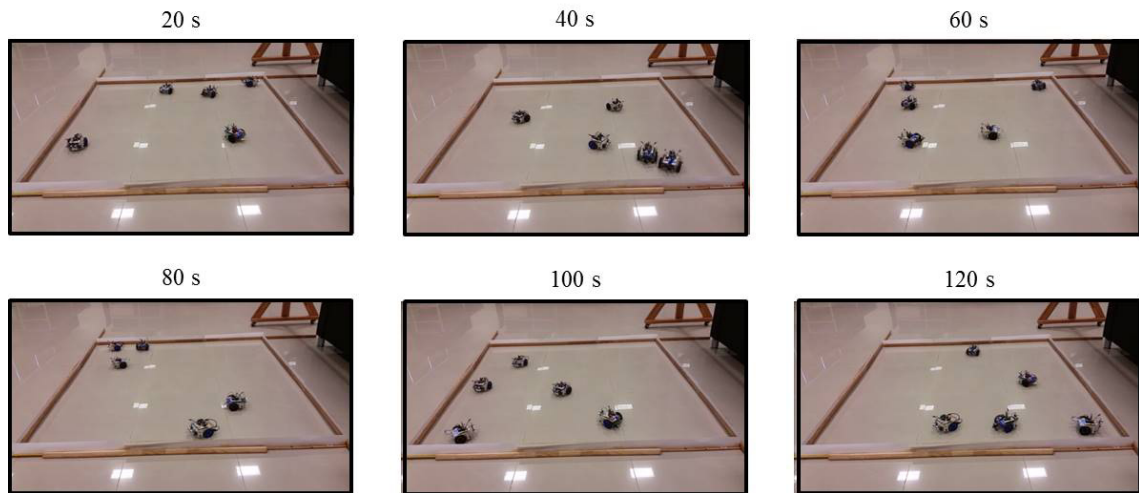


Figura 5.5: Experimento 4

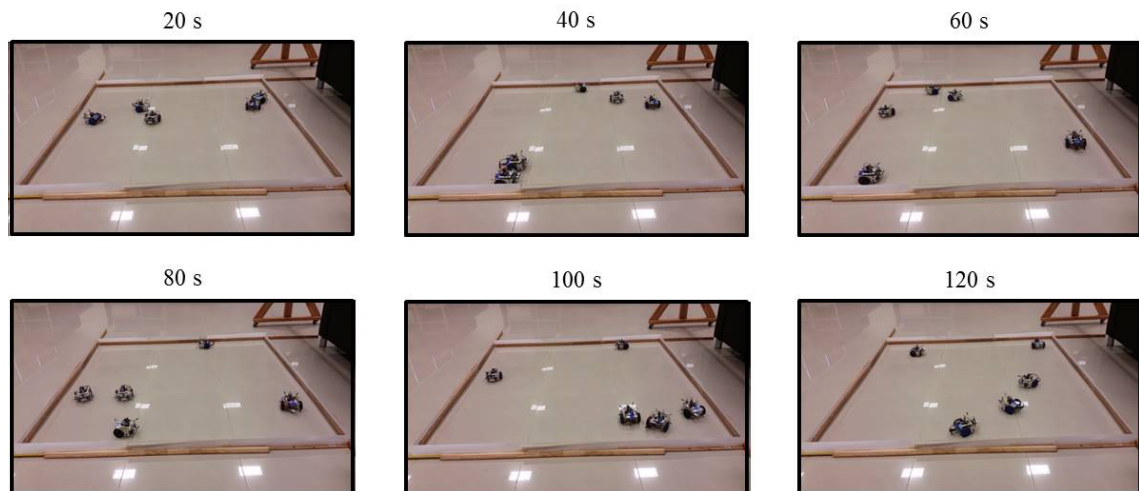


Figura 5.6: Experimento 5

por toda el área, en lugar de seguir los muros.

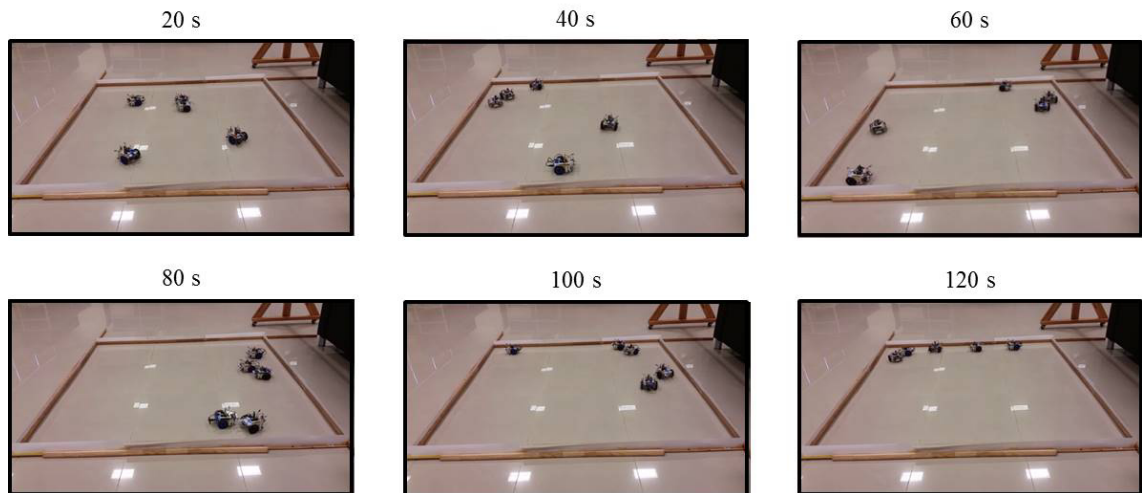


Figura 5.7: Experimento 6

Los experimentos con un parámetro de repulsión alto se muestran en las Figuras 5.8, 5.9 y 5.10. Como ya se mencionó, un radio de repulsión grande hace que los robots se mantengan separados, evitando colisiones y dispersándose en el área de prueba.

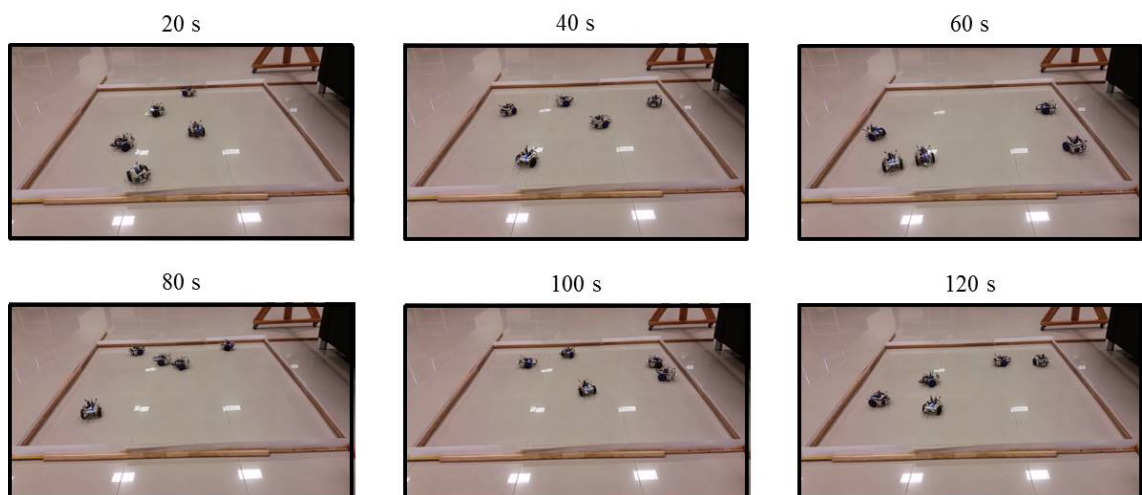


Figura 5.8: Experimento 7

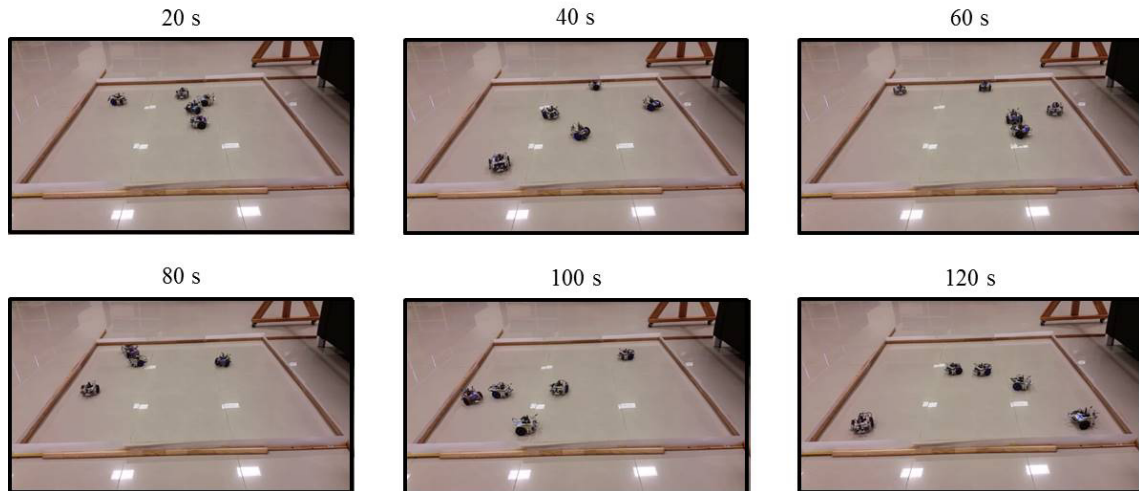


Figura 5.9: Experimento 8

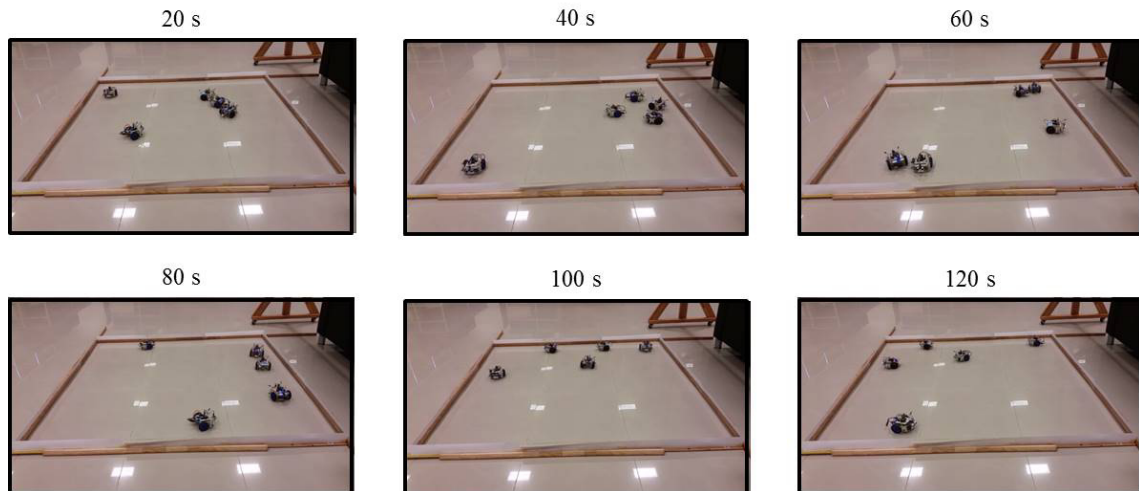


Figura 5.10: Experimento 9

Un comportamiento similar al obtenido en las simulaciones con pocos robots se aprecia en mayor medida cuando el parámetro de atracción tiene un valor medio. Durante algunos instantes de tiempo dos robots se mueven en la misma dirección, siguiendo uno a otro.

En la Figura 5.11 se muestra la trayectoria de 2 robots durante 20 segundos (00:01:40 - 00:02:00) del experimento 2. Durante este tiempo un robot sigue a otro cruzando el área de prueba hasta que se encuentran con otros robots y tienen que girar. De manera similar, la Figura 5.12 corresponde al movimiento de dos robots

durante 16 segundos (00:01:06 - 00:01:22) del experimento 8.

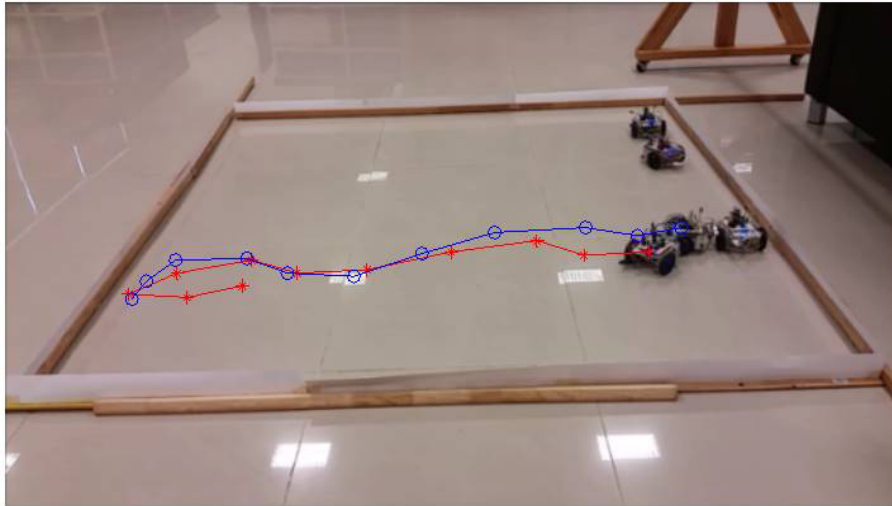


Figura 5.11: Trayectoria de 2 robots en el experimento 2

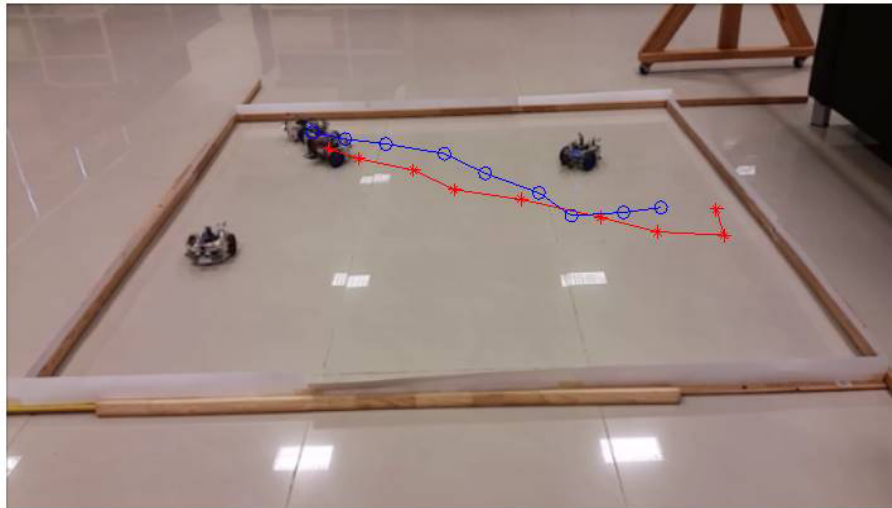


Figura 5.12: Trayectoria de 2 robots en el experimento 8

CAPÍTULO 6

CONCLUSIONES Y TRABAJO FUTURO

En esta investigación se adaptó el modelo de comportamiento de seres en grupo, propuesto por Iain Couzin, para su implementación en pequeños enjambres de robots que utilicen sólo la información local percibida por sus sensores.

Como la idea central detrás de la robótica de enjambre radica en la utilización de grandes cantidades de robots muy simples, las políticas de comportamiento se diseñan considerando condiciones mínimas de percepción y sin la necesidad de transmisión de información.

Con el fin de validar el modelo propuesto con diferentes cantidades de robots, se realizaron simulaciones con las políticas propuestas. Las simulaciones realizadas dieron como resultado diversos comportamientos que emergieron a partir de variaciones en los parámetros de repulsión y atracción que rigen a cada individuo. Estos comportamientos pueden tener aplicaciones en la realización de diversas tareas con sólo realizar cambios en ciertos parámetros.

Uno de los comportamientos que surgieron en las simulaciones con grupos pequeños de robots, es que se orientan. Sin contar con una política de orientación, la repulsión y atracción provoca que grupos de hasta 10 robots se muevan en la misma dirección durante algún tiempo. Este comportamiento no surge en enjambres más grandes, donde simplemente se atraen y se repelen.

Se implementaron las políticas de comportamiento en un sistema embebido para realizar pruebas reales en un enjambre de 5 robots. Las pruebas con estos robots demostraron que es posible implementar un modelo que no dependa de información global del enjambre para su operación. Los sensores utilizados en los robots no permiten distinguir entre los robots y la barrera que rodea el área. Debido a esto, el comportamiento que surgió en los experimentos con estos robots no corresponde exactamente a lo observado en las simulaciones.

Debido a limitaciones en cuanto a los sensores, es difícil obtener los mismos resultados vistos en las simulaciones, aunque es posible apreciar ciertos comportamiento que corresponden a lo observado en las simulaciones durante algunos instantes de tiempo.

Con lo presentado en este trabajo, quedan abiertos ciertos temas que pueden explorarse a mayor detalle en futuras investigaciones. Uno de estos temas es la utilización de un modelo dinámico para que las simulaciones sean más cercanas a la realidad. Otro aspecto que puede mejorar el modelo propuesto es la utilización de comunicación local entre los robots, con el fin de que los individuos con información sobre el entorno o sobre algún objetivo puedan transmitirla a otros miembros. Finalmente, se podrían mejorar los sensores del robot, para tener mayor área de percepción y poder diferenciar entre robots y obstáculos.

APÉNDICE A

REPORTE TÉCNICO DE LOS ROBOTS

A.1 DISEÑO

La estructura del robot está formada por dos placas de acrílico con un diámetro de 12.5 cm. Entre las dos placas se encuentran los motores y la batería. Las ruedas son de goma para evitar el deslizamiento y tienen un diámetro de 6 cm. La placa de circuito impreso se ubica sobre la estructura. Una vista de la estructura mecánica puede verse en la Figura A.1.

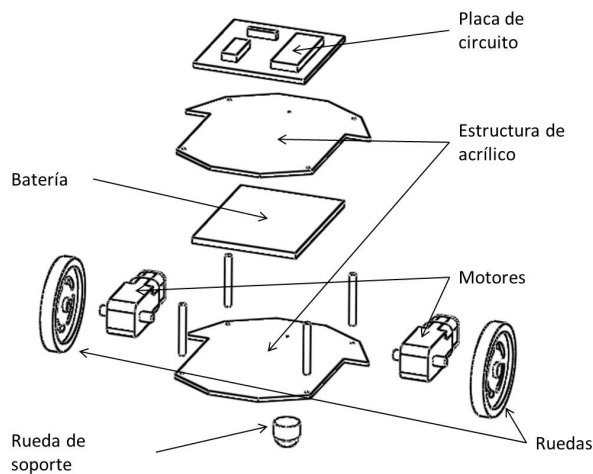


Figura A.1: Vista explosionada de la estructura del robot

A.2 HARDWARE

A.2.1 MICROCONTROLADOR

Para la parte de procesamiento del robot se utiliza una placa Arduino Micro, la cual incorpora un microcontrolador de 8 bits, comunicación USB y todos los elementos necesarios para su funcionamiento. La Tabla A.1 muestra un resumen de las características de esta placa.

Tabla A.1: Características del Arduino Micro

Microcontrolador	ATmega32u4
Voltaje de operación	5 V
Pines digitales I/O	20
Canales PWM	7
Canales de entrada analógica	12
Corriente DC por pin I/O	40 mA
Memoria Flash	32 KB
SRAM	2.5 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

A.2.2 SENSORES Y ACTUADORES

Al frente del robot se ubica un sensor ultrasónico HC-SR04, cuyas características se muestran en la Tabla A.2.

A la derecha e izquierda del robot se utilizan dos sensores infrarrojos TCRT5000, con un rango aproximado de detección de 40 cm.

Tabla A.2: Características del sensor ultrasónico

Ángulo efectivo	<15 grados
Rango de distancia	2 cm - 400 cm
Resolución	0.3 cm

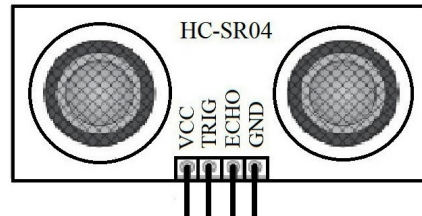


Figura A.2: Asignación de pines del sensor ultrasónico HC-SR04

Tabla A.3: Características de los sensores infrarrojos

Voltaje de polarización del emisor	1.5 V
Corriente del emisor	60 mA
Longitud de onda del emisor	950 nm
Distancia para operación pico	2.5 mm

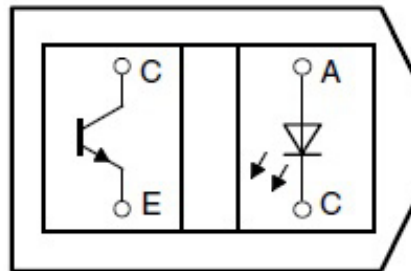


Figura A.3: Asignación de pines del sensor infrarrojo TCRT5000.

El robot utiliza dos motores de corriente directa con reducción de velocidad y se utiliza un circuito integrado SN754410 para controlarlos. La Tabla A.4 muestra las características de los motores empleados.

Tabla A.4: Características de los motores

Voltaje de operación	3 V - 12 V
Relación de reducción	120:1
Velocidad sin carga	170 rpm (3 V)
Consumo de corriente	70 mA (250 mA max.)
Par	800 gf/cm

A.2.3 ALIMENTACIÓN

Cada robot cuenta con una batería recargable Li-ion de 4.1 V y un módulo (Figura A.4) para elevar la tensión a 5V, con salida USB para alimentar la placa Arduino. Este módulo también se utiliza para cargar la batería, alimentándolo con 5V mediante el puerto micro USB.

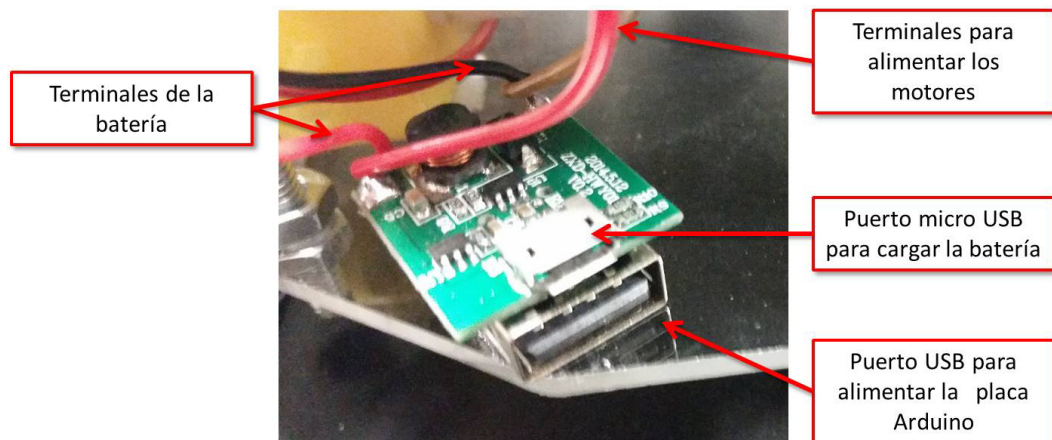


Figura A.4: Módulo convertidor DC-DC y cargador de batería.

La Figura A.5 muestra el diagrama esquemático del circuito impreso que controla cada robot.

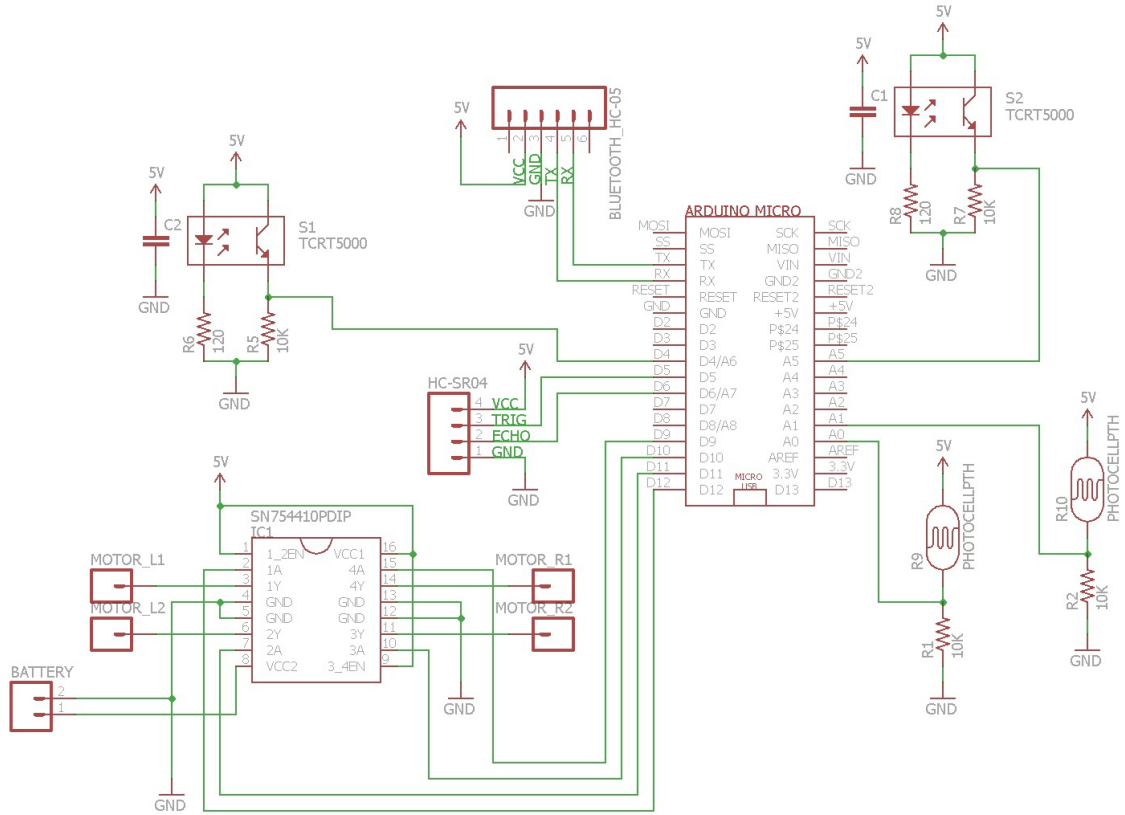


Figura A.5: Diagrama esquemático

A.3 SOFTWARE

La programación del microcontrolador se realizó en el entorno de desarrollo integrado de Arduino (Arduino IDE 1.6).

Para la comunicación bluetooth con la computadora, se utilizó la aplicación *Bluetooth Serial Terminal* para *Windows 8*, mientras que para la comunicación con un celular se utilizó la aplicación *Bluetooth Terminal* para *Android*.

A.4 FUNCIONAMIENTO

Para encender los robots es necesario conectar un cable USB - micro USB del módulo de la batería a la placa Arduino. Como los motores están alimentados directamente de la batería hay que conectar el cable rojo que se muestra en la Figura A.6.

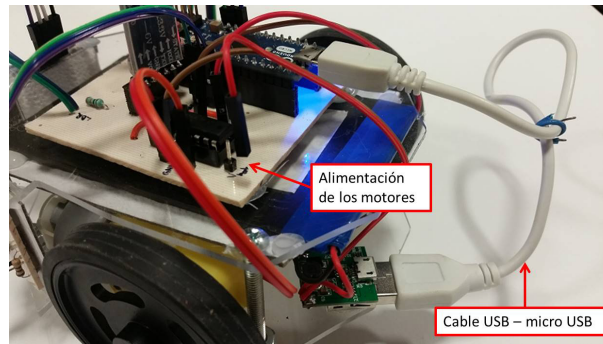


Figura A.6: Alimentación del arduino y los motores

Una vez encendido, el robot entrará en modo automático hasta que reciba un comando por el puerto serie. Los comandos utilizados se muestran en el código de Arduino anexo en el apéndice B.

APÉNDICE B

CÓDIGO EN ARDUINO

```
//*****  
// Efecto de los parámetros de repulsión, atracción y orientación  
// en un enjambre de robots  
// MCIE Mecatrónica  
// UANL, FIME  
// Mario Aguilera Ruiz  
//*****  
  
// Declaración de librerías usadas  
#include <NewPing.h>  
#include <SoftwareSerial.h>  
  
// Definición de pines I/O  
#define TRIGGER_PIN 5 // salida digital para sensor ultrasónico  
#define ECHO_PIN 6 // entrada digital para sensor ultrasónico  
int motorR1 = 12; // salida digital para motor  
int motorL1 = 10; // salida digital para motor  
int motorR2 = 11; // salida analógica para motor  
int motorL2 = 9; // salida analógica para motor
```

```
int ldrI = A0;          // entrada analógica para fotorresistencia izq.
int ldrD = A1;          // entrada analógica para fotorresistencia der.
int irD = A5;           // entrada analógica para infrarrojo der.
int irI = A6;           // entrada analógica para infrarrojo izq.
//función de inicialización para ultrasónico
NewPing sonar(TRIGGER_PIN, ECHO_PIN, 400);

//Definición de variables y constantes
int val;                // valor leído por puerto serie.
int distance;           // distancia calculada por ultrasonido
int blue = 0;           // bandera para
int ldrI_val, ldrD_val; // variables para lectura de fotorresistencias
int irI_val, irD_val;   // variables para lecturas de infrarrojos
int dR=5;               // diastancia de repulsión
int dO=10;              // distancia de orientación
int dA=20;              // distancia de atracción
int Q1[]={0, 1};       // vector en la dirección del sensor 1
int Q2[]={-1, 0};      // vector en la dirección del sensor 2
int Q3[]={1, 0};       // vector en la dirección del sensor 3
int q1A,q2A,q3A;       // elementos detectados por sensor en zona de atracción
int q1R,q2R,q3R;       // elementos detectados por sensor en zona de repulsión
int detR;               // elementos detectados en zona de Repulsión
int detO;               // elementos detectados en zona de Orientación
int detA;               // elementos detectados en zona de Atracción
int DQ1;                // distancia medida por el sensor 1
int DQ2;                // distancia medida por el sensor 2
int DQ3;                // distancia medida por el sensor 3
int ddx,ddy;           // componentes de la diección deseada
int dd,dd2;            // dirección deseada
```

```
void setup() {
  Serial.begin(9600);          // iniciar comunicación serie
  Serial1.begin(9600);
  pinMode(motorL1, OUTPUT); // configuración de pines de los motores
  pinMode(motorR1, OUTPUT);
  pinMode(motorL2, OUTPUT);
  pinMode(motorR2, OUTPUT);
}

void loop() {
  if (Serial.available()){    // verificar si hay comandos
    blue = '0';
  }
  if (Serial1.available()){
    blue = '0';
  }
  while (blue == '0') {      // leer comando
    if (Serial1.available()) {
      val = Serial1.read();
    }
    if (Serial.available()) {
      val = Serial.read();
    }

    if (val == 'F') {
      Forward();
    }
    if (val == 'B') {
      Backward();
    }
  }
}
```

```
    if (val == 'L') {
        TurnLeft(5);
        Stop();
    }
    if (val == 'R') {
        TurnRight(5);
        Stop();
    }
    if (val == 'S') {
        Stop();
    }
    if (val == 'X') {
        blue = '1';
    }
    if (val == '1') {
        Stop();
        dR=5;
        d0=10;
        dA=20;
    }
}

// Lectura de los sensores
ldrI_val = analogRead(ldrI); // lectura de fotorresistencias
ldrD_val = analogRead(ldrD);

int irI_val1 = analogRead(irI); // lectura 1 de infrarrojos
int irD_val1 = analogRead(irD);
int irI_val2 = analogRead(irI); // lectura 2 de infrarrojos
int irD_val2 = analogRead(irD);
```

```
int irI_val3 = analogRead(irI);    // lectura 3 de infrarrojos
int irD_val3 = analogRead(irD);

irI_val=(irI_val1 + irI_val2 + irI_val3)/3; // promedio de las 3 lecturas
irD_val=(irD_val1 + irD_val2 + irD_val3)/3;

// Conversión ADC a cm de los sensores infrarrojos
DQ2 = 126.43*pow(irI_val,-0.692); //prom    // cálculo de dist. en cm
DQ3 = 126.43*pow(irD_val,-0.692); //prom

DQ1 = Ultrasonico();    // lectura de ultrasónico

// iniciar variables en cero
detR=0; det0=0; detA=0;
q1R=0; q2R=0; q3R=0; q1A=0; q2A=0; q3A=0;

// Repulsión
if (DQ1<=dR || DQ2<=dR || DQ3<=dR){
    detR=1;
    if (DQ1<=dR){
        q1R=1;
    }
    if (DQ2<=dR){
        q2R=1;
    }
    if (DQ3<=dR){
        q3R=1;
    }
    // componentes de la dirección deseada de repulsión
    ddx= -(Q1[0]*q1R + Q2[0]*q2R + Q3[0]*q3R);
```



```
    ddy= -(Q1[1]*q1R + Q2[1]*q2R + Q3[1]*q3R);
}

// Orientación
if (detR==0){
    if (DQ1<=d0){
        det0=1;
    }
    if (DQ2<=d0){
        det0=1;
    }
    if (DQ3<=d0){
        det0=1;
    }
}

if (detR==0 && det0==1 && detA==0){
    // componentes de la dirección deseada de orientación
    // mantener la misma dirección
    ddx=0;
    ddy=1;
}

// Atracción
if (detR==0 && det0==0){
    if (DQ1<=dA){
        detA=1;
        q1A=1;
    }
    if (DQ2<=dA){
        detA=1;
    }
}
```

```
    q2A=1;
  }
  if (DQ3<=dA){
    detA=1;
    q3A=1;
  }
  if (detA==1){
    // componentes de la dirrección deseada de atracción
    ddx= Q1[0]*q1A + Q2[0]*q2A + Q3[0]*q3A;
    ddy= Q1[1]*q1A + Q2[1]*q2A + Q3[1]*q3A;
  }
}

//Nada detectado
  if (detR==0 && detO==0 && detA==0){
    // si no se detecta nada, mantener dirrección
    ddx=0;
    ddy=1;
  }

float ddrad=atan2(ddy,ddx); // Cálculo de la dirección deseada en radianes
dd=ddrad*180/3.14159;      // conversión de radianes a grados

// Conversión al sistema de referencia del robot
if (dd>-90){
  dd2=dd-90;
}
if (dd<=-90){
  dd2=dd+270;
}
```

```
if (dd2>=90 && dd2<180){
    TurnLeft(90);
    Forward();
}
if(dd2<90 && dd2>0){
    TurnLeft(dd2);
    Forward();
}

if (dd2<=-90 && dd2>-180){
    TurnRight(90);
    Forward();
}
if (dd2>-90 && dd2<0){
    dd2=abs(dd2);
    TurnRight(dd2);
    Forward();
}

if (dd2==0){
    Forward();
}

if (dd2>=178 && dd2<=182){
    if (irI_val>irD_val){
        TurnRight(90);
        Forward();
    }
    if (irI_val<=irD_val){
```

```
        TurnLeft(90);
        Forward();
    }
}

delay(50);
Stop();
//Mostrar(); // función para mostrar datos por puerto serie
} //End of void loop

// Funciones de movimiento
void Forward(){
    digitalWrite(motorL1, 0);
    digitalWrite(motorR1, 0);
    analogWrite(motorL2, 180);
    analogWrite(motorR2, 180);
}

void Backward(){
    digitalWrite(motorL1, 1);
    digitalWrite(motorR1, 1);
    digitalWrite(motorL2, 0);
    digitalWrite(motorR2, 0);
}

void TurnLeft(int grados){
    digitalWrite(motorL1, 0);
    digitalWrite(motorR1, 1);
    digitalWrite(motorL2, 1);
    digitalWrite(motorR2, 0);
    int time= 3.39*grados;
    delay(time);
}
```

```
}  
  
void TurnRight(int grados){  
    digitalWrite(motorL1, 1);  
    digitalWrite(motorR1, 0);  
    digitalWrite(motorL2, 0);  
    digitalWrite(motorR2, 1);  
    int time= 3.39*grados;  
    delay(time);  
}  
  
void Stop(){  
    digitalWrite(motorL1, 1);  
    digitalWrite(motorR1, 1);  
    digitalWrite(motorL2, 1);  
    digitalWrite(motorR2, 1);  
}  
  
int Ultrasonico(){  
    int distance1=sonar.ping_cm();  
    delay(5);  
    int distance2=sonar.ping_cm();  
    delay(5);  
    int distance3=sonar.ping_cm();  
    int prom=(distance1+distance2+distance3)/3; // promedio de 3 lecturas  
    return prom;  
}
```

BIBLIOGRAFÍA

- [1] A. Kolpas, M. Busch, H. Li, I. D. Couzin, L. Petzold, and J. Moehlis, “How the spatial position of individuals affects their influence on swarms: a numerical comparison of two popular swarm dynamics models,” *PloS one*, vol. 8, no. 3, pp. e58525–e58525, 2012.
- [2] I. D. Couzin and N. R. Franks, “Self-organized lane formation and optimized traffic flow in army ants,” *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 270, no. 1511, pp. 139–146, 2003.
- [3] S. V. Viscido, M. Miller, and D. S. Wethey, “The dilemma of the selfish herd: the search for a realistic movement rule,” *Journal of Theoretical Biology*, vol. 217, no. 2, pp. 183–194, 2002.
- [4] C. Blum and X. Li, “Swarm intelligence in optimization,” in *Swarm Intelligence* (C. Blum and D. Merkle, eds.), Natural Computing Series, pp. 43–85, Springer Berlin Heidelberg, 2008.
- [5] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.
- [6] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, “Collective memory and spatial sorting in animal groups,” *Journal of theoretical biology*, vol. 218, no. 1, pp. 1–11, 2002.

-
- [7] D. W. Payton, M. J. Daily, B. Hoff, M. D. Howard, and C. L. Lee, “Pheromone robotics,” in *Intelligent Systems and Smart Manufacturing*, pp. 67–75, International Society for Optics and Photonics, 2001.
- [8] D. Payton, R. Estkowski, and M. Howard, “Pheromone robotics and the logic of virtual pheromones,” in *Swarm Robotics*, pp. 45–57, Springer, 2005.
- [9] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo, “Evolving aggregation behaviors in a swarm of robots,” in *Advances in artificial life*, pp. 865–874, Springer, 2003.
- [10] M. Dorigo, E. Tuci, R. Groß, V. Trianni, T. H. Labella, S. Nouyan, C. Ampatzis, J.-L. Deneubourg, G. Baldassarre, S. Nolfi, *et al.*, “The swarm-bots project,” in *Swarm Robotics*, pp. 31–44, Springer, 2005.
- [11] B. T. Fine and D. A. Shell, “Unifying microscopic flocking motion models for virtual, robotic, and biological flock members,” *Autonomous Robots*, vol. 35, no. 2-3, pp. 195–219, 2013.
- [12] C. Moeslinger, T. Schmickl, and K. Crailsheim, “A minimalist flocking algorithm for swarm robots,” in *Advances in Artificial Life. Darwin Meets von Neumann*, pp. 375–382, Springer, 2011.
- [13] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.
- [14] E. Şahin, “Swarm robotics: From sources of inspiration to domains of application,” in *Swarm robotics*, pp. 10–20, Springer, 2005.
- [15] Y. Tan and Z.-y. Zheng, “Research advance in swarm robotics,” *Defence Technology*, vol. 9, no. 1, pp. 18–39, 2013.
- [16] D. P. Stormont, “Autonomous rescue robot swarms for first responders,” in *Computational Intelligence for Homeland Security and Personal Safety, 2005*.

- CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 151–157, IEEE, 2005.
- [17] F. Aznar, M. Sempere, M. Pujol, R. Rizo, and M. Pujol, “Modelling oil-spill detection with swarm drones,” in *Abstract and Applied Analysis*, vol. 2014, Hindawi Publishing Corporation, 2014.
- [18] Y. Altshuler, V. Yanovsky, I. A. Wagner, and A. M. Bruckstein, “Efficient cooperative search of smart targets using uav swarms,” *Robotica*, vol. 26, no. 04, pp. 551–557, 2008.
- [19] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, “Control of multiple uavs for persistent surveillance: algorithm and flight test results,” *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 5, pp. 1236–1251, 2012.
- [20] Y. C. Tan and B. E. Bishop, “Evaluation of robot swarm control methods for underwater mine countermeasures,” in *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, pp. 294–298, IEEE, 2004.
- [21] E. Vassev, R. Sterritt, C. Rouff, and M. Hinchey, “Swarm technology at nasa: building resilient systems,” *IT Professional*, no. 2, pp. 36–42, 2012.
- [22] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.
- [23] A. Baturone, *Robótica: manipuladores y robots móviles*. Marcombo, 2005.
- [24] R. Kamal, *Embedded systems 2E*. Tata McGraw-Hill Education, 2008.
- [25] Scilab Enterprises, *Scilab: Free and Open Source software for numerical computation*. Scilab Enterprises, Orsay, France, 2015.
- [26] Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis, *Arduino: open-source project official website*. Arduino, 2015.