



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TEMA: CONCEPTO Y PROTOTIPO DE UN ENJAMBRE DE
VEHÍCULOS AUTÓNOMOS ADECUADO PARA UN PROCESO
EVOLUTIVO INFINITO**

AUTOR:

SR. PRÓCEL MEDINA, GALO SEBASTIÁN

**DIRECTOR: ING. ARCENTALES VITERI, ANDRÉS
RICARDO PH.D.**

SANGOLQUÍ

2016



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

CERTIFICACIÓN

Certifico que el trabajo de titulación: “**CONCEPTO Y PROTOTIPO DE UN ENJAMBRE DE VEHÍCULOS AUTÓNOMOS ADECUADO PARA UN PROCESO EVOLUTIVO INFINITO**”, realizado por el señor **GALO SEBASTIÁN PRÓCEL MEDINA**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **GALO SEBASTIÁN PRÓCEL MEDINA** para que lo sustente públicamente.

Sangolquí, Septiembre de 2016

Ing. Andrés Ricardo Arcentales Viteri PhD.

DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

AUTORÍA DE RESPONSABILIDAD

Yo, **GALO SEBASTIÁN PRÓCEL MEDINA** con cédula de identidad N°: 1718163932, declaro que este trabajo de titulación “**CONCEPTO Y PROTOTIPO DE UN ENJAMBRE DE VEHÍCULOS AUTÓNOMOS ADECUADO PARA UN PROCESO EVOLUTIVO INFINITO**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, Septiembre de 2016

Sr. Galo Sebastián Prócel Medina

C.C. 1718163932



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

AUTORIZACIÓN

Yo, **GALO SEBASTIÁN PRÓCEL MEDINA**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca virtual de la institución el presente trabajo de titulación “**CONCEPTO Y PROTOTIPO DE UN ENJAMBRE DE VEHÍCULOS AUTÓNOMOS ADECUADO PARA UN PROCESO EVOLUTIVO INFINITO**”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, Septiembre de 2016

Sr. Galo Sebastián Prócel Medina

C.C. 1718163932

DEDICATORIA

Quiero dedicar este proyecto de investigación a mi familia, por su constante apoyo y amor incondicional a lo largo de toda mi vida, porque ustedes me han formado y gracias a ustedes soy la persona que soy el día de hoy.

A mi padre Galo, por ser mi ejemplo, mi modelo a seguir y siempre estar a mi lado, incondicionalmente, pese a toda circunstancia.

A mi madre Margarita, por ser la mujer más importante en mi vida y quien siempre supo guiarme con amor, paciencia y firmeza.

A mis hermanos Alan y Estefany, por su admiración, que es mi fuente más pura de inspiración para ser cada día una mejor persona tanto personal como profesionalmente.

AGRADECIMIENTO

A Dios, por el regalo de la vida, y todas las bendiciones que derrama sobre mí.

A mi familia, por siempre estar presente, por guiar mis pasos, por corregir mis errores, por todo el amor que me profesan.

A todos mis amigos, quienes han estado en los buenos y malos momentos a mi lado, personas que han enriquecido mi vida de tantas maneras.

A la Universidad de las Fuerzas Armadas – ESPE, a todos los docentes que me han brindado sus conocimientos y sabiduría a lo largo de mi formación profesional y además supieron formarme como ser humano, y en especial al Dr. Andrés Arcentales, quien supo guiarme con paciencia y brindarme su conocimiento para la culminación de este proyecto.

A todo el equipo de la Unidad de Relaciones de Cooperación Interinstitucional – URCI por darme a conocer nuevas oportunidades y brindarme las mejores experiencias de mi vida.

A la Universidad de Ciencias Aplicadas de Brandemburgo y al Dr. Guido Kramann por abrirme las puertas y permitir el desarrollo de este proyecto.

ÍNDICE DE CONTENIDO

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDO	vi
RESUMEN	xii
ABSTRACT	xiii
CAPÍTULO I	1
INTRODUCCIÓN	1
1.1. Antecedentes.....	1
1.2. Justificación e importancia	2
1.3. Alcance del proyecto	3
1.4. Objetivos.....	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos.....	4
CAPÍTULO II.....	5
MARCO TEÓRICO.....	5
2.1. Introducción.....	5
2.2. Algoritmos evolutivos	5
2.2.1. Componentes de un algoritmo evolutivo.....	7
2.2.2. Aplicaciones de los algoritmos evolutivos	11
2.3. Enjambres robóticos	12
2.3.1. Inspiración natural.....	13

2.3.2. Características constitutivas	14
2.3.3. Métodos de diseño.....	17
2.3.4. Métodos de análisis	19
2.3.5. Comportamientos colectivos	21
2.4. Aplicaciones combinadas	23
2.5. Descripción del proyecto	27

CAPÍTULO III

29

DISEÑO E IMPLEMENTACIÓN DE HARDWARE

29

3.1. Introducción.....	29
3.2. Consideraciones de diseño	29
3.2.1. Comunicación:	30
3.2.2. Sensores y Actuadores:	30
3.2.3. Procesamiento:	30
3.2.4. Sistema de energización	31
3.2.5. Tamaño del enjambre	32
3.2.6. Entorno	32
3.3. Resumen de características del enjambre	32
3.4. Implementación del sistema de detección de colisiones	33
3.5. Hardware adicional.....	36
3.6. Diseño e implementación del entorno	38
3.7. Programación del agente	41

CAPÍTULO IV.....

43

IMPLEMENTACIÓN DEL SOFTWARE.....

43

4.1. Introducción.....	43
------------------------	----

4.2. Software de Simulación.....	43
4.2.1. Consideraciones físicas	43
4.2.2. Simulación del movimiento	45
4.2.3. Evaluación del enjambre	46
4.3. Software de Optimización	49
4.3.1. Diseño del algoritmo	49
4.3.2. Resumen de características.....	54
CAPÍTULO V	55
EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS	55
5.1. Introducción.....	55
5.2. Análisis de resultados de optimización	56
5.3. Análisis de resultados de simulación.....	58
5.4. Análisis de resultados de implementación.....	60
CAPÍTULO VI.....	64
CONCLUSIONES Y RECOMENDACIONES	64
6.1. Conclusiones.....	64
6.2. Recomendaciones	66
BIBLIOGRAFÍA	67
ANEXOS	70

ÍNDICE DE TABLAS

Tabla 1 Resumen de características ATMEGA32L.....	30
Tabla 2 Resumen de características del enjambre a implementarse	33
Tabla 3 Características del micro-motor FF-130SH	34
Tabla 4 Resumen de características del agente	38
Tabla 5 Costos de implementación	40
Tabla 6 Representación genética del comportamiento.....	50
Tabla 7 Resumen de parámetros del algoritmo.....	54
Tabla 8 Comportamientos más óptimos.....	56
Tabla 9 Comportamientos menos óptimos.....	59

ÍNDICE DE FIGURAS

Figura 1	Proceso de un algoritmo evolutivo	6
Figura 2	Terminología de los algoritmos genéticos	8
Figura 3	Progreso típico de un algoritmo evolutivo.....	9
Figura 4	Aplicaciones de algoritmos evolutivos	12
Figura 5	Kilobots, desarrollados por Wyss Institute.....	14
Figura 6	Comunicación indirecta con emisión de luz.....	15
Figura 7	Formación de patrones con Kilobots.....	15
Figura 8	S-BOT desarrollado por EPFL para navegación por terreno.....	16
Figura 9	Enjambre robótico trabajando en equipo.....	22
Figura 10	Navegación grupal en terreno.....	23
Figura 11	Diseño 3D de un S-BOT en el simulador.....	24
Figura 12	Feromonas visualizadas en tiempo real a través de realidad aumentada... 25	25
Figura 13	Reglas de movimiento optimizado para un mapeo distribuido	26
Figura 14	Entorno creado para limpieza dinámica.....	26
Figura 15	Secuencia de actividades para el desarrollo del proyecto.....	28
Figura 16	Micro-motores FF-130SH	33
Figura 17	Conexión directa de los micro-motores.....	34
Figura 18	Lecturas analógicas con motor bloqueado (1) y liberado (2).	35
Figura 19	Conexión de los micro-motores con resistencias pull-down.	35
Figura 20	Lecturas analógicas con pull-down, motor bloqueado (1) y liberado (2)..	36
Figura 21	Conexiones para el programador AVRISP mkII.	36
Figura 22	Ubicación de los motores con respecto al suelo.....	37
Figura 23	Agente. Derecha: Primera versión. Izquierda: Versión final.....	37
Figura 24	Ensamblaje 3D del entorno.....	39
Figura 25	Método de polarización.....	39
Figura 26	Sistema total implementado.....	41
Figura 27	Posibles interacciones en la simulación.....	45
Figura 28	Parámetro de evaluación.....	47
Figura 29	Posición inicial.....	47
Figura 30	Eficiencia del enjambre según número de agentes.....	48

Figura 31 Evaluación en función del tiempo.	49
Figura 32 Distancia total según el número de simulaciones promediadas.....	51
Figura 33 Histograma función de selección.....	52
Figura 34 Genes más aptos.	56
Figura 35 Resultado de la ejecución de un algoritmo.	57
Figura 36 Convergencia del algoritmo.....	58
Figura 37 Genes menos aptos.	58
Figura 38 Comparación de resultados obtenidos en la simulación.	59
Figura 39 Posición inicial del enjambre.....	60
Figura 40 Comparación de los resultados obtenidos en la implementación física. ...	60
Figura 41 Comparación entre simulación e implementación con el gen más apto. ...	61
Figura 42 Comparación entre simulación e implementación con el gen aleatorio. ...	62
Figura 43 Comparación entre simulación e implementación con el gen menos apto.	62

RESUMEN

El proyecto titulado “Concepto y Prototipo de un Enjambre de Vehículos Autónomos Adecuado para un Proceso Evolutivo Infinito”, propuesto por la Universidad de Ciencias Aplicadas de Brandeburgo, tiene como objetivo diseñar e implementar un grupo de robots simples y de bajo costo para que en su conjunto desarrollen comportamientos complejos, conocido como inteligencia de enjambre, y resuelvan una tarea específica. Dicho comportamiento será optimizado fuera de línea, utilizando un algoritmo evolutivo. El proceso completo podrá ser simulado mediante un software desarrollado en su totalidad y enfocado a la tarea seleccionada. Como primer punto, se procede a la implementación física del sistema (hardware). Se construyen los agentes y el entorno de pruebas. Se programa un código general para que todos agentes puedan implementar un comportamiento específico, y así evaluar su desempeño en la vida real. Toda la implementación física se basa en la premisa del agente funcional más simple posible. En base al diseño del enjambre y al comportamiento físico de cada agente, se diseñó un software de simulación. Se toma en consideración todas las interacciones físicas posibles para generar una simulación acorde a la realidad. Es necesario obtener parámetros de desempeño a través de la simulación por lo que se programa una interfaz gráfica completa que despliega toda la información requerida. Una vez desarrollado el hardware y software del proyecto, se programa un algoritmo evolutivo que optimiza el comportamiento del enjambre para resolver la tarea propuesta. Este algoritmo utiliza la simulación del enjambre para evaluar los posibles comportamientos y retornar el más óptimo entre ellos. Para validar el proceso realizado, dicho comportamiento es implementado y evaluado en el hardware y es comparado con los resultados de la simulación, validando y concluyendo el proceso.

PALABRAS CLAVES:

- ENJAMBRE
- ROBOTICO
- EVOLUTIVO
- SIMULADOR
- ALGORITMO

ABSTRACT

The project entitled "Concept and Prototype of an Autonomous Swarm Vehicle Suitable for an Infinite Evolutionary Process" proposed by the Brandenburg University of Applied Sciences, aims to design and implement a group of simple and inexpensive robots which together can develop elaborate behaviors, known as swarm intelligence, to solve a specific task. Such behavior is optimized offline, using an evolutionary algorithm. The whole process can be simulated using the developed software focused on the selected task. The first phase of the project is to work on the physical part of the system (hardware). Agents and the test environment are implemented. A generic code is programmed so the agents can execute a specific behavior and evaluate its performance in real life. All the physical implementation is based on the premise of the simplest possible agent. In the second phase is designed and programmed the simulation software for the swarm. It takes into account all possible physical interactions to generate a simulation according to the reality. It is necessary to obtain performance parameters through simulation so it is designed a complete graphical interface that displays all the required information. Once the hardware and software of the project are complete, an evolutionary algorithm is programmed. This algorithm optimizes the behavior of the swarm to solve the proposed task. This algorithm uses the swarm simulation to evaluate behaviors and returns the optimum in between the explored behaviors. Finally, this behavior is implemented and evaluated in the hardware and it is compared with simulation results to validate and conclude the process.

KEYWORDS:

- SWARM
- ROBOTICS
- EVOLUTIONARY
- SIMULATOR
- ALGORITHM

CAPÍTULO I

INTRODUCCIÓN

Como parte del plan de fortalecimiento e internacionalización de la Universidad de las Fuerzas Armadas ESPE, se genera un convenio con la Universidad de Ciencias Aplicadas de Brandeburgo para desarrollar proyectos de investigación en colaboración mutua. Dentro de este marco, es propuesto el tema de investigación aquí presentado, para ser desarrollado como trabajo de investigación para la obtención del título de ingeniero en electrónica, automatización y control.

1.1. Antecedentes

La robótica de enjambre es un enfoque relativamente nuevo de sistemas multirobóticos, el cual consiste en una gran cantidad de robots simples, llamados agentes, que al interactuar de forma conjunta, generan comportamientos complejos o elaborados (Beni, 2004). Este nuevo enfoque está inspirado por sistemas naturales como colonias de hormigas, enjambres de abejas, escuelas de peces, entre otros (Dudek, Jenkin, Milios, & Wilkes, 1993). De forma genérica, esta interacción es conocida como inteligencia de enjambre.

La inteligencia de enjambre en la naturaleza es utilizada en tareas como construcción de panales, búsqueda de objetivos, caza grupal, movilización, entre otras, existiendo un alto grado de efectividad. Por este motivo, las investigaciones actuales se han enfocado en replicar la eficiencia de dichos comportamientos en la solución de problemas de la vida real. Esto, finalmente se traduce en aplicaciones de distinta índole, como médicas, agrícolas, de construcción, militares y más (Sahin, 2004).

Al aplicar los conceptos de inteligencia de enjambre en sistemas multirobóticos (robótica de enjambre), surgen varios aspectos a tomarse en cuenta, como por ejemplo el comportamiento de los agentes, formas de intercomunicación, diseño físico del enjambre, tamaño de la población, entre otros (Nolfi, Bongard, Husbands, & Floreano, 2016). Existen numerosas investigaciones que profundizan individualmente cada aspecto, entre ellas tenemos: optimización de arquitecturas de control (Balch, 2004),

importancia de la intercomunicación (Das, Behera, & Panigrahi, 2016), morfología según la aplicación y formas de aprendizaje (Xu & Chen, 2008).

En la temática del comportamiento del agente, la tendencia actual se dirige al aprendizaje de máquina (Barash, 1999). Para este propósito, las herramientas comúnmente utilizadas son técnicas de optimización estocástica, conocidas como algoritmos evolutivos. Las aplicaciones de los algoritmos evolutivos son varias y están presentes en áreas como diseño de circuitos, cálculo de estrategias de mercado, reconocimiento de patrones, acústica, ingeniería aeroespacial, y demás. De esta forma, se pretende obtener un comportamiento natural del enjambre artificial, siendo los mismos agentes los que aprenden a comportarse según la tarea a realizar o las condiciones del entorno (Beni, 2004).

1.2. Justificación e importancia

Existen tareas muy complejas para ser realizadas por un solo robot, por lo que es necesario en estos casos la utilización de más de uno. Varios robots simples pueden ofrecer una solución más económica y sencilla que un solo robot, grande y complejo. Además, los sistemas de múltiples robots son generalmente más flexibles y toleran mejor los fallos que un solo robot actuando por su cuenta. Tomando en cuenta estas ventajas de los sistemas multirobóticos y la efectividad del comportamiento de enjambres en la naturaleza, surge el estudio formal de lo que se conoce como enjambres robóticos.

Estos sistemas pueden ser utilizados en escenarios en donde las tareas son muy peligrosas o exigen mucho esfuerzo para un operador humano, o incluso cuando el tiempo de respuesta es crucial. Ejemplos de estas tareas son operaciones de búsqueda y rescate, exploración espacial, vigilancia, monitoreo, limpieza, mantenimiento, entre otros.

Por otro lado, la inteligencia artificial, como campo de estudio, despierta gran interés debido a sus múltiples aplicaciones que no solamente se limitan a la ingeniería, sino que resultan transversales en temáticas variadas que van desde negocios hasta medicina. La inteligencia de enjambre, siendo un enfoque especializado de la inteligencia artificial, tiene de igual forma altos niveles de efectividad frente a ciertas

tareas específicas como son recolección o construcción, y actualmente ya existen investigaciones que abordan estos temas.

Teniendo en cuenta el trasfondo científico que existe en torno a estos temas, se propone este proyecto de investigación, para obtener un primer acercamiento a una tendencia mundial actual de desarrollo tecnológico, que ofrece beneficios a corto y largo plazo. Con este trabajo se establecerá un punto de inicio para futuras investigaciones en nuestra universidad y país, además que se desarrollará un prototipo físico del sistema que puede ser utilizado para proyectos futuros.

1.3. Alcance del proyecto

Actualmente en la Universidad de Ciencias Aplicadas de Brandeburgo, existen trabajos previos relacionados a enjambres robóticos. Con base en estos prototipos, se rediseñará e implementará un sistema multirrobótico para cumplir con una tarea específica, pudiendo ser recolección, búsqueda, movilidad, entre otras, implementando algoritmos evolutivos para optimizar el cumplimiento de dicha tarea además de evidenciar características propias de un enjambre.

Para esto, el trabajo estará dividido en varias partes. En primer lugar, es necesario definir una tarea acorde a las capacidades del enjambre a construirse, teniendo en cuenta el tipo de agentes y los medios de interacción con el entorno.

Posteriormente, se procederá con el trabajo sobre hardware. Esto involucra la correcta selección de sensores y actuadores que permitan un desenvolvimiento óptimo del enjambre. Además, se construirá el escenario de trabajo adecuado a la tarea escogida y considerando la dimensión del sistema físico.

Luego, se trabajará en el software, primero definiendo el comportamiento de los agentes físicos, y segundo desarrollando un entorno de simulación para el enjambre.

Una vez definido el hardware y el software, se procederá a desarrollar un algoritmo evolutivo que optimice el comportamiento del enjambre en la tarea seleccionada, basándose en la simulación.

Finalmente se procederá con la implementación total del sistema, cargando a los agentes con el comportamiento optimizado, y ejecutando el proceso para resolver la tarea.

Con los resultados obtenidos será posible realizar un análisis de desempeño, comparación con las simulaciones y finalmente concluir el proyecto.

Esto servirá para establecer el trabajo futuro o las modificaciones que se crean necesarias.

1.4. Objetivos

1.4.1. Objetivo general

- Diseñar y construir un conjunto de vehículos autónomos que implementen algoritmos evolutivos para solucionar una tarea específica, basándose en inteligencia de enjambre.

1.4.2. Objetivos específicos

- Definir una tarea específica que deberá ser resuelta por el enjambre a crearse, teniendo en consideración la complejidad de la misma, así como sensores y actuadores necesarios, número de agentes requeridos y espacio físico.
- Diseñar e implementar el hardware del agente, para acoplarse a la tarea asignada, así como para interactuar con otros agentes, optimizando costo y tamaño del mismo.
- Diseñar un software de simulación apropiado para el enjambre creado y basado en la tarea escogida, para obtener resultados de desempeño y evaluación del sistema.
- Desarrollar un algoritmo evolutivo que optimice el comportamiento del enjambre mediante los parámetros de eficiencia seleccionados, basado en los resultados de desempeño obtenidos por la simulación.
- Implementar el comportamiento optimizado en el sistema real para analizar y comparar con los resultados simulados y establecer las conclusiones del proyecto.

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción

En este capítulo, se realizará un análisis detallado de toda la literatura revisada y se presentará la investigación actual y estado del arte de todos los componentes del presente trabajo de investigación. Con esto se podrá obtener inspiración para el posterior desarrollo del proyecto. Se presentará un resumen de los algoritmos evolutivos, con todas sus ventajas y desventajas dependiendo del uso de los mismos. Se presentará su definición y sus principios básicos. Además, se indicará como está compuesta la clasificación actual y los problemas típicos de aplicación. Posteriormente serán abarcados los más recientes avances y los planes a futuro para este campo de la inteligencia artificial.

Una vez presentada la temática de algoritmos evolutivos, se proseguirá con la definición y características constitutivas de los enjambres robóticos. Se detallará métodos de diseño y análisis. Serán presentados los comportamientos colectivos en los que se está desarrollando la investigación actual. Posteriormente se presentará los trabajos actuales que combinan ambas tendencias y principalmente en qué tipo de situaciones son aplicables ambos campos.

El capítulo finalizará con un detalle a breves rasgos de todo el trabajo realizado. De la tendencia actual se obtendrá la inspiración inicial y se seleccionará una tarea en específico para el enjambre a desarrollarse. Todo el contenido presentado en este capítulo servirá como base para fundamentar el desarrollo tecnológico en el presente proyecto.

Será presentado un diagrama esquemático del contenido del presente trabajo escrito para que el lector pueda guiarse a través del mismo.

2.2. Algoritmos evolutivos

En el campo de optimización matemática, un algoritmo evolutivo (EA) es una heurística de búsqueda que imita el proceso de selección natural. Esta heurística se utiliza de forma iterativa para generar soluciones útiles a problemas de optimización. (Lauer & Riedmiller, 2000). Los algoritmos genéticos pertenecen a la clase más grande

de los algoritmos evolutivos (EA), que generan soluciones a problemas de optimización utilizando técnicas inspiradas en la evolución natural, como la herencia, mutación, selección y de cruce.

En un algoritmo evolutivo, se evoluciona una población de soluciones candidatas (llamados individuos, criaturas, o fenotipos) a un problema de optimización hacia las mejores soluciones. Cada candidato tiene un conjunto de propiedades (sus cromosomas o genotipo) que puede ser mutado y alterado. Tradicionalmente, las soluciones están representados en código binario como cadenas de 0s y 1s, pero otras codificaciones son también posibles. (Xu & Chen, 2008)

La evolución comienza generalmente a partir de una población de individuos generados al azar, y es un proceso iterativo. Cada población presente en una iteración del algoritmo se llama generación. En cada generación, se evalúa la aptitud de cada individuo de la población; esta aptitud es por lo general el valor de la función objetivo en el problema de optimización a resolver. Los individuos más aptos son estocásticamente seleccionados de la población actual, y el genoma de cada individuo es modificado (recombinado y posiblemente mutado aleatoriamente) para formar una nueva generación. La nueva generación de soluciones candidatas se utiliza a continuación en la siguiente iteración del algoritmo. Comúnmente, el algoritmo termina cuando se ha producido un número máximo de generaciones, o se ha alcanzado un nivel de aptitud satisfactorio. El proceso simplificado se lo presenta en la Figura 1.

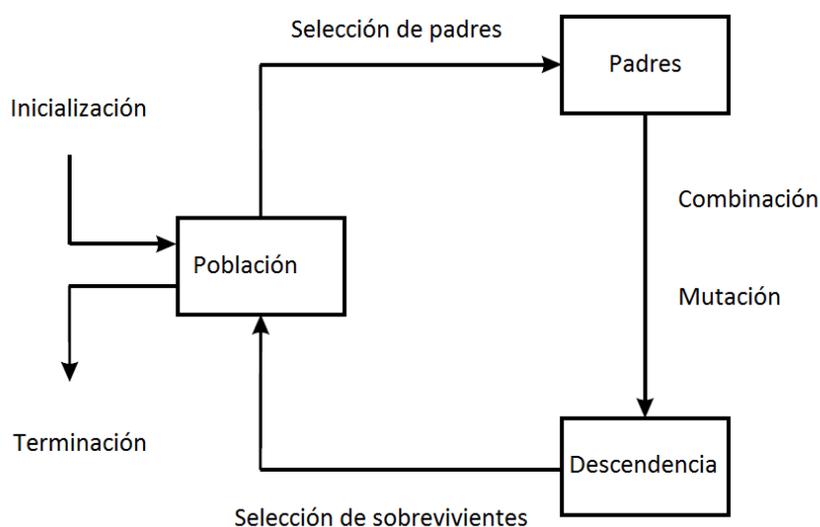


Figura 1 Proceso de un algoritmo evolutivo

Las diferentes clasificaciones de los algoritmos evolutivos poseen de forma general el mismo principio de funcionamiento. Las diferencias se encuentran en ciertos detalles técnicos constitutivos.

Para esta clasificación se utiliza la representación de un posible candidato a solución:

- Algoritmos genéticos: Los candidatos son representados como cadenas de caracteres de un alfabeto finito
- Estrategias de evolución: Los candidatos son representados como vectores de valores reales
- Programación evolutiva: Los candidatos son representados como máquinas de estado finitas
- Programación genética: Los candidatos son representados como estructuras de árbol

Todas estas representaciones tienen su origen en distintos puntos a través de la historia de la investigación de este campo. La utilización de cierto algoritmo en específico depende de la forma utilizada para codificar las posibles soluciones, o que esta codificación sea dada de forma natural.

2.2.1. Componentes de un algoritmo evolutivo

Un algoritmo evolutivo generalmente posee cierto número de componentes y operaciones necesarias para su desarrollo. Estos componentes son específicos y varían de un tipo de algoritmo evolutivo a otro. Estos son:

- Representación de los individuos
- Función de evaluación (aptitud)
- Población
- Mecanismo de selección de padres
- Operadores de variación, combinación y mutación
- Mecanismo de selección del superviviente

2.2.1.1. Representación de los individuos

La representación de los individuos establece un puente entre el algoritmo evolutivo y la vida real. De forma general, los posibles objetos o individuos que serán soluciones del problema son conocidos como fenotipos, y su codificación para ser optimizados dentro de un algoritmo evolutivo se llama genotipo. El primer paso para el diseño de un algoritmo se basa en mapear los fenotipos en genotipos. La convergencia de un algoritmo evolutivo depende estrechamente de la forma en que han sido mapeados los candidatos a soluciones (fenotipos). Todos estos conceptos pueden ser analizados en la Figura 2.

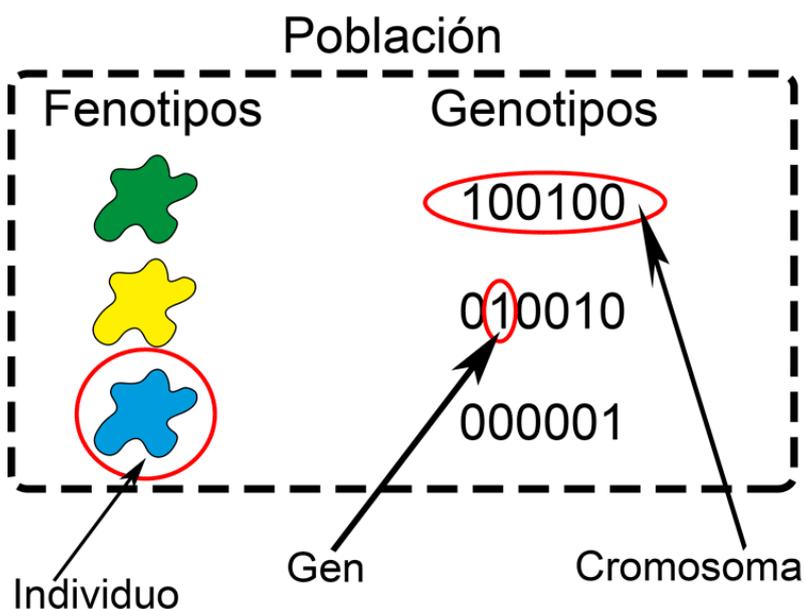


Figura 2 Terminología de los algoritmos genéticos

2.2.1.2. Función de aptitud

La función de aptitud o de evaluación representa los requerimientos de adaptación del proceso evolutivo. De forma precisa, define el concepto de “mejora” según sea el caso de aplicación del algoritmo evolutivo. El objetivo del algoritmo es maximizar la función de aptitud. Esta función de aptitud representa una medida de calidad del fenotipo para resolver el problema propuesto. En la Figura 3 puede observarse un ejemplo gráfico del funcionamiento de un algoritmo evolutivo, en donde el conjunto de individuos, o población (puntos rojos), evoluciona a lo largo del tiempo para ubicarse en los puntos más óptimos, o máximos, de la función de aptitud.

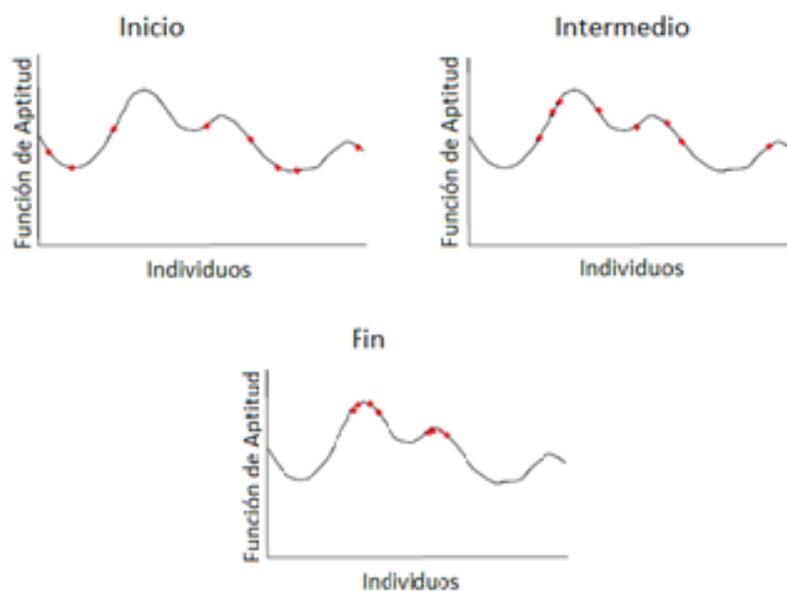


Figura 3 Progreso típico de un algoritmo evolutivo

Fuente: (Fonseca & Fleming, 1995)

2.2.1.3. Población

La población contiene todas las posibles soluciones. Se define como un grupo de genotipos, y representa a la unidad de evolución. Los miembros de la población pueden ser seleccionados, combinarse, mutar y/o morir, generando en cada iteración un nuevo conjunto o población. Las operaciones de selección y reemplazo afectan directamente a este nivel. En varias aplicaciones de algoritmos evolutivos, definir la población puede ser tan sencillo como especificar un número de miembros o genotipos, aunque existen otros enfoques donde existen parámetros adicionales como estructuras espaciales. Generalmente el tamaño de la población se mantiene constante a lo largo de la búsqueda evolutiva.

2.2.1.4. Selección

Los genotipos para ser padres son seleccionados de la población actual. La función de selección es la encargada de seleccionarlos. Según la teoría de la evolución de Darwin, los mejores especímenes deben sobrevivir y crear una nueva descendencia. Hay muchos métodos de cómo seleccionar los mejores genotipos, por ejemplo, la rueda de selección, la selección de Boltzmann, la selección del torneo, la selección de

rango, selección de estado estacionario y algunos otros (Fonseca & Fleming, 1995). De forma general, una función de selección será típicamente probabilística, es decir que mientras más apto sea el individuo (función de aptitud), tendrá mayores probabilidades de ser escogido para convertirse en padre y generar nueva descendencia. Igualmente, los individuos menos aptos tienen una pequeña probabilidad de ser escogidos, pero sigue siendo posible, esto con el propósito de no limitar el espacio de búsqueda o estancarse en mínimos locales.

2.2.1.5. Mutación

Es un operador genético utilizado para mantener la diversidad genética de una generación para la próxima. Es análogo a la mutación biológica. La mutación altera uno o más valores de los componentes en un gen de su estado inicial. Con la mutación, el gen puede cambiar por completo del gen anterior. De ahí que el algoritmo genético puede llegar a una mejor solución mediante el uso de mutación. La mutación se produce durante la evolución

Según diversos estudios, una probabilidad de mutación razonable debe ser baja. Si es demasiado alta, la búsqueda se convertirá en una búsqueda al azar primitiva (Barash, 1999).

2.2.1.6. Combinación

Es un operador genético utilizado para variar la programación de uno o más genes de una generación a la siguiente. Es análogo a la reproducción y cruce biológico, sobre la cual se basan los algoritmos evolutivos. La combinación es un proceso de tomar secciones genéticas de varios padres y producir un solo niño de ellos. Hay métodos para la selección de los genes a combinarse.

2.2.1.7. Reemplazo

Al igual que las funciones de selección, la función de reemplazo selecciona el individuo menos apto de la población y lo elimina. Este proceso puede ser tan sencillo como escoger en cada iteración el o los individuos menos aptos para borrarlos, o puede

poseer igualmente una función probabilística que actúe de forma análoga con la función de selección.

2.2.1.8. Inicialización

El proceso de inicialización se mantiene simple en la mayoría de los casos de aplicación de algoritmos evolutivos. La población inicial es generada de forma aleatoria. Puede de igual forma generarse utilizando métodos heurísticos. Esto inicializaría una población con un mayor valor de aptitud general que un inicio aleatorio.

2.2.1.9. Terminación

El algoritmo se repite hasta que se alcanza una condición de terminación. Esta condición puede ser:

- Una solución específica satisface el criterio de parada.
- Se alcanza un número predeterminado de generaciones.
- Se alcanza un límite de recursos (tiempo/dinero)
- Convergencia de las soluciones encontradas
- La inspección manual
- Combinaciones de las anteriores

2.2.2. Aplicaciones de los algoritmos evolutivos

Los algoritmos evolutivos están clasificados dentro de la categoría de algoritmos de optimización, ya que evidentemente no son los únicos, pero a través de los años se ha ido comprobando que pueden llegar a ser muy robustos. Las aplicaciones de los algoritmos evolutivos han sido muy conocidas en áreas como diseño de circuitos, cálculo de estrategias de mercado, reconocimiento de patrones, acústica, ingeniería aeroespacial, y demás. Cualquier problema de optimización puede aplicar un algoritmo evolutivo para su resolución. La efectividad del mismo depende de la forma de representación del espacio de soluciones y la correcta selección de parámetros.

Los algoritmos evolutivos se han consolidado como algoritmos robustos y confiables frente a una gran variedad de problemas. Sin embargo, aún existen métodos

adaptados al problema que resultan más eficientes. En la Figura 4 se puede identificar una comparación entre algoritmos evolutivos, métodos adaptados al problema, y búsqueda randómica. Los algoritmos genéticos superan fácilmente a algoritmos de búsqueda randómica, pero sin embargo no resultan tan efectivos frente a un pequeño rango de problemas específicos solucionables mediante métodos adaptados al problema.

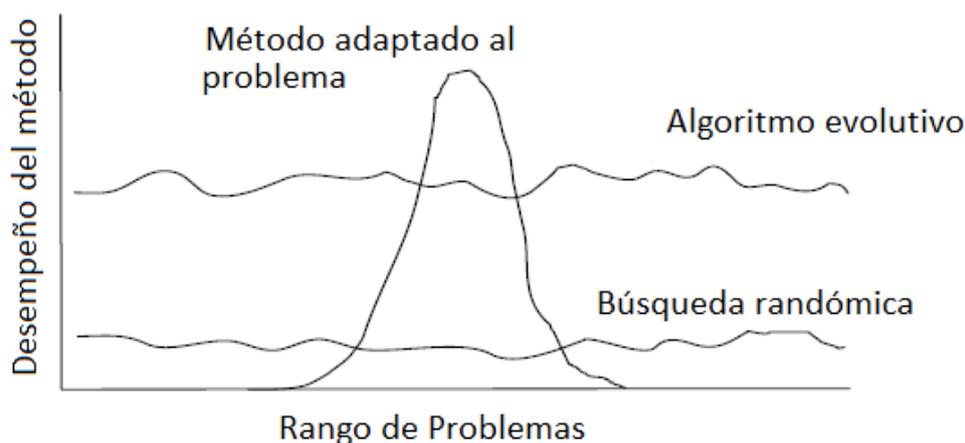


Figura 4 Aplicaciones de algoritmos evolutivos

Fuente: (Fonseca & Fleming, 1995)

2.3. Enjambres robóticos

La robótica de enjambres puede ser definida como el estudio e implementación de enjambres de agentes físicos relativamente simples que pueden llevar a cabo, de forma colectiva, tareas que están más allá de las capacidades de un solo agente. Existen varias diferencias con otros estudios sobre sistemas multi-robot. La robótica de enjambres hace hincapié en la auto-organización e inteligencia colectiva sin perder de vista los problemas de escalabilidad y robustez (Dorigo, 2005). Este énfasis promueve el uso de robots relativamente simples, dotados de capacidades sensoriales específicas, mecanismos de comunicación escalables y exploración de estrategias de control descentralizadas.

Con los recientes avances tecnológicos, el estudio de los enjambres robóticos se está volviendo más factible. Existe un número de proyectos en curso que tienen como objetivo desarrollar y/o controlar un gran número de agentes físicos (Sahin, 2004). En

Europa, la Comisión de las Comunidades Europeas (CCA) ha estado financiando los estudios de robótica de enjambre a través de su programa de Tecnologías Futuras y Emergentes (FET). En los EEUU, la Agencia para la Investigación en Proyectos Avanzados de Defensa (DARPA) ha financiado proyectos de robótica de enjambre a través de su programa de Software para robótica distribuida (SDR). En la Figura 5 se observa el primer enjambre robótico producido a gran escala, con 1024 agentes, por parte del Instituto Wyss, de la Universidad de Harvard.

2.3.1. Inspiración natural

El comportamiento de un enjambre natural tiene tres características principales que son deseables en un sistema robótico y pueden ser vistas como motivaciones para el enfoque de la robótica de enjambre (Sahin, 2004):

- **Robustez:**

El enjambre se mantendrá operativo, a pesar de fallos en los individuos, o alteraciones en el medio ambiente. Esta robustez se puede explicar debido a varios factores: La redundancia del sistema, la coordinación descentralizada, la sencillez de los agentes, y las capacidades de sensado.

- **Flexibilidad**

El enjambre puede generar diferentes soluciones para diferentes tareas. Por ejemplo, una colonia de hormigas puede dedicarse a la recolección o a la caza, siendo el mismo agente pero con un comportamiento modificado.

- **Escalabilidad**

El enjambre puede funcionar casi independientemente de su tamaño. Los mecanismos de coordinación garantizan el funcionamiento del enjambre pesar de su tamaño.



Figura 5 Kilobots, desarrollados por Wyss Institute.

Fuente: (Dorigo, 2005)

2.3.2. Características constitutivas

Tanto la miniaturización y el costo de implementación son factores clave en la robótica de enjambres. Estas son limitaciones en la construcción de grandes grupos de robots. Por lo tanto, se debe enfatizar en la simplicidad del agente. Existen consideraciones generales a tomar en cuenta al momento de definir y desarrollar un enjambre robótico (Balch, 2004).

2.3.2.1. Comunicación

Una de las características fundamentales de un enjambre robótico es la forma de comunicación, o interacción entre agentes. Las opciones en la literatura se pueden dividir en dos categorías: la comunicación directa e indirecta (Dorigo, 2005). En la comunicación directa se establece la comunicación específica entre dos o más agentes del enjambre. Para esto, existen métodos como Bluetooth, infrarrojo, Wi-Fi, entre otros (Blum & Li, 2008). Por otro lado, para las comunicaciones indirectas, los agentes no interactúan entre sí, pero utilizan métodos de broadcast para informar al resto del enjambre. Existen métodos como huellas químicas, emisiones de luz (Figura 6), modificaciones del entorno entre otros.

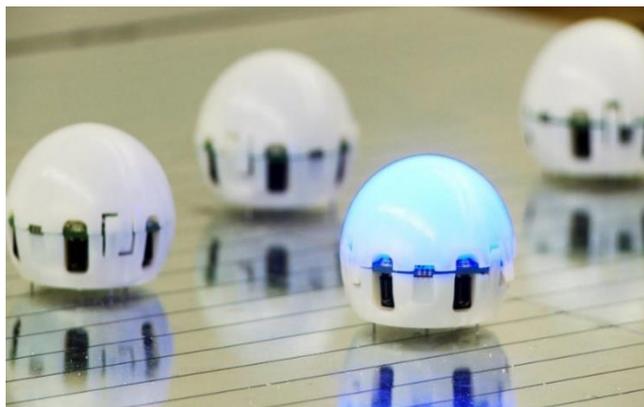


Figura 6 Comunicación indirecta con emisión de luz.

Fuente: *(Beni, 2004)*

Existe un tercer enfoque que está siendo desarrollado actualmente en donde se utilizan agentes sin capacidades de comunicación (De Silva, Ghrist, & Muhammad, 2005). En este caso, se simplifica el hardware necesario, y se presta más atención al comportamiento individual del agente. Además, es posible desarrollar agentes de tamaños más reducidos.

La comunicación, o ausencia de ella, es la consideración principal para equipos de robots. Esto permite ciertos tipos de coordinación que serían imposibles sin sistemas de comunicación. Por otro lado, la comunicación aplicada a un sistema robótico puede aumentar los costos y la complejidad de forma exponencial. La selección del tipo de comunicación, o incluso su ausencia, depende exclusivamente de la aplicación para el enjambre robótico.

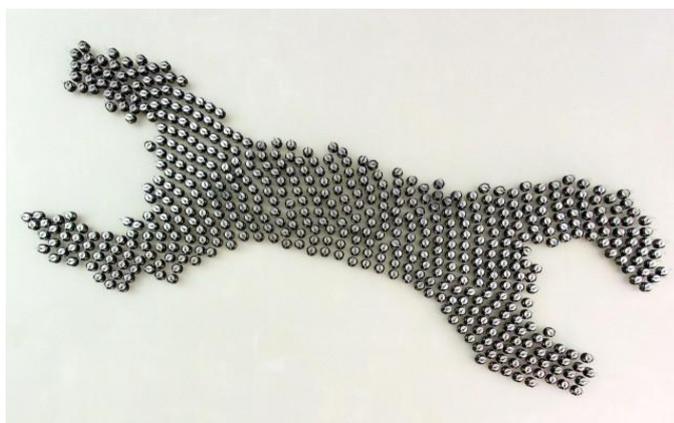


Figura 7 Formación de patrones con Kilobots.

Fuente: *(Dorigo, 2005)*

2.3.2.2. Diversidad

La diversidad se refiere a las diferencias, ya sean físicas o de comportamiento, entre los agentes de un enjambre robótico. La misma temática de diversidad establece un problema propio de investigación en el cual se analizan las ventajas y desventajas en sistemas homogéneos y heterogéneos.

De forma similar al apartado 2.3.2.1, la aplicabilidad de sistemas homogéneos o heterogéneos depende exclusivamente de la tarea escogida. Según la misma es posible definir la necesidad de implementar sistemas heterogéneos en hardware, es decir con sistemas de actuadores y sensores diferentes entre sí, o simplemente efectuar una diferenciación en el comportamiento de cada agente.

Estos diferentes conceptos fueron aplicados en la construcción de S-BOT (Figura 8), en donde cada agente fue dotado de varios actuadores y sensores, como cámaras, infrarrojo, pinzas retráctiles, motores, entre otros, para crear un enjambre robótico adaptable según la tarea a desarrollar. Este enjambre puede recolectar, ubicar objetivos, movilizar objetos pesados, atravesar obstáculos y muchas tareas más. (Dorigo, 2005)



Figura 8 S-BOT desarrollado por EPFL para navegación por terreno

Fuente: (Dorigo, 2005)

2.3.2.3. Tamaño del enjambre

Al hablar de enjambres en el ámbito biológico, se entiende como sistemas compuestos desde varios cientos hasta miles de agentes. Sin embargo, en términos

prácticos, para un enjambre robótico, esto resulta imposible debido a costos y limitaciones físicas.

Existen 4 grandes grupos en los cuales se analiza el desempeño de un enjambre robótico:

- Individuo: 1 robot. En este caso es imposible hablar de enjambre
- Par: 2 robots. El grupo más simple.
- Grupo limitado: Múltiples robots. El número es relativamente pequeño en comparación con la tarea o entorno.
- Grupo infinito: $n \gg 1$.

De igual forma, resulta determinante la tarea escogida, y en este caso, el entorno de trabajo, para determinar el número justo de agentes para conformar el enjambre.

Dos robots pueden desarrollar tareas evidentemente imposibles para un solo agente. Casi cualquier operación que involucre la presencia espacial en dos lugares de forma simultánea, resultaría imposible para un solo agente.

Múltiples robots podrían en este caso acelerar la finalización de la tarea, o incluso dar paso a la realización de nuevas tareas de sincronización.

Finalmente, establecer un número específico de agentes resulta subjetivo y depende del investigador. Una misma tarea puede ser completada por un gran número de agentes sencillos, mientras que requeriría un menor número de agentes más complejos. Todo esto tiene su repercusión en costos de implementación.

2.3.3. Métodos de diseño

El diseño involucra la planificación y desarrollo del enjambre a partir requerimientos y especificaciones iniciales. En el campo de la robótica de enjambres, resulta imposible establecer, de manera formal, uno o varios métodos de diseño de comportamiento individual que genere un comportamiento global deseado.

Es necesario la presencia del diseñador o investigador, y en este caso la intuición del mismo tiene un papel fundamental en este proceso. Según varios estudios al respecto (Brambilla, Ferrante, Birattari, & Dorigo, 2013), la mayoría de métodos de diseño caen en dos categorías principales: Diseño basado en el comportamiento y diseño automático.

2.3.3.1. Diseño basado en comportamiento

El diseño basado en comportamiento es actualmente el método más utilizado. En esta aproximación, se establece de forma manual, el comportamiento individual de los agentes para generar un comportamiento global. Es un proceso iterativo de prueba y error. De forma recurrente, el comportamiento individual es implementado, estudiado y mejorado, hasta que se obtenga el comportamiento global deseado. Otro enfoque recientemente expuesto realiza el proceso inverso, especificando el comportamiento global y buscando el comportamiento individual de forma iterativa.

Para este método, la inspiración viene de la observación de enjambres naturales y el entendimiento de sus interacciones. Este proceso se facilita cuando se entiende el comportamiento individual del agente e incluso pueden encontrarse modelos matemáticos del mismo. (Beni, 2004)

2.3.3.2. Diseño automático

Otra forma de diseñar un enjambre son los llamados métodos automáticos. En este caso, se generan de forma automática los comportamientos sin la intervención directa del diseñador. De igual forma estos métodos caen en dos categorías principales: Aprendizaje reforzado y robótica evolutiva.

- Aprendizaje reforzado

El aprendizaje reforzado se refiere de forma tradicional a un tipo específico de problemas de aprendizaje. Un agente aprende un comportamiento a través de interacciones consecutivas de prueba y error con el entorno y recibe retroalimentación en forma de recompensas positivas y negativas por sus acciones. El objetivo final es identificar una política o comportamiento óptimo que maximice la recompensa obtenida del entorno. (Kaelbling, Littman, & Moore, 1996)

Sin embargo, existen varios inconvenientes al implementar este método en enjambres físicos. El enjambre cumple con su tarea de forma global, pero en el aprendizaje reforzado, el proceso de recompensa es a nivel individual. De esta forma, el problema principal es la descomposición de la recompensa global en recompensas individuales.

- Robótica evolutiva

Por otro lado, la robótica evolutiva, utiliza técnicas de computación evolutivas (algoritmos evolutivos) para optimizar el comportamiento del enjambre. La computación evolutiva está inspirada en el principio darwiniano de la selección natural y evolución.

La robótica evolutiva puede ser descrita de la siguiente manera: Al inicio del proceso, es generada una población de comportamientos individuales. En cada iteración, se ejecuta cierto número de experimentos con cada comportamiento individual, implementado por todo el enjambre. En cada experimento, se establece una función de aptitud que evaluará el desempeño del comportamiento global del enjambre, producto de la implementación del comportamiento individual. En este punto, serán seleccionados los comportamientos con una mayor aptitud y modificados por operadores evolutivos como combinación y mutación para las subsiguientes iteraciones. (Chang, Lee, Fu, & Marcus, 2005)

Como puede verse, este enfoque está aplicado mayoritariamente a enjambres homogéneos en donde todos los miembros del enjambre implementan el mismo comportamiento individual. El comportamiento individual puede ser representado de varias maneras como máquinas de estado, o acciones predefinidas.

2.3.4. Métodos de análisis

La metodología de análisis resulta fundamental para validar el comportamiento del enjambre y resulta esencial en cualquier proceso de ingeniería. En esta fase, es de interés del investigador, identificar la existencia de ciertas propiedades en el comportamiento global del enjambre. El objetivo último es validar y comprobar que el enjambre exhibe el comportamiento global deseado con las propiedades deseadas. Todo este análisis se lo realiza por medio de modelos. (Brambilla, Ferrante, Birattari, & Dorigo, 2013)

Los enjambres robóticos pueden ser modelados en dos diferentes niveles: Nivel individual, o microscópico, en el cual es modelado el comportamiento individual de cada agente y sus interacciones tanto con el entorno como entre ellos; y Nivel colectivo, o macroscópico, en el cual son modelados los enjambres en su totalidad

como uno solo, mediante sus características globales. Adicionalmente existe otro tipo de análisis con los robots reales.

2.3.4.1. Modelos microscópicos

Los modelos microscópicos toman a consideración cada robot de forma individual y analizan las interacciones entre ellos y con el entorno. Es de especial atención el nivel de detalle para el modelo y afectará directamente a los resultados que puedan ser obtenidos.

Para estos existen varios niveles de abstracción. En el nivel más simple en el cual se considera a cada robot como masas puntuales. Un nivel intermedio considera las representaciones 2D de los robots y con su consecuente análisis de la cinemática involucrada. Los modelos más complejos incluyen representaciones 3D con un análisis dinámico y físico, en donde se incluye el modelamiento de cada sensor y actuador.

El nivel de abstracción a utilizarse depende del tipo de análisis que sea requerido. El componente principal de todos estos niveles, es el modelamiento específico del comportamiento del agente. Los modelos microscópicos en donde se simulen las interacciones de los elementos constitutivos del enjambre mediante el uso de un computador son tradicionalmente llamados *simulaciones*. Las simulaciones son las herramientas más utilizadas para analizar y validar sistemas de enjambres robóticos.

2.3.4.2. Modelos macroscópicos

Los modelos macroscópicos consideran a los sistemas de enjambres robóticos en su conjunto. Es decir que no son tomados en cuenta los elementos individuales o agentes. Se realiza una descripción del sistema a un nivel superior.

Para realizar este modelamiento, es necesario involucrar matemática avanzada. Se toman en cuenta teorías de control clásico así como de estabilidad, y se modela el sistema utilizando ecuaciones diferenciales y tasas de variación.

2.3.4.3. Análisis de robots reales

El uso de robots reales, en oposición al uso de simulaciones, es igualmente una herramienta fundamental en la robótica de enjambre. Este enfoque está justificado en el hecho de que no es posible simular todos los aspectos de la realidad.

El trabajo con un enjambre físico permite:

- Probar la robustez del sistema
- Validar simulaciones con sensores y actuadores de señales ruidosas.
- Discriminar entre comportamientos colectivos realizables y aquellos que únicamente trabajan bajo condiciones irreales.

Es necesario recalcar el hecho que la mayoría de experimentos presentados en los trabajos de investigación utilizan entornos adaptados y controlados. Esto dista bastante de las condiciones reales en las cuales se puede potencialmente aplicar el sistema robótico. La implementación en este tipo de entornos no debe tomarse como una verificación del enjambre sino simplemente como una prueba real de los sensores y actuadores de los agentes.

En la gran mayoría de trabajos, el proyecto llega simplemente a la etapa de simulación (Das, Behera, & Panigrahi, 2016), y en ciertos casos, a una implementación sutil (una o dos ejecuciones) que tiene por objetivo demostrar la aplicabilidad del sistema diseñado. Esto puede ser ocasionado principalmente por los costos de implementación de un sistema robótico con las características de un enjambre.

2.3.5. Comportamientos colectivos

Existen una serie de tareas estándar que son mencionadas reiteradamente en la literatura. En todas las opciones se enfatiza el trabajo en equipo (Figura 9) y la flexibilidad de los sistemas enjambre. Estos comportamientos colectivos (tareas a realizar) son:

Formación y patrones

- Conglomeración
- Auto-organización en patrones
- Distribución de antenas
- Cubrir áreas
- Limpieza de áreas
- Mapeado



Figura 9 Enjambre robótico trabajando en equipo.

Fuente: (*Beni, 2004*)

Enfocado en objetivos:

- Búsqueda de objetivos
- Retorno a nido
- Búsqueda de sustancias químicas
- Recolección
- Caza

Comportamientos complejos:

- Transporte cooperativo
- Minado
- Pastoreo
- Movimiento en bandada

Comportamientos genéricos:

- Evasión de obstáculos
- Navegación en terreno (Figura 10)



Figura 10 Navegación grupal en terreno.

Fuente: (Dorigo, 2005)

2.4. Aplicaciones combinadas

Los trabajos e investigaciones asociados con las temáticas de enjambres de robots y algoritmos evolutivos han ido aumentando a través de los años. Los enjambres robóticos son aplicados en tareas cada vez más complejas de analizar y es necesaria la implementación de procesos de optimización como lo son los algoritmos evolutivos.

Empezando con los SWARM-BOTS (Dorigo, 2005), es un proyecto robótico patrocinado por el programa de Tecnologías Futuras y Emergentes de la Comisión Europea. Este es un sistema robótico bastante avanzado que incluye implementación física y el desarrollo de software de simulación. Se utilizan algoritmos adaptativos para controlar un grupo de robots reales o simulados, para que realicen una variedad de tareas que requieren la cooperación y la coordinación (Figura 11). De forma específica, la locomoción del agente está dominada por un algoritmo genético que optimiza los pesos de conexión en una red neuronal y controla el movimiento de cada motor según los sensores infrarrojos para evadir obstáculos.

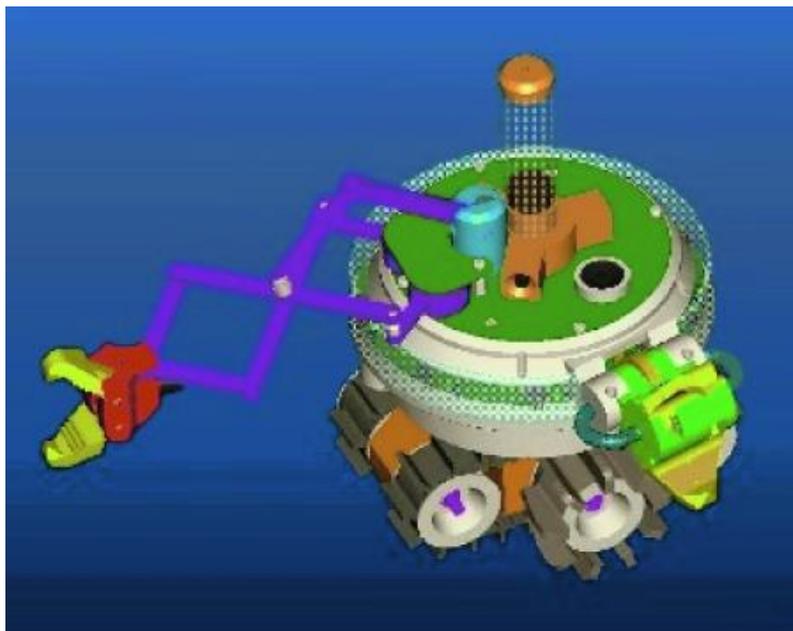


Figura 11 Diseño 3D de un S-BOT en el simulador.

Fuente: (Dorigo, 2005)

En otras aplicaciones innovadoras se encuentran en los Pherobots que son parte del proyecto Robótica de Feromonas realizado por investigadores de los laboratorios HRL, EEUU (Payton, Estkowski, & Howard, 2004). Aquí el concepto se basa en las feromonas químicas y el gradiente de feromonas que utilizan ciertos insectos sociales para transmitir mensajes de forma indirecta y crear comportamientos elaborados. Cada pherobot contiene un arreglo omnidireccional de sensores y emisores infrarrojos que simula feromonas virtuales, siendo posible utilizar esta información para ubicar obstáculos y agentes próximos. El comportamiento del agente es optimizado según esta información para generar un movimiento acorde a las feromonas que detecte. Fue desarrollado un dispositivo de realidad aumentada para observar dichas feromonas en tiempo real (Figura 12)



Figura 12 Feromonas visualizadas en tiempo real a través de realidad aumentada

Fuente: (Payton, Estkowski, & Howard, 2004)

En problemas de localización distribuida y mapeo con enjambres robóticos (Rothermich, Ecemis, & Gaudiano, 2004), existen varias implementaciones simultáneas de algoritmos evolutivos para resolver los problemas que se plantean. En la propuesta inicial del programa de Software para Robótica Distribuida (SDR) del DARPA, el objetivo se centra en mapear y encontrar objetos de interés en áreas desconocidas. Para este propósito, investigadores de la corporación Icosystem plantean un enjambre capaz de mapear y localizar mediante el uso de señales IR además de odometría. Los algoritmos empleados resuelven varios problemas: localización colaborativa, mapeo colaborativo (Figura 13), asignación de tareas dinámica. Cada algoritmo es optimizado fuera de línea (computación evolutiva) y posteriormente implementado.

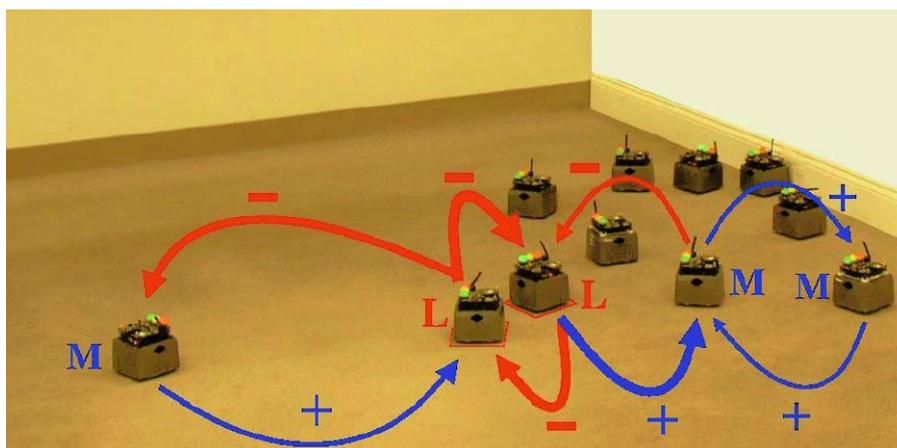


Figura 13 Reglas de movimiento optimizado para un mapeo distribuido

Fuente: (Rothermich, Ecemis, & Gaudiano, 2004)

Enfoques más completos se encuentran en enjambres robóticos para problemas de limpieza dinámica (Altshuler, Bruckstein, & Wagner, 2005). Para este objetivo son diseñados modelos de comportamiento para aprendizaje reforzado y se va más allá al optimizar la búsqueda de la política mediante algoritmos evolutivos. Como fue mencionado en los métodos de diseño, el principal inconveniente para aprendizajes reforzados está en la implementación, por este motivo dichos tipos de proyectos son simulados (Figura 14).

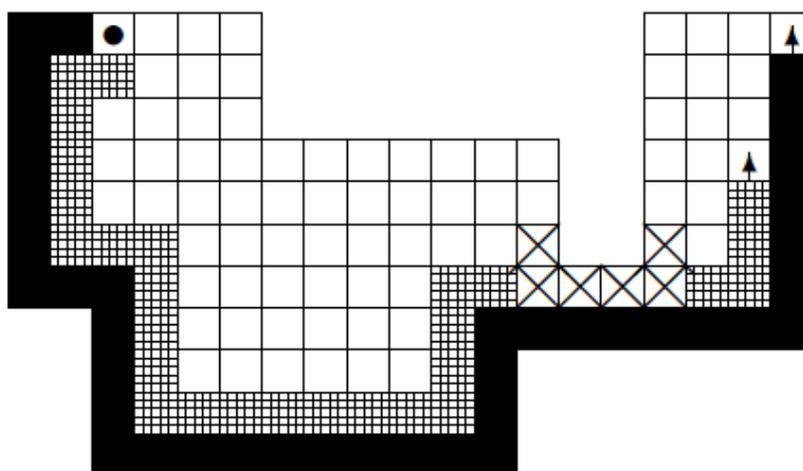


Figura 14 Entorno creado para limpieza dinámica.

Fuente: (Altshuler, Bruckstein, & Wagner, 2005)

2.5. Descripción del proyecto

Una vez presentado el estado del arte de las temáticas de este proyecto de investigación, se describe el trabajo a realizarse y se explica el orden lógico que se llevó a cabo para cumplir con totalidad el objetivo planteado.

La característica más importante como fue mencionado a lo largo del presente capítulo, es la selección de la tarea que realizará el enjambre. Basándose en las tareas listadas en la sección 2.3.5, fue seleccionada la limpieza de un área. Esta tarea simplifica el hardware necesario ya que, de forma conceptual, la idea más básica para limpiar un área, es mover los obstáculos fuera de ella. En este caso, es necesario un sistema de movimiento omnidireccional y un método para detectar la presencia de obstáculos. Esta tarea puede ser realizada de igual forma sin la implementación de un sistema de comunicación por lo que simplifica aún más el diseño de hardware. Otras tareas fueron tomadas en cuenta como recolección o ubicación de objetivos, sin embargo, el hardware necesario se vuelve más complejo y costoso (Beni, 2004; Dorigo, 2005).

Para el desarrollo del proyecto, se diseñó e implementó el sistema físico para poder identificar las interacciones cinéticas que rigen al sistema y sirvieron de base para diseñar la simulación. El sistema físico consta del conjunto de agentes, o enjambre, y del entorno físico, utilizado en este caso como fuente de alimentación. Con esta información, se diseñó un simulador, o modelo microscópico, de 2 dimensiones, que abarca la mayor cantidad de interacciones posibles entre agente-entorno-obstáculo. Con el simulador fue posible evaluar el desempeño del enjambre sin la necesidad de implementar cada cambio efectuado en el comportamiento. Para esto, fue necesario definir un método de evaluación del desempeño del enjambre simulado que pueda ser a su vez implementado en el enjambre robótico.

Una vez desarrollado el simulador del sistema, se diseñó un algoritmo evolutivo (algoritmo genético) que optimiza la conducta de cada agente, evaluando el comportamiento global obtenido en el cumplimiento de la tarea (limpieza del área), basándose en los resultados de la simulación. El desempeño del algoritmo fue verificado para comprobar la convergencia del mismo en un solo resultado.

El resultado optimizado fue implementado en el enjambre físico, así como simulado en software. Se procedió a realizar experimentos y comparaciones respectivas para validar el trabajo realizado. Todo el proceso lógico llevado a cabo puede resumirse en la Figura 15.

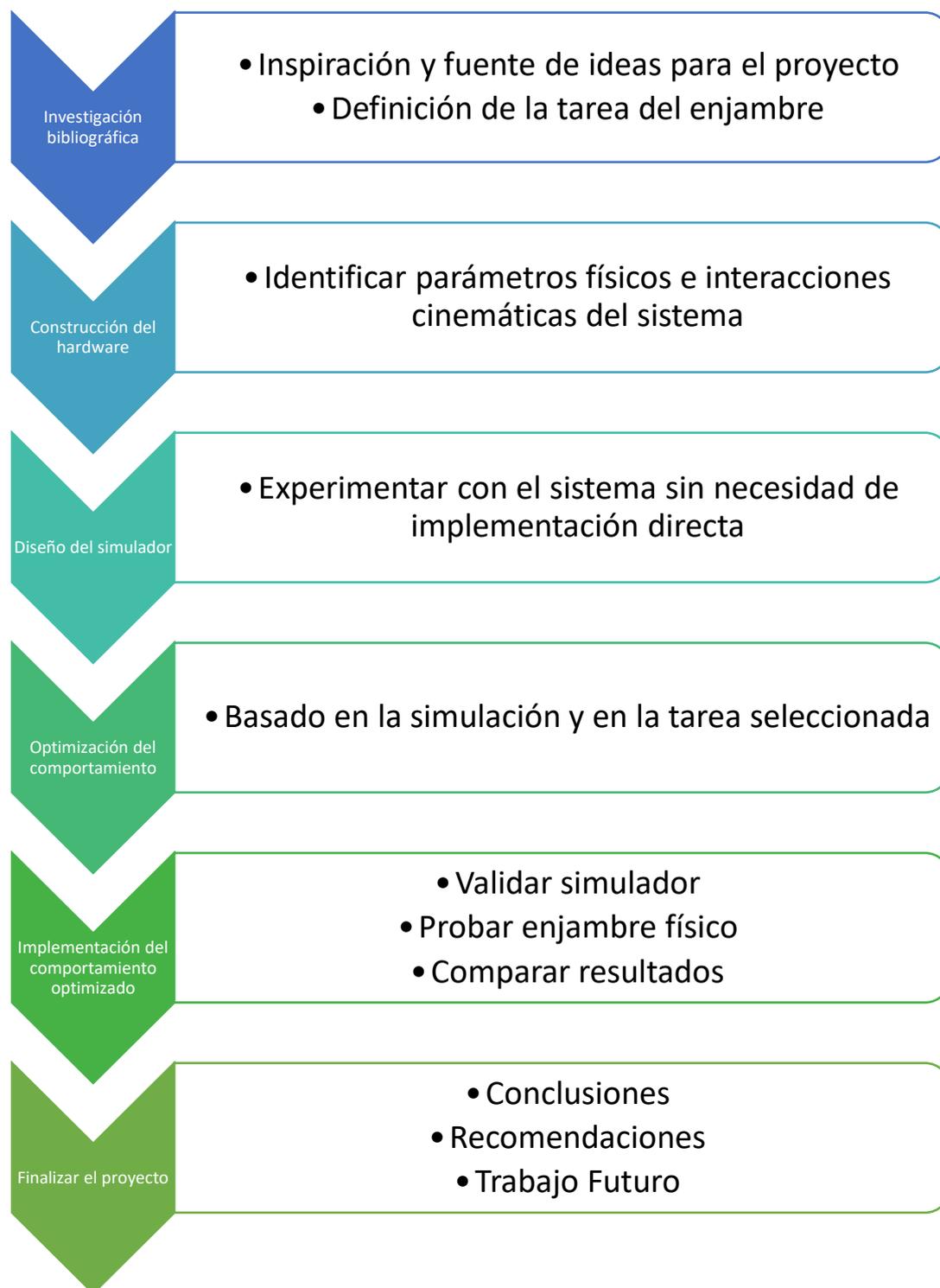


Figura 15 Secuencia de actividades para el desarrollo del proyecto

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN DE HARDWARE

3.1. Introducción

En este capítulo se detalla el proceso de diseño e implementación del enjambre robótico, basándose en las consideraciones de diseño presentadas y en la tarea seleccionada. Se describe cuáles serán las características constitutivas de los agentes. Se diseñó el sistema de detección de colisiones que corresponde con el sistema principal del agente, dado que no posee capacidades de comunicación ni sensores adicionales. Se presenta las conexiones adicionales necesarias para el funcionamiento del agente.

Adicionalmente, se detalla el proceso de diseño y construcción del entorno de pruebas. Es presentada una tabla descriptiva con los costos de implementación del hardware. Como temática adicional al hardware implementado, se explica el funcionamiento del código embebido en cada agente del enjambre.

3.2. Consideraciones de diseño

Las consideraciones de diseño se conforman por un conjunto de parámetros y características, necesarias de definir previo a la construcción del enjambre. Todo el sistema físico será implementado con un propósito en específico, alcanzable mediante un comportamiento global que implemente las consideraciones aquí descritas:

- Comunicación
- Sensores y Actuadores
- Procesamiento
- Sistema de energización
- Tamaño del enjambre
- Entorno

3.2.1. Comunicación:

La tarea fue escogida teniendo en cuenta la premisa del agente más sencillo posible en términos de hardware. Dicha consideración permite minimizar el tamaño y el costo del robot ya que la tarea seleccionada, la limpieza de un área, en su concepción más básica, no necesita la implementación de sistemas de comunicación. Para este enfoque, el agente solamente podrá detectar su entorno a través de su sistema de detección de colisiones.

3.2.2. Sensores y Actuadores:

Para los actuadores, se ha seleccionado un sistema de movimiento simple, con dos motores. Para la etapa de sensado se optó por la retroalimentación de los mismos motores para identificar las variaciones de corriente, que se originan al variar de forma externa la rotación del eje, por ejemplo, cuando el robot se detiene de forma imprevista debido a un obstáculo en su camino.

Con este sistema es posible crear un robot simple capaz de detectar colisiones y reaccionar frente a ellas.

3.2.3. Procesamiento:

Para procesar las señales de entrada y generar las señales de salida es suficiente un microcontrolador. Se selecciona el microcontrolador ATMEGA32L ya que cumple con los requisitos necesarios para este proyecto. Adicionalmente, este microcontrolador posee un bajo consumo de energía, que es una característica deseable en estos sistemas. Los detalles más importantes de este microcontrolador se resumen en la Tabla 1.

Tabla 1
Resumen de características ATMEGA32L.

Característica	Valor
SRAM	2 KB
ROM	32 KB
EEPROM	1024 B
Pines I/O	32

Continua 

Corriente máx. (c/Pin)	40 mA
Corriente máx.	200 mA
Velocidad	16 MHz
Bits ADC	10 bits
Salidas PWM	4
Vcc (Min.)	2.7 V
Timers	3
Canales ADC	8

Fuente: www.atmel.com

3.2.4. Sistema de energización

Otro punto importante a considerar es la forma de energizar al agente. El enjambre debe ser autónomo, con el fin de poner en práctica y evaluar los resultados del algoritmo genético completo. Así que la energía puede ser un problema si se energiza con baterías normales. Hubo varios enfoques que fueron considerados.

- **Baterías recargables:** Cada agente tiene una batería recargable y cuando está casi descargada, el agente se dirige a un punto de carga. Podría ser útil como una tarea por sí misma.
- **Recarga inductiva:** La idea es utilizar módulos de carga inductiva para energizar el agente. Los módulos constan de dos partes, bobinas energizadas fijada en el suelo, y el módulo receptor en el agente que da la energía. La desventaja es el precio de estos módulos. Se compromete el costo de todo el proyecto.
- **Tiras energizadas:** Constan de tiras alternadas de 0V y 5V sobre el suelo y en conjunto con un sistema de conmutación en el agente, es posible tener siempre energizado al enjambre.
- **Placas energizadas:** Dos placas polarizadas, en el suelo y en el techo del entorno, darán la energía necesaria a los agentes, los cuales implementaran el sistema físico de contacto necesario.

De todas las opciones presentadas, se seleccionaron las placas energizadas, tanto por su facilidad de implementación como en el costo que representan. El ambiente será diseñado y construido con la ayuda del laboratorio mecánico de la Universidad de Ciencias Aplicadas de Brandeburgo.

3.2.5. Tamaño del enjambre

Como se ha mencionado antes, se enfatiza la simplicidad del robot. Siendo este el caso, el agente resultante es muy simple y, teóricamente, el tamaño enjambre puede ser tan grande como sea necesario. A efectos prácticos, y debido al espacio físico disponible, el enjambre va a tener 10 agentes.

3.2.6. Entorno

El entorno a desarrollarse delimitará el área del experimento y además será el encargado de energizar a los agentes. Con estas premisas, el entorno debe estar formado por placas conductoras lo suficientemente amplias y a su vez rígidas para cumplir con el objetivo.

Con ayuda del laboratorio de mecánica de la Universidad de Ciencias Aplicadas de Brandeburgo se pudo disponer de dos placas de aluminio de 2mm de espesor de 80x80 cm.

3.3. Resumen de características del enjambre

En la sección 3.2 fueron presentadas una serie de consideraciones que definen el hardware a diseñarse, así como el comportamiento global deseado. Es necesario recalcar que el enjambre físico no implementó ningún sistema de comunicación y su sistema de sensado está compuesto por la retroalimentación del sistema de movimiento. Esto enfatiza la premisa de emplear el agente funcional más sencillo posible. Las características son resumidas en la Tabla 2.

Tabla 2
Resumen de características del enjambre a implementarse

Características Enjambre	
Comunicación (Agente)	No
Actuadores (Agente)	Dos motores. Capacidad de movimiento omnidireccional del agente (2-D)
Sensores (Agente)	Retroalimentación de motores. Capacidad de detectar colisiones.
Procesamiento (Agente)	Microcontrolador (ATMEGA32L)
Energía (Agente)	Sistema de contacto a placas polarizadas.
Nº de agentes	10
Entorno	Placas polarizadas 80x80 cm
Tarea	Limpieza de área

3.4. Implementación del sistema de detección de colisiones

El agente implementará un sistema de detección de colisiones mediante la retroalimentación de los motores. Para minimizar el tamaño del agente es necesario utilizar motores capaces de activarse directamente con las salidas digitales del controlador. Para este propósito se utilizaron los micro-motores FF-130SH (Figura 16). Los detalles técnicos del mismo pueden observarse en la Tabla 3.



Figura 16 Micro-motores FF-130SH

Fuente: <http://www.mabuchi-motor.co.jp/>

Tabla 3
Características del micro-motor FF-130SH

Tensión	Sin carga			Máxima Eficiencia		Bloqueado
Rango	Nominal	Rapidez	Corriente	Rapidez	Corriente	Corriente
Operativo	V	r/min	A	r/min	A	A
2.5-10 V	6	7400	0.050	6060	0.23	1.08

Fuente: <http://www.mabuchi-motor.co.jp/>

Para obtener la retroalimentación, se conectan las salidas digitales que controlan a los motores, a través de diodos de protección, a las entradas analógicas disponibles según lo indica la Figura 17. Adicional a esta conexión, es necesario colocar un capacitor de 47 nF entre los terminales de cada motor, como recomendación del fabricante, para disminuir el ruido eléctrico generado por el mismo.

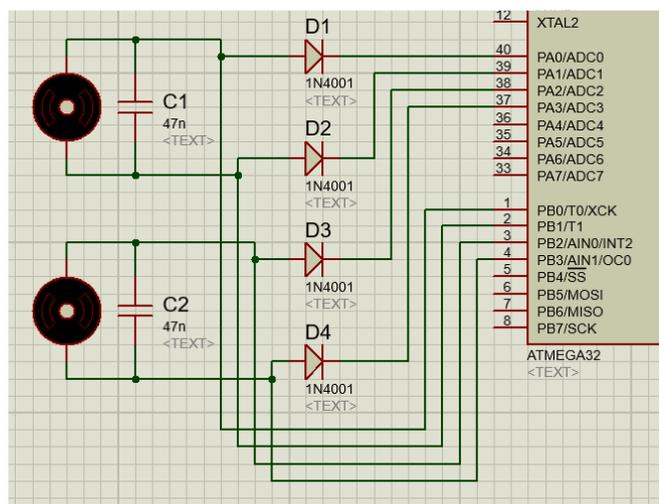


Figura 17 Conexión directa de los micro-motores

Una vez conectado el sistema, se realizaron pruebas de funcionamiento para verificar que exista una variación en las entradas analógicas al frenar de forma externa el motor. Los resultados son presentados en la Figura 18. Es posible evidenciar que existe una diferencia entre los estados de giro libre y bloqueado, pero no están lo suficientemente definidos como para utilizar de forma directa el valor del ADC. La variación de tensión identificada en las entradas analógicas es de aproximadamente 0.4V .

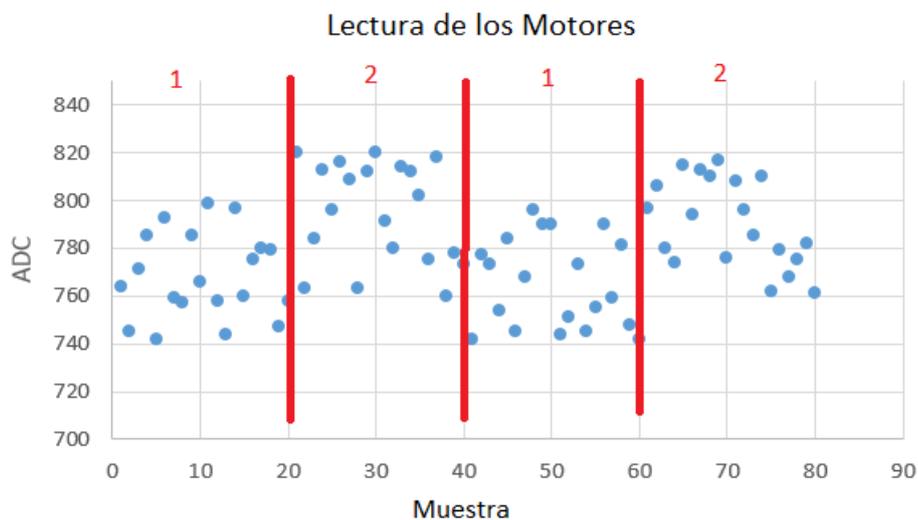


Figura 18 Lecturas analógicas con motor bloqueado (1) y liberado (2).

Para minimizar el ruido eléctrico presente en la señal, fue necesario añadir una red de resistencias tipo pull-down como se muestra en la Figura 19. Esta configuración, aunque no óptima, soluciona el problema del ruido sin la necesidad de complejas implementaciones como filtros o amplificadores.

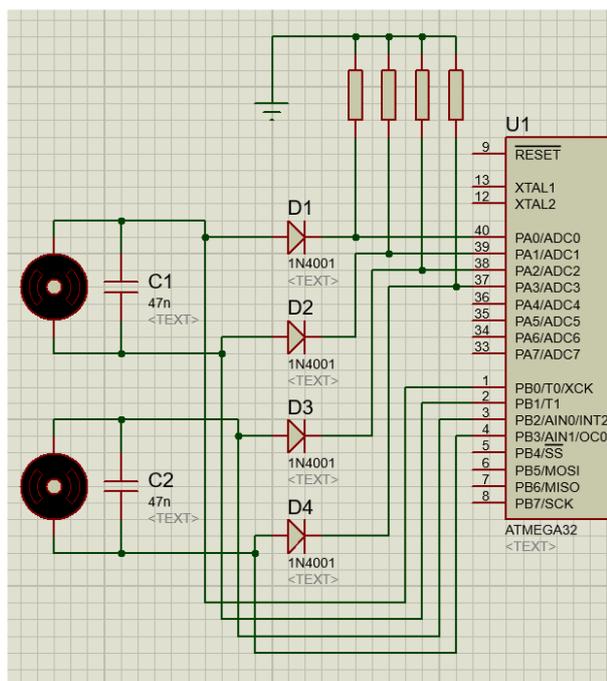


Figura 19 Conexión de los micro-motores con resistencias pull-down.

Los resultados se indican en la Figura 20. En este caso se evidencia una clara mejoría en la señal adquirida, pero sigue presente una variación indeseada que dificulta el análisis. Se utilizará una etapa de filtrado por software para solucionar el problema.

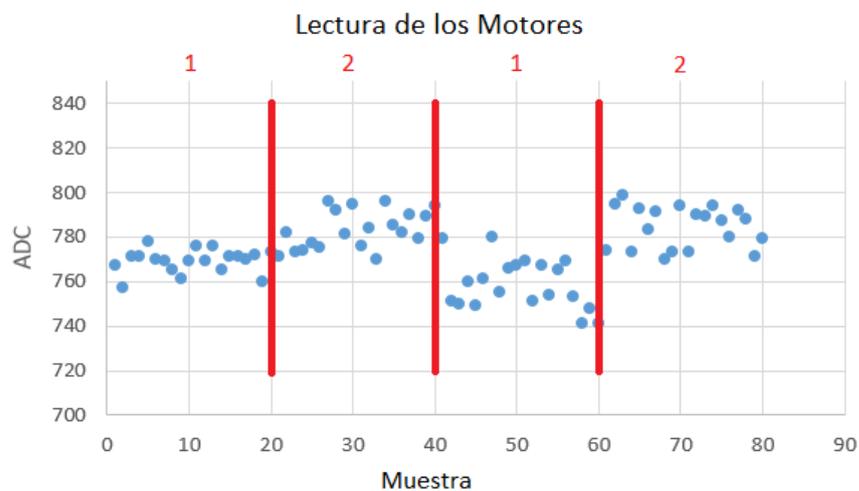


Figura 20 Lecturas analógicas con pull-down, motor bloqueado (1) y liberado (2).

3.5. Hardware adicional

Para completar el diseño físico del agente, se añadió el circuito necesario para la conexión con el programador AVRISP mkII (Figura 21), dos piezas metálicas que funcionan como contactos con las placas polarizadas y un capacitor de 1F que mantiene la tensión en caso de que, durante el movimiento del agente, no exista contacto con las placas polarizadas. El circuito completo así como un diagrama esquemático de las conexiones son presentados en el Anexo A.

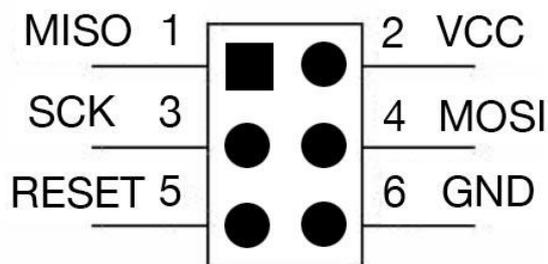


Figura 21 Conexiones para el programador AVRISP mkII.

Fuente: <http://www.atmel.com/>

Para el movimiento del agente, los motores deben colocarse en una posición de trípode, con el contacto metálico (GND) como tercer punto de apoyo. Los motores se ubicarán a 45° con respecto a la horizontal para maximizar el contacto del eje del motor con el terreno (Figura 22). Surge un inconveniente adicional cuando el agente se desplaza sobre las placas de aluminio. La fricción entre los ejes de los motores y las placas no es la suficiente para generar movimiento.

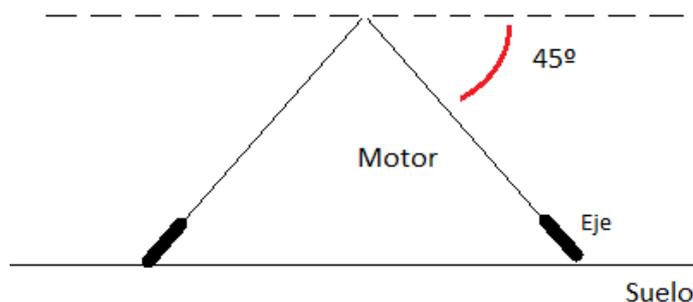


Figura 22 Ubicación de los motores con respecto al suelo.

Es necesario aumentar la fricción entre ambas superficies. Fueron probadas distintas soluciones al problema siendo la más óptima revestir los ejes de los motores con plástico termo-encogible. En la Figura 23 podemos observar las dos versiones construidas del agente.



Figura 23 Agente. Derecha: Primera versión. Izquierda: Versión final.

Como puede observarse, todo el sistema fue recubierto por una semiesfera plástica para proteger el circuito. La versión final mide 8cm de diámetro. Todas las características físicas y eléctricas de importancia se presentan en la Tabla 4.

Tabla 4
Resumen de características del agente

Características del Agente	Valor
Peso	60 g
Rapidez	1 cm/s
Tensión	5 V
Corriente	Nominal: 40 mA. Máxima: 95 mA.
Diámetro	8 cm

3.6. Diseño e implementación del entorno

El entorno tiene que ser la fuente de energía para el enjambre. Como se mencionó anteriormente, está hecho de dos placas de aluminio polarizadas, 80x80 cm. El diseño fue desarrollado con herramientas CAD (SolidWorks) y el resultado se muestra en la Figura 24. Los planos de construcción se pueden encontrar en el Anexo E. En la implementación real, se produjo un comportamiento inesperado. La placa superior se pandeó, debido a su propio peso. Esto causó efectos en el enjambre. La deflexión del techo aumentó la fuerza normal ejercida sobre los agentes y por ende la fricción, causando la detección de colisiones al desplazarse en esta área. Para solucionar el problema fueron añadidas 4 columnas adicionales a lo largo de las paredes y una columna cilíndrica en medio del entorno.

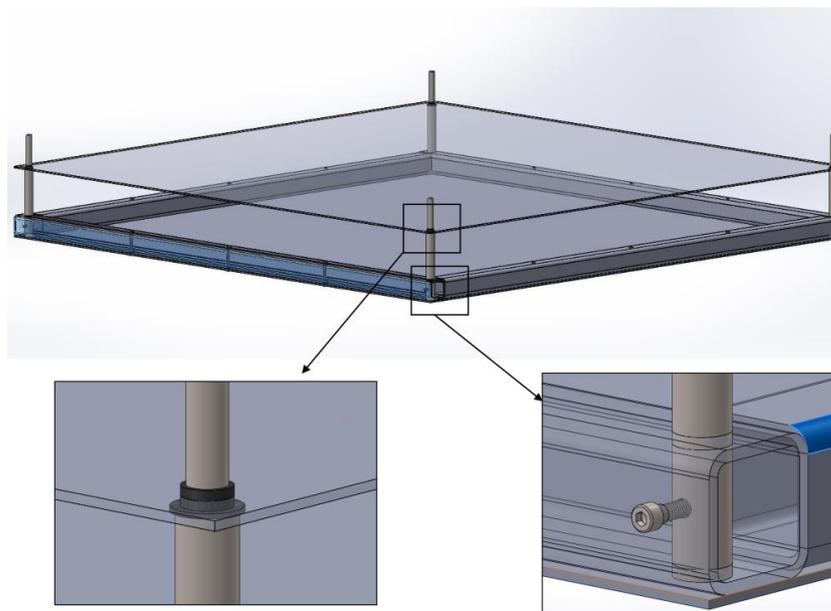


Figura 24 Ensamblaje 3D del entorno.

Las placas son energizadas como se muestra en la Figura 25. Pueden observarse los contactos metálicos implementados. Para el contacto con GND se utilizó un segmento de cable rígido mientras que para el contacto de 5V, se utilizó un segmento de cable rígido con una esponja metálica en el extremo, debido a que genera más puntos de contacto con el techo y su suavidad disminuye el efecto de la presión ejercida por las placas.

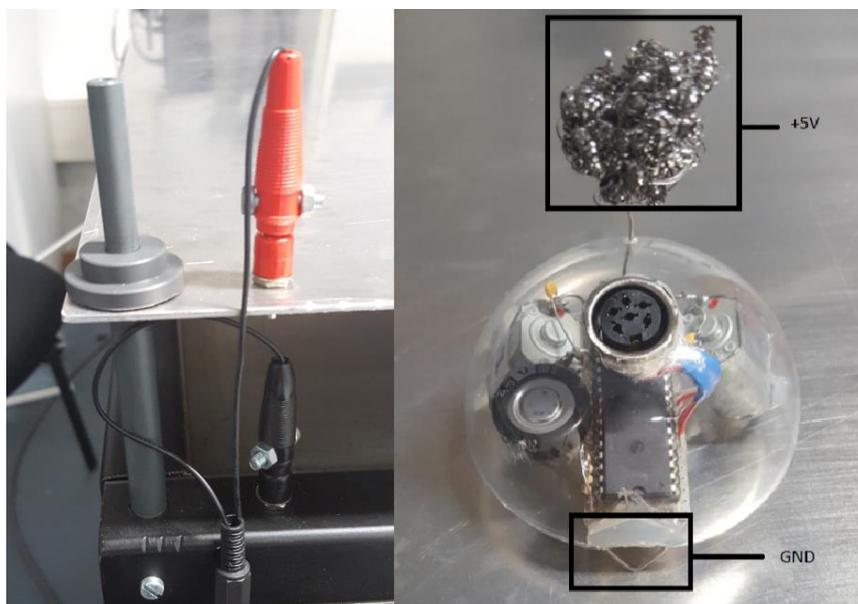


Figura 25 Método de polarización.

Una vez implementado el sistema físico, se detallará el costo final del mismo. Es necesario destacar que el valor final del proyecto fue asumido por la Universidad de Ciencias Aplicadas de Brandeburgo. Los valores detallados se presentan en la Tabla 5. De igual forma en la Figura 26 puede observarse el sistema total construido.

Tabla 5
Costos de implementación

Ítem	Cantidad	V. Unitario (\$)	V. Total (\$)
Placa de cobre 21x29cm	2	2.98	5.96
ATMEGA32L	11	4.75	52.25
Micromotores FF-130SH	22	1.27	27.94
Diodo 1N4001	44	0.02	0.88
Resistor 10K 1/4W	55	0.02	1.1
Capacitor electrolítico 1uF	11	0.03	0.33
Capacitor cerámico 47nF	22	0.02	0.44
Capacitor electrolítico 1F	11	1.5	16.5
Cable cobre AWG24 flexible (Rollo 10 m)	1	3	3
Cable cobre AWG12 rígido (Rollo 5 m)	1	3	3
Tubo termoencogible 4mm (10 cm)	4	0.02	0.08
Esponja metálica	1	0.5	0.5
Esferas plásticas 8cm diámetro (paquete)	1	10	10
Esferas plásticas 6cm diámetro (paquete)	1	10	10
Cilindro plástico 12mm Φ	4	2.2	8.8
Plancha de Aluminio 80x80cm	2	67.65	135.3
Tubo de acero cuadrado 30mmx2mm (60")	1	92.09	92.09
		Total:	368.17



Figura 26 Sistema total implementado.

3.7. Programación del agente

Para la programación del agente, fue utilizado Eclipse. Este es un entorno de desarrollo integrado (IDE) usado en la programación de computadoras, especialmente para Java. También contiene varios plug-ins con el fin de programar en otros idiomas como C ++, Python, PHP, Ruby y demás. En este caso, es necesario contar con el siguiente software:

- Eclipse IDE para C / C ++ Desarrolladores
- AVR GCC toolchain
- AVR Eclipse Plugin

Es necesario configurar previamente algunas características del microcontrolador antes de programar el comportamiento.

Dentro de esta configuración inicial se añade la biblioteca necesaria para entender y compilar símbolos AVR. Se debe declarar la frecuencia del microcontrolador; en este caso, funcionará a 1 MHz. Con esta frecuencia, el microcontrolador consumirá el mínimo de energía. Además no necesita circuito externo, ya que ocupa el oscilador interno RC.

Después de eso, se declaran dos macros con el fin de simplificar la operación de asignación de bits (no se incluye en la biblioteca AVR). Por último, el ADC es habilitado y se lo configura para funcionar con referencia externa a la frecuencia máxima del controlador.

Una vez configurado, se procede al programa. El código resulta sencillo. El agente sólo reaccionará cuando se detecte una colisión. Para este proyecto, se establece un giro, ya sea derecha o izquierda, y un movimiento hacia adelante como reacción a la colisión. Las funciones implementadas permiten organizar el código de una manera lógica mediante un estilo de programación estructurada. Se implementan funciones de encendido y apagado del robot para mejorar la interacción con el usuario.

Es necesario implementar una función para identificar y ejecutar el comportamiento que se requiera implementar, ya que este será codificado para su posterior optimización. Se implementó además una función adicional que siempre está leyendo las entradas analógicas y actualiza los valores de las variables de los motores. Esta función implementa una etapa de filtrado, que nos ayudará a eliminar el ruido presente en la señal. El código completo se muestra en el Anexo B.

CAPÍTULO IV

IMPLEMENTACIÓN DEL SOFTWARE

4.1. Introducción

El software de este proyecto consta de dos partes:

- Software de simulación
- Software de optimización

Para el software de simulación, se diseñó y programó un simulador, acorde a las características físicas del enjambre construido. Este simulador generará los parámetros de evaluación necesarios para verificar el desempeño del enjambre.

Con respecto al software de optimización, se diseñó y programó un algoritmo evolutivo basado en las métricas de un algoritmo genético, para optimizar fuera de línea (sin implementación física) el comportamiento del enjambre. Para esto, el algoritmo toma como parámetro de evaluación (función de aptitud) los valores devueltos por el simulador diseñado. Este comportamiento optimizado fue implementado posteriormente en el sistema físico para validar y concluir el proyecto.

Todo el software se desarrolló en Processing. Es un lenguaje de programación de código abierto y entorno de desarrollo integrado (IDE) enfocado en las artes electrónicas, y las comunidades de diseño visual con el propósito de enseñar los fundamentos de la programación informática en un contexto visual, y servir como el fundamento de cuadernos electrónicos.

4.2. Software de Simulación

4.2.1. Consideraciones físicas

Para poder diseñar un software de simulación adecuado para el sistema físico creado, es necesario analizar la interacción física existente en el mismo. En el caso del agente, no posee algún sistema de retroalimentación que le indique si efectivamente se mueve en la dirección deseada. Esto ocasiona varias dificultades como que los movimientos de rotación no centrados resultan impredecibles. Además, ligeros cambios en la fricción entre el agente y el terreno ocasionan cambios abruptos de dirección. Debido a configuración física del agente, al moverse hacia atrás se pierde momentáneamente el contacto entre el agente y las placas pudiendo apagarse

súbitamente el robot. Por otro lado, los obstáculos resultan tan livianos que no son detectables para el agente. Todos estos problemas fueron resueltos tomando las siguientes consideraciones:

- Los agentes deberán ser ajustados individualmente, tanto en hardware como en software, para poder tener un movimiento rectilíneo confiable.
- Los agentes solo podrán realizar giros centrados en sí mismos, y moverse frontalmente. Otro tipo de movimiento no está considerado.
- El obstáculo será un objeto pasivo en la simulación, es decir únicamente será desplazado por los agentes mas no intervendrá en su comportamiento.

Adicionalmente, son necesarias de definir unas reglas físicas básicas para el movimiento del agente. Estas son implementadas en el programa a través de bloques `if-else` y `do-while`. Todas las posibles interacciones pueden observarse en la Figura 27.

Interacción Agente-Entorno (Figura 27. a,b)

- El agente posee una posición y velocidad inicial
- El agente no puede atravesar paredes ni la columna central
- Cualquier contacto del agente, ya sea con las paredes o con la columna central, disparará el comportamiento, lo cual implica un cambio de dirección (giro).

Interacción Agente-Agente (Figura 27. c)

- El agente no puede atravesar otros agentes.
- Cualquier contacto entre agentes disparará el comportamiento (giro).

Interacción Agente-Obstáculo (Figura 27. d)

- El obstáculo se moverá cuando sea tocado por algún agente. Se moverá en la dirección indicada por la línea que une los centros del agente y el obstáculo durante la colisión, alejándose de ella.
- El obstáculo no afecta el comportamiento del agente.

Interacción Obstáculo-Obstáculo (Figura 27. e)

- El obstáculo se moverá cuando sea tocado por otro obstáculo. Se moverá en la dirección indicada por la línea que une los centros de ambos obstáculos durante la colisión, alejándose de ella.

Interacción Obstáculo-Entorno (Figura 27. f, g)

- El obstáculo tiene posición inicial
- El obstáculo no se mueve por sí solo.
- El obstáculo no puede atravesar paredes ni la columna central
- Cualquier contacto del obstáculo, ya sea con las paredes o con la columna central, cambiará la dirección del mismo según el ángulo de contacto. La pared ejerce una fuerza perpendicular a la misma.

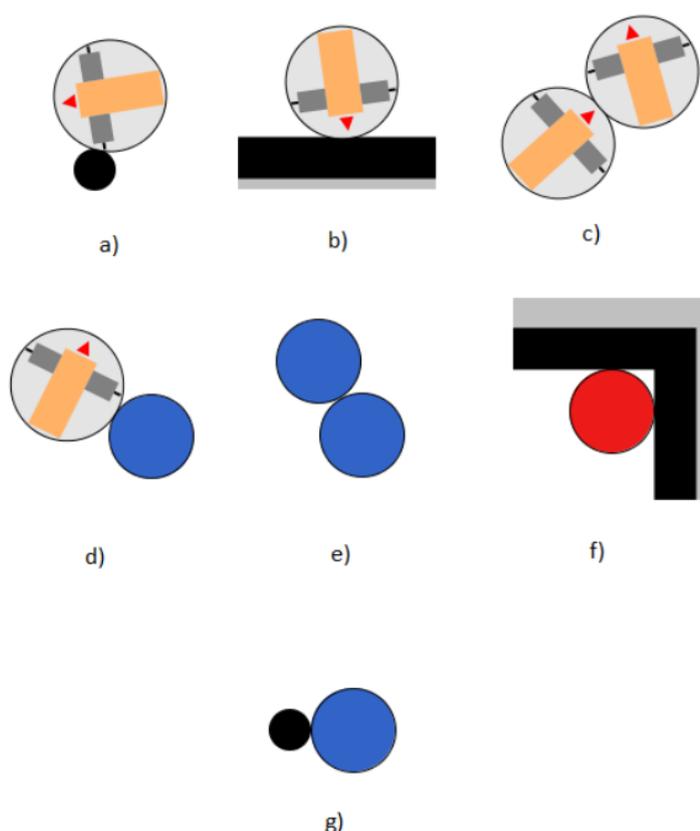


Figura 27 Posibles interacciones en la simulación.

4.2.2. Simulación del movimiento

Para la representación del movimiento dentro de la simulación, se programa un bucle infinito que constantemente está recalculando las posiciones de cada componente simulado y dibujándolos a una tasa de refrescamiento específica, de tal forma que se obtenga resultados similares en las velocidades reales y simuladas de los robots y obstáculos.

4.2.2.1. Agente

Para simular el movimiento del agente, fue utilizado un sistema vectorial de posición y velocidad. La velocidad está representada por una magnitud vectorial que se añade a la posición actual del agente en cada ejecución del lazo de movimiento. De igual forma, constantemente se redibuja al agente en su nueva posición actual rotado en dirección de la velocidad. Para realizar los giros en el propio eje, se modifica la dirección de la velocidad paulatinamente desde el ángulo inicial hasta el ángulo deseado, y mientras gira el agente, la velocidad no influye a la posición actual.

Los cuadros por segundo (fps) y la magnitud de la velocidad fueron calculados para que efectivamente el agente tenga una rapidez de 1cm/s (rapidez real) en la simulación.

4.2.2.2. Obstáculo

De igual forma, el movimiento del obstáculo está definido por un sistema vectorial con posición y velocidad. La velocidad afectará a la posición del obstáculo siempre y cuando exista un agente o una pared que esté en contacto con el mismo. En cualquier otro caso, la velocidad será nula.

4.2.3. Evaluación del enjambre

Para evaluar de forma precisa el enjambre es necesario analizar la tarea escogida. En este caso, la limpieza del área es el objetivo. Para evaluar de forma imparcial este objetivo es necesario considerar:

- El concepto de limpieza para esta tarea consiste en mover los obstáculos fuera del área. Teniendo en cuenta que en este caso el área está limitada por un borde, mientras más cercanos al borde estén los obstáculos, más limpia estará el área.
- Tener una posición inicial tanto para obstáculos como para agentes. De esta forma se establece un punto de referencia inicial.
- Definir un tiempo de ejecución de la simulación.

Para aplicar el concepto de limpieza utilizaremos la columna central como punto de referencia. Se medirá la distancia entre ese punto y los obstáculos. La sumatoria de todas estas distancias será el parámetro de evaluación (Figura 28).

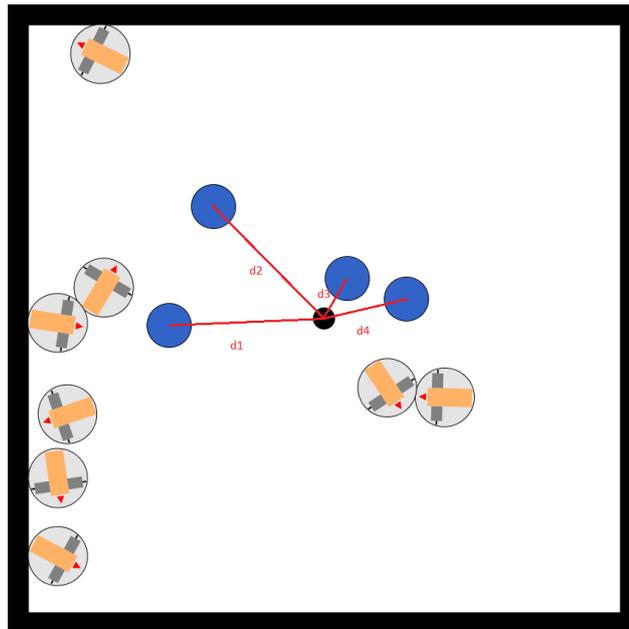


Figura 28 Parámetro de evaluación.

En la posición inicial del sistema los obstáculos quedarán lo más cercanos posibles a la columna central, y los agentes, ubicados a 30 cm de la misma. Con esta configuración podrá evidenciarse claramente el desplazamiento final de los obstáculos. (Figura 29)

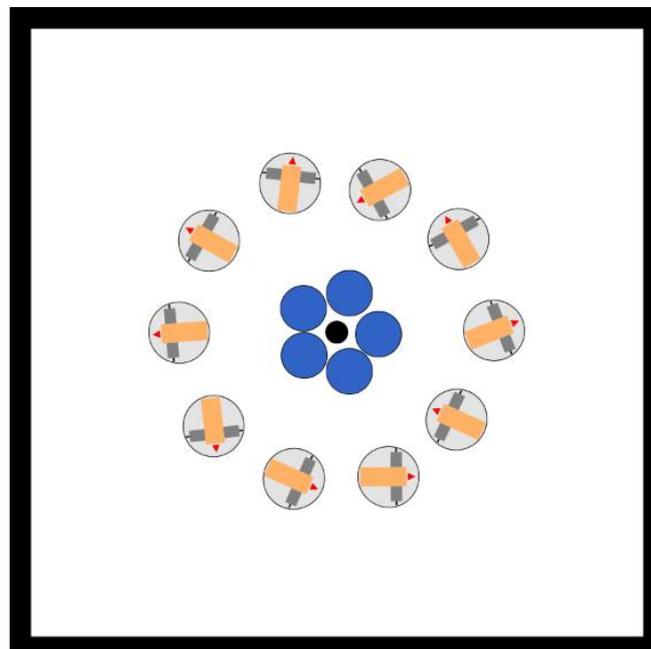


Figura 29 Posición inicial.

Una vez implementado el sistema físico, se evidenció que en el área de trabajo, la utilización de 10 agentes es contraproducente como se puede observar en la Figura 30. Los resultados obtenidos difieren mucho entre sí, por lo que se toma la decisión de reducir la cantidad de agentes a 8. De igual forma, se utilizará un obstáculo por cada 2 agentes, por lo que se utilizarán 4 obstáculos. Si tomamos a consideración la posición inicial con 4 obstáculos, la distancia total inicial entre obstáculos y columna central es 120 mm.

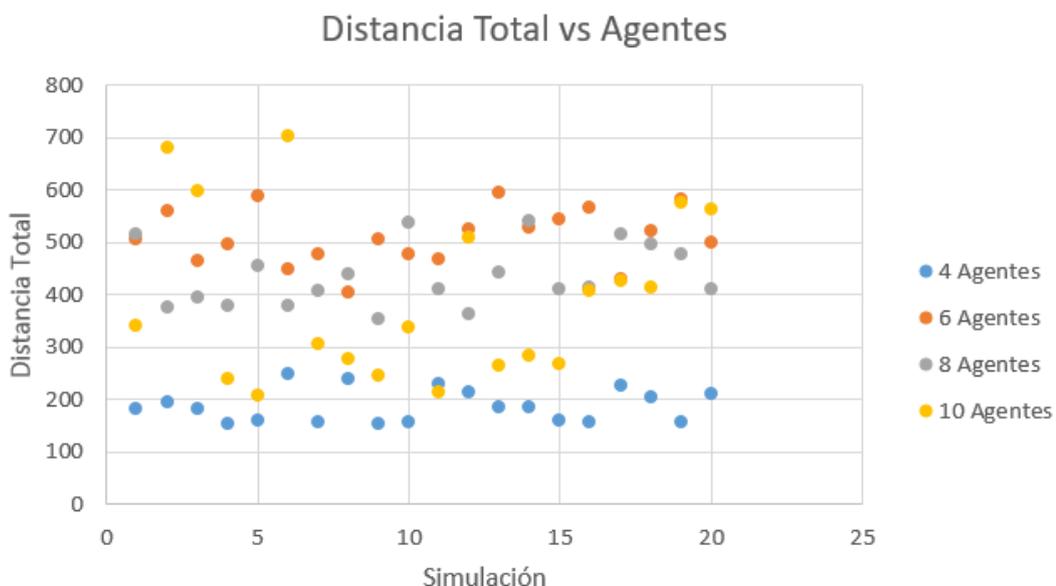


Figura 30 Eficiencia del enjambre según número de agentes.

Como último punto, el tiempo de simulación resulta crucial. Al ser un entorno limitado, eventualmente cualquier movimiento podría terminar solucionando el problema. De forma experimental se establece el tiempo de simulación en 120 segundos, siendo el más óptimo según la Figura 31.

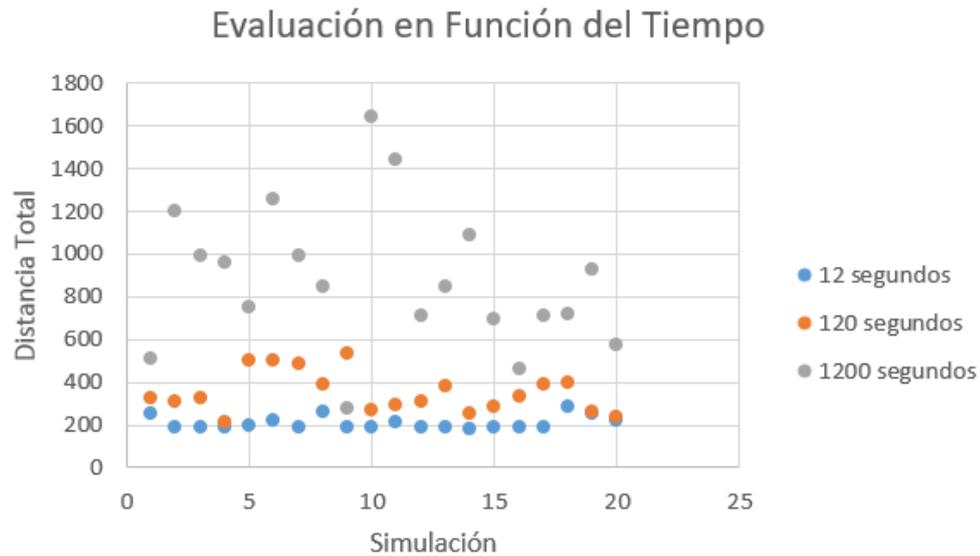


Figura 31 Evaluación en función del tiempo.

4.3. Software de Optimización

Para optimizar el comportamiento del enjambre en la tarea de limpieza del área, se implementó un algoritmo evolutivo con métricas genéticas. Este algoritmo está programado en Processing y toma como parámetros de optimización, los valores obtenidos en la simulación. Por esto se establece un entrenamiento fuera de línea del sistema.

4.3.1. Diseño del algoritmo

El diseño del algoritmo se basa en la correcta selección de parámetros y variables para obtener un buen desempeño en la tarea de optimización asignada. Todos los parámetros que fueron tomados en consideración son explicados en las siguientes subsecciones. Estos son:

- Problema de optimización
- Representación genética
- Función de aptitud
- Población e inicialización
- Selección
- Operadores genéticos
- Finalización

4.3.1.1. Problema de optimización

El problema se puede resumir en la siguiente declaración: Encontrar el comportamiento más óptimo para cada agente con el fin de maximizar la limpieza de la zona.

Para lograr este objetivo es necesario tener una forma de medir la limpieza. En este enfoque, la limpieza se mide por la suma de todas las distancias entre la posición final de cada obstáculo y el centro de la zona después de cada simulación de un enjambre que ha implementado un comportamiento específico.

Como puede verse, un algoritmo genético es aplicable a este problema; hay una búsqueda de optimización y una forma de medirlo. Este problema no tiene una solución matemática específica por lo que es necesario un método heurístico.

Para utilizar un algoritmo genético, es necesario codificar el comportamiento como un gen, y una forma de medir la efectividad del mismo.

4.3.1.2. Representación genética

El movimiento real de cada agente es muy complejo, por lo que, para este primer enfoque, es limitado tanto en la simulación como en la implementación física. El gen está formado por 9 bits. En la Tabla 6 se detalla el significado de cada bit.

Tabla 6
Representación genética del comportamiento

Bit	0	1	2	3	4	5	6	7	8
Función	Sentido de giro	Ángulo inferior			Ángulo superior				
Valor	0: Derecha 1: Izquierda	Valor decimal*24°			Valor decimal*24°				

4.3.1.3. Función de aptitud

La forma de evaluar la limpieza del área está dada por la suma de todas las distancias entre la posición final de cada obstáculo y el centro de la zona después de cada simulación de un enjambre que ha implementado un comportamiento específico. Es importante tener en cuenta, debido a la aleatoriedad, incluso con el mismo

comportamiento, la medida de la limpieza es diferente cada vez debido a lo aleatoriedad intrínseca del sistema.

Para garantizar la aplicabilidad de esta medida, se ejecutará varias veces el mismo comportamiento y se promediará todas las distancias para obtener el valor resultante. Esto se utiliza como la función de aptitud para evaluar un gen.

El número de simulaciones necesarias para obtener un resultado de confianza se selecciona de forma experimental. Como podemos ver en la Figura 32, no existe una variación significativa al variar las simulaciones en un rango de 10 a 80 simulaciones. Serán utilizadas 20 simulaciones para este caso.

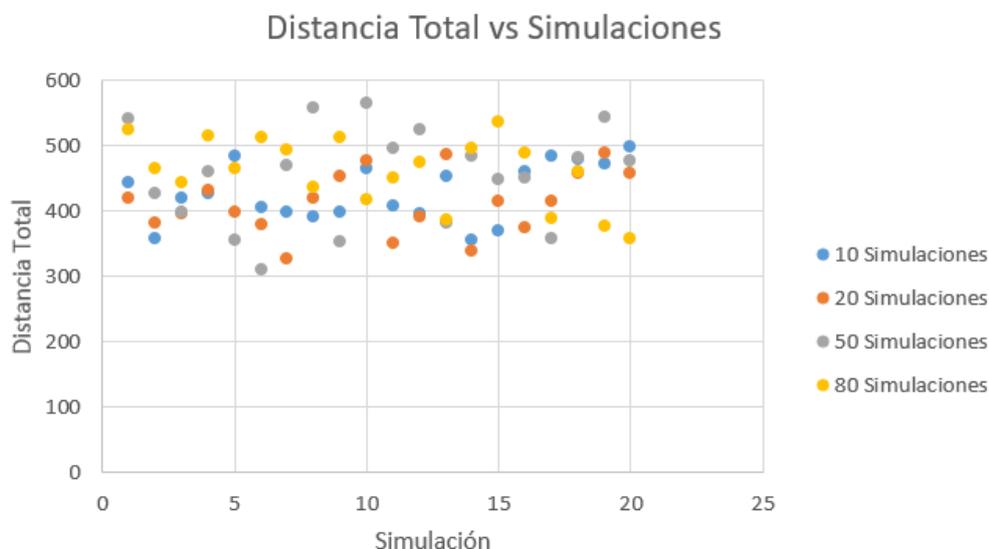


Figura 32 Distancia total según el número de simulaciones promediadas.

4.3.1.4. Población e inicialización

La población inicial se selecciona al azar. Luego, el algoritmo evalúa cada miembro de la población para tener valores iniciales de aptitud.

Para el tamaño de la población, si hay muy pocos genes, el algoritmo tiene pocas opciones para realizar una combinación y sólo se explora una pequeña parte del espacio de búsqueda. Por otro lado, si hay demasiados genes, el algoritmo se ralentiza. La investigación muestra que después de un límite (que depende principalmente de la codificación y el problema) no es útil aumentar el tamaño de la población, ya que no hace la solución del problema más rápido. Teniendo en cuenta que con 9 bits para la

codificación del gen, se puede tener 512 combinaciones. Se utilizará 25 genes como población, esto representa aproximadamente el 5% de todas las posibles soluciones.

4.3.1.5. Selección

La función de selección para este caso se explica a continuación:

1. Evaluar todos los individuos (genes) de la población.
2. Ordenar los genes de acuerdo con la función de aptitud. Los genes se encuentran organizados dentro de un arreglo, siendo los más aptos, los que poseen un índice más alto.
3. Evaluar una función de selección para encontrar el índice de los padres. Esta función tiende a seleccionar los índices más altos.

La función de selección es:

$$index = floor(population.length * random(0,1)^2)$$

La función se aplicó a una población de 100 genes, y consiguió 100 valores. El histograma resultante se muestra en la Figura 33.

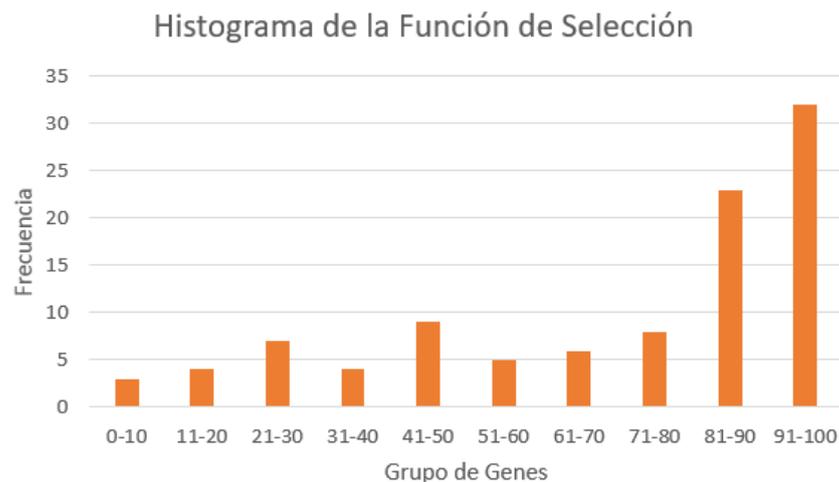


Figura 33 Histograma función de selección.

4.3.1.6. Operadores genéticos

Combinación: Es un operador genético utilizado para variar la programación de uno o más genes de una generación a la siguiente. Es análogo a la reproducción y cruce biológico, sobre la cual se basan los algoritmos genéticos. La combinación es un

proceso de tomar secciones genéticas de varios padres y producir un solo niño de ellos. Hay métodos para la selección de los genes. El operador de combinación utilizado en este caso es la siguiente.

Para este proyecto, se establece una combinación estándar entre los 2 genes (padres). En este caso, el niño recibirá una sección genética al azar (50%) de cualquiera de sus padres. Una sección genética completa es un grupo de bits con una función específica. Como se explicó anteriormente, en este caso, hay 3 secciones genéticas completas en cada gen.

Mutación: Es un operador genético utilizado para mantener la diversidad genética de una generación para la próxima. Es análogo a la mutación biológica. La mutación altera uno o más valores de los componentes en un gen de su estado inicial. Con la mutación, el gen puede cambiar por completo del gen anterior. De ahí que el algoritmo genético puede llegar a una mejor solución mediante el uso de mutación. La mutación se produce durante la evolución

Según diversos estudios, una probabilidad de mutación razonable debe ser baja. Si es demasiado alta, la búsqueda se convertirá en una búsqueda al azar primitiva.

Para este proyecto, el operador de mutación se aplica a cada bit en el gen. Tiene por defecto una muy baja probabilidad de que ocurra (5%).

4.3.1.7. Finalización

El algoritmo se repite hasta que se alcanza una condición de terminación. Esta condición puede ser:

- Una solución específica satisface el criterio de parada.
- Se alcanza un número predeterminado de generaciones.
- Se alcanza un límite de recursos (tiempo/dinero)
- Convergencia de las soluciones encontradas
- La inspección manual
- Combinaciones de las anteriores

Para este caso, se utiliza el criterio de convergencia. En algún momento, la mayoría de los genes dentro de la población serán el mismo, convergiendo en el mejor

resultado. Este parámetro se define al 80% de la convergencia, es decir cuando existan 20 de 25 genes iguales.

4.3.2. Resumen de características

En la sección 4.3.1, se analizan y establecen una serie de parámetros para el algoritmo evolutivo, acorde a las características de este proyecto. Los parámetros seleccionados permiten la convergencia del algoritmo como se observa en la sección 5.2. Es necesario mencionar que los parámetros fueron escogidos a criterio del investigador analizando las características del enjambre físico y la simulación.

Tabla 7
Resumen de parámetros del algoritmo

Parámetro	Valor
Número de simulaciones a promediar	20 simulaciones
Tiempo de simulación	120 segundos
Agentes	8 agentes
Obstáculos	4 obstáculos
Representación genética	9 bits (512 combinaciones)
Población genética	25 genes
Combinación	Aleatoria 50%
Número de padres	2
Probabilidad de mutación	5%
Convergencia	80%

CAPÍTULO V

EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS

5.1. Introducción

Los experimentos a desarrollarse tendrán por objetivo verificar la convergencia real del algoritmo evolutivo y posteriormente comparar los resultados obtenidos en la simulación con los resultados obtenidos al implementarlos en el enjambre robótico construido.

Las pruebas se dividirán en 3 etapas:

- Pruebas del algoritmo de optimización: Con los valores estándar de los parámetros, definidos en el capítulo 4, se ejecutará varias veces el algoritmo y se comprobará si efectivamente converge la solución obtenida.
- Pruebas de simulación: Serán implementados en el enjambre simulado los resultados obtenidos por optimización, se tomarán datos y se procederá a realizar las comparaciones para validar el modelo.
- Pruebas de implementación: Serán implementados en el enjambre real los resultados obtenidos por optimización, se tomarán datos y se procederá a realizar las comparaciones para concluir el proyecto.

Tanto para las pruebas de simulación como de implementación, debe ser establecido un protocolo imparcial que permita una comparación objetiva de los resultados a obtener. El protocolo será el siguiente.

- Obtener mediante la ejecución del algoritmo, el gen más apto para la tarea.
- Obtener mediante la ejecución del algoritmo, el gen menos apto para la tarea
- Simular el sistema con: Gen más apto, gen menos apto, gen aleatorio
- Implementar, ejecutar y registrar los datos de ejecución del enjambre real con el gen más apto implementado.
- Implementar, ejecutar y registrar los datos de ejecución del enjambre real con el gen menos apto implementado.
- Implementar, ejecutar y registrar los datos de ejecución del enjambre real con el mismo gen aleatorio previamente simulado.
- Comparar los datos obtenidos entre pares.

5.2. Análisis de resultados de optimización

Para comprobar la convergencia del algoritmo se procede a realizar 50 ejecuciones con los parámetros anteriormente resumidos (ver Tabla 7). Los resultados pueden observarse en la Figura 34 en donde se muestran los genes más aptos y cuantas veces fueron seleccionados por el algoritmo evolutivo. En este caso es posible observar la tendencia del algoritmo a converger sobre un grupo de comportamientos específicos, en lugar de una sola solución.

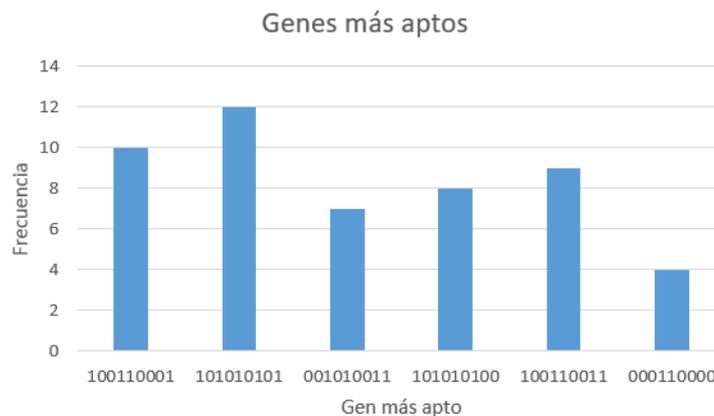


Figura 34 Genes más aptos.

Si se analiza los comportamientos óptimos, resultan 6 de 512 posibilidades (1.172%). Con este resultado se puede observar la convergencia del algoritmo, y resulta más notorio al momento de analizar el fenotipo de dichos genes. Esta traducción se presenta en la Tabla 8. El comportamiento óptimo del agente es independiente del sentido de giro, siempre y cuando el ángulo sea 72° o 105° .

Tabla 8
Comportamientos más óptimos

Gen	Comportamiento
100110001	Giro izquierda. 72°
101010101	Giro izquierda. 105°
001010011	Giro derecha. 105°
101010100	Giro izquierda. 105°
100110011	Giro izquierda. 72°
000110000	Giro derecha. 72°

Ahora, al analizar los resultados obtenidos por el algoritmo, mostrados en la Figura 35, la solución encontrada (20 de 25 genes) menciona el “Giro derecha. 72°”, sin embargo, los genes divergentes (5 de 25) corresponden a otras representaciones del mismo comportamiento “Giro derecha. 72°” o en su defecto otras soluciones de las anteriormente encontradas. Esto nos indica que a pesar de que la convergencia está predefinida al 80%, la solución óptima es encontrada con anterioridad.

GENE	FITNESS
000110010	740.758
000110010	741.704
000110010	741.994
000110011	743.897
000110010	755.654
100100011	757.055
000110010	762.542
000110010	763.522
100110011	775.267
000110010	783.209
000110010	787.831
000110011	788.905
000110010	793.045
000110010	795.544
001010101	796.247
000110010	801.592
000110010	805.660
000110010	811.372
000110010	813.888
000110010	814.518
000110010	815.486
000110010	822.816
000110010	842.564
000110010	843.504
000110010	885.171

Figura 35 Resultado de la ejecución de un algoritmo.

Esta hipótesis es confirmada al analizar los tiempos de convergencia mostrados en la Figura 36. En este caso es posible identificar cierta tendencia creciente hasta que el algoritmo alcanza aproximadamente el 60% de convergencia. A partir de este punto, la pendiente es casi nula, lo que lleva a una ralentización del algoritmo. Esto puede deberse a la dualidad de las soluciones (izquierda, derecha) e incluso a la existencia de más soluciones óptimas entre la población final.

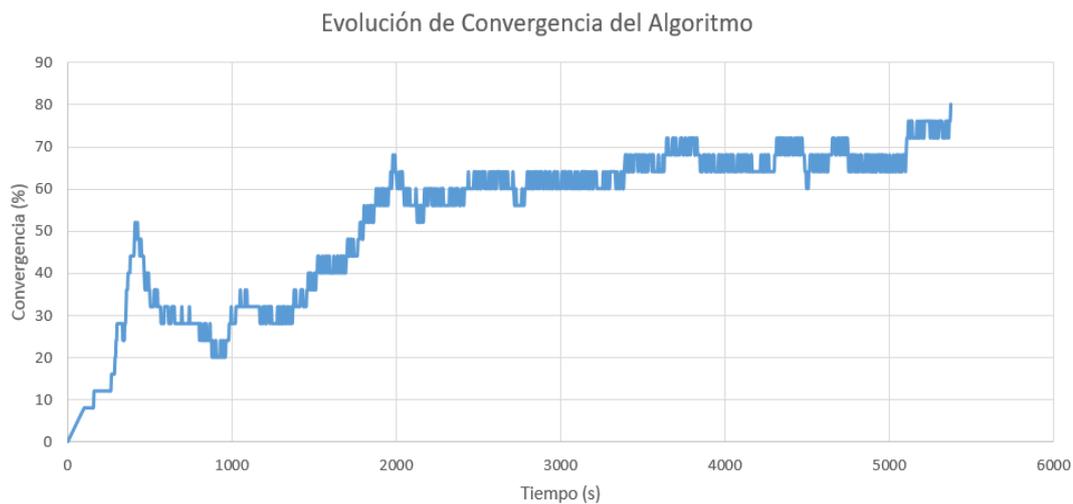


Figura 36 Convergencia del algoritmo.

5.3. Análisis de resultados de simulación

Al realizarse 50 ejecuciones del algoritmo, se obtuvieron los genes más aptos, presentados previamente en la Figura 34, y los genes menos aptos, presentados en la Figura 37 con su respectiva traducción en la Tabla 9.

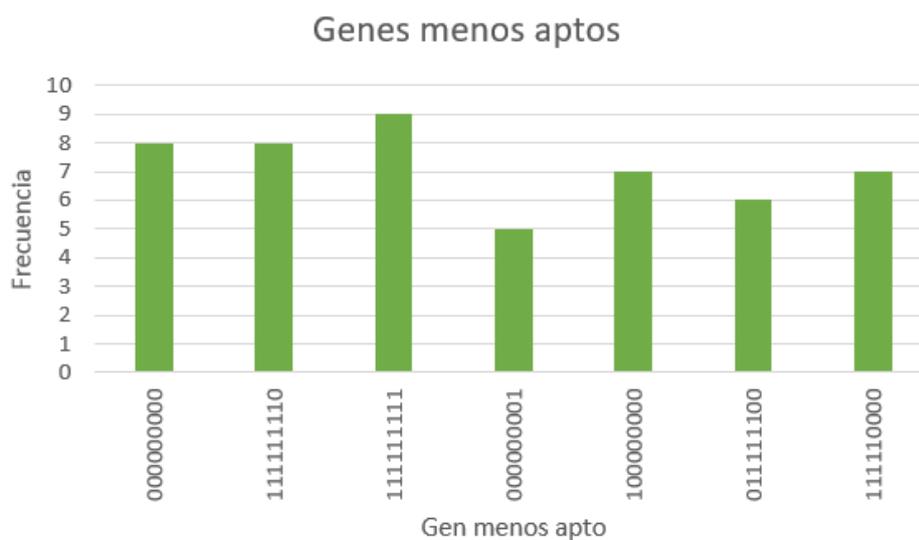


Figura 37 Genes menos aptos.

Tabla 9
Comportamientos menos óptimos

Gen	Comportamiento
000000000	Giro derecha. 0°
111111110	Giro izquierda. 360°
111111111	Giro izquierda. 360°
000000001	Giro derecha. 0°-24°
100000000	Giro izquierda. 0°
011111100	Giro derecha. 360°
111110000	Giro izquierda. 360°

Se procede a simular el enjambre con estos genes para obtener los datos de desempeño. Los resultados se muestran a continuación en la Figura 38. En esta primera validación, es evidente una superioridad del gen más apto con respecto al gen menos apto e incluso con un gen aleatorio en los 3 experimentos realizados.

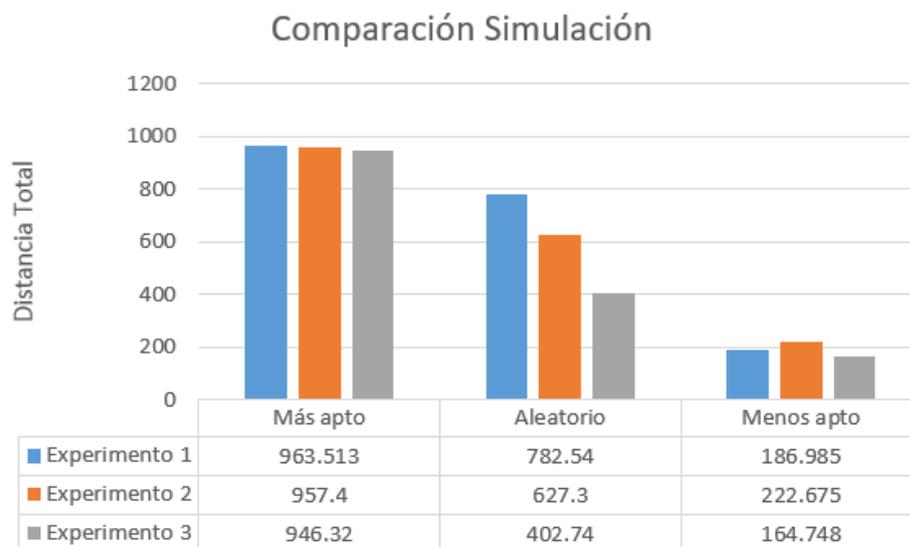


Figura 38 Comparación de resultados obtenidos en la simulación.

5.4. Análisis de resultados de implementación

Los resultados obtenidos de la optimización, serán implementados en el enjambre físico creado. De igual forma que con la experimentación en simulación, los resultados serán cuantificados, presentados y comparados. Para tener un punto inicial de comparación, las pruebas tanto en simulación como en la implementación física poseen el mismo estado inicial (Figura 39).

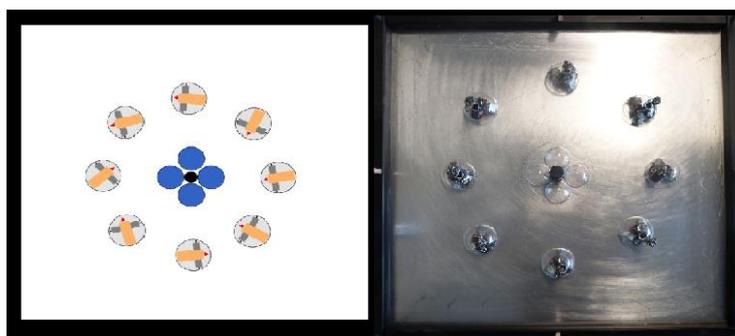


Figura 39 Posición inicial del enjambre.

De forma similar al comportamiento visto en el enjambre simulado, en los resultados de las implementaciones físicas, se puede evidenciar la clara superioridad del gen más apto con respecto al menos apto y a un gen aleatorio (Figura 40). De estos primeros resultados es posible validar el proceso de optimización y el software de simulación realizados.

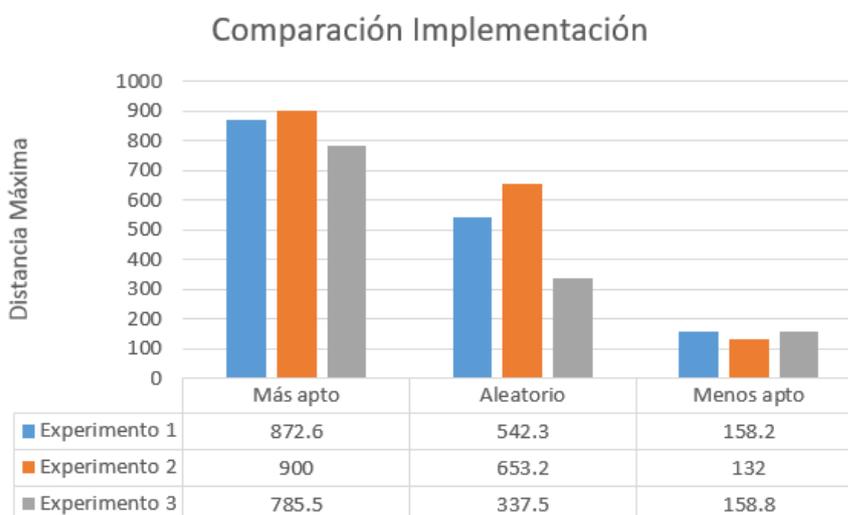


Figura 40 Comparación de los resultados obtenidos en la implementación física.

Adicionalmente a los cuadros comparativos presentados, se procedió a realizar comparaciones entre simulaciones e implementaciones tanto para los genes más aptos como para los menos aptos y el gen aleatorio. Se debe considerar que a pesar de que este gen fue elegido aleatoriamente, la implementación física y la simulación fueron realizadas con el mismo. Las siguientes figuras representan la comparación entre simulación e implementación física del gen más apto (Figura 41), gen aleatorio (Figura 42), y gen menos apto (Figura 43).

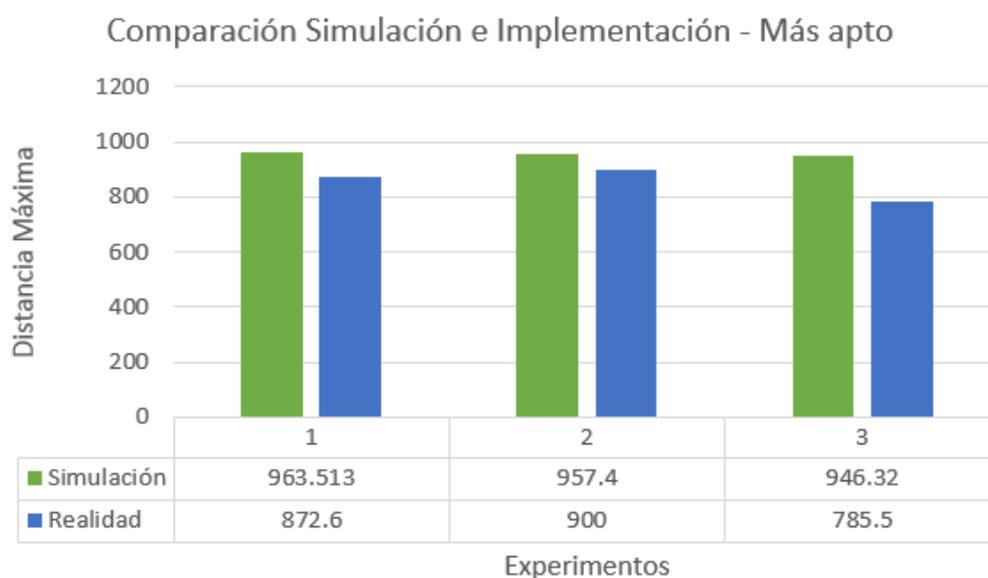


Figura 41 Comparación entre simulación e implementación con el gen más apto.

Para el caso de la comparación con el gen más apto, se puede evidenciar una mayor correlación entre los valores obtenidos. Los valores de desempeño tienden a ubicarse entre los 800 y 1000 (mm). Sin embargo al analizar los resultados en las comparaciones entre gen aleatorio (Figura 42), y entre gen menos apto (Figura 43), es muy claro que no existe correlación entre los valores.

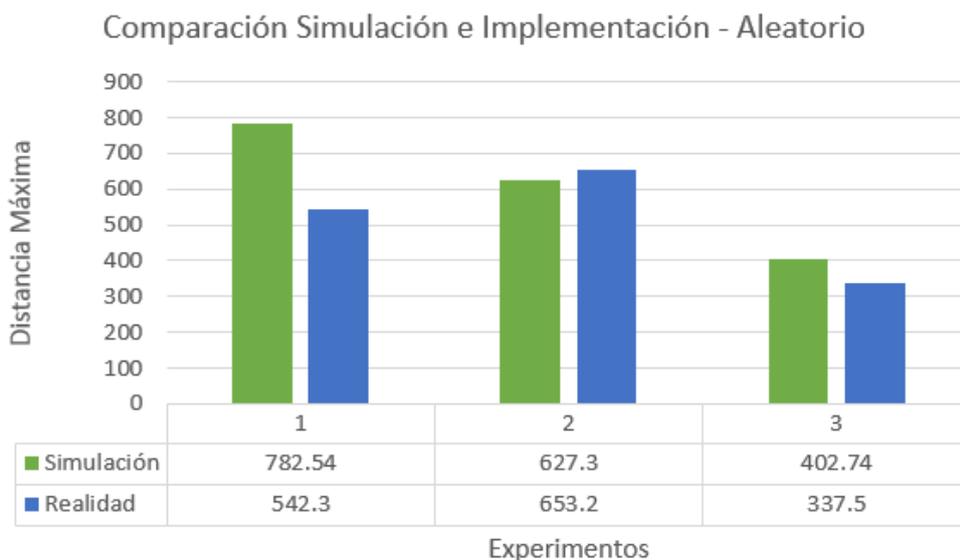


Figura 42 Comparación entre simulación e implementación con el gen aleatorio.

A pesar de que resultan numéricamente menores que los valores obtenidos en las comparaciones entre gen más apto, no existe otro punto de similitud, lo que lleva a pensar en la idea de que el software de simulación no corresponde en su totalidad al sistema físico implementado.

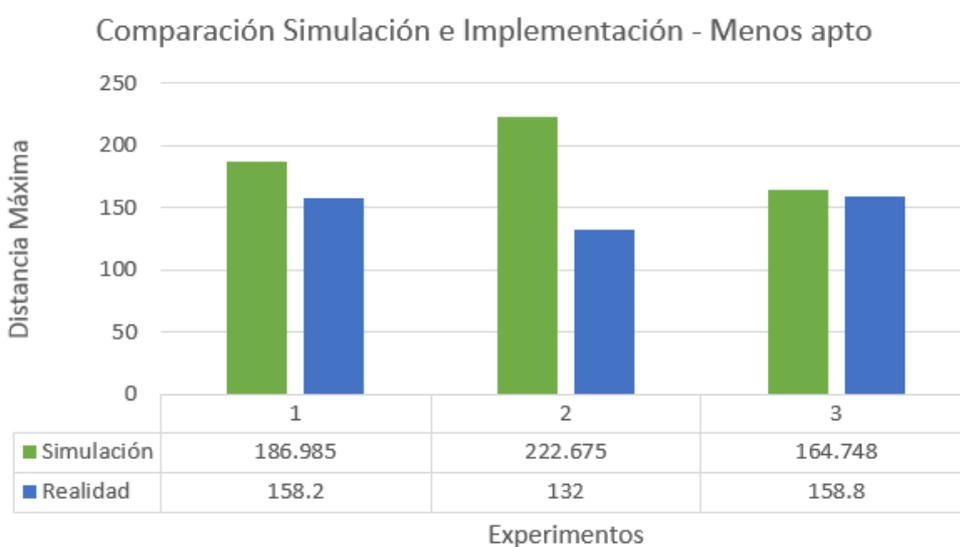


Figura 43 Comparación entre simulación e implementación con el gen menos apto.

Esto puede explicarse debido a diversos motivos. En primer lugar, la simulación idealizaba el movimiento y comportamiento del agente, ya que al ser un entorno creado matemáticamente, los giros son precisos, los movimientos exactos y no existen factores externos que afecten a los resultados. Por el otro lado, en la implementación física, como se supo mencionar en los apartados anteriores, existieron una serie de dificultades como la inexactitud de los movimientos del agente debido a su misma concepción sencilla a nivel de hardware. Adicionalmente factores externos como la fricción entre agente y terreno, o las momentáneas desconexiones entre el agente y las placas, provocaron un comportamiento mucho más complejo del que se logró simular

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se diseñó e implementó un enjambre robótico apto para la limpieza de un área, y se optimizó su comportamiento basándose en los resultados obtenidos en el software de simulación creado. Esta optimización utilizó algoritmos evolutivos con métricas genéticas para identificar el mejor comportamiento de un conjunto de 512 comportamientos posibles. Cada comportamiento fue evaluado mediante una función de aptitud definida en la simulación creada, la cual implementó todas las posibles interacciones entre agente-entorno-obstáculo.

En base a los resultados de la optimización, se puede evidenciar claramente la convergencia del algoritmo en un conjunto de 6 resultados (genotipos), los cuales al traducirlos, generan 4 comportamientos posibles (fenotipos). Teniendo en cuenta que existen 512 posibles combinaciones en el espacio de soluciones para este problema, se puede concluir que efectivamente el problema es optimizable, basándose en los parámetros de evaluación escogidos.

El análisis de los fenotipos seleccionados como los más aptos, permite identificar un comportamiento específico para optimizar la limpieza del área en un tiempo de 2 minutos. Dicho comportamiento es independiente del sentido de giro, siempre y cuando el agente gire sobre su propio eje, ya sea un ángulo de 72° o 105° , después de detectar una colisión. Por otro lado, al analizar los comportamientos más ineficientes, se puede ver que efectivamente son los que no reaccionan (giros de 0°), o realizan giros de 360° sobre su propio eje al detectar colisiones.

Al observar los resultados presentados en la Figura 38 y en la Figura 40, se verifica que el comportamiento seleccionado como más óptimo es superior, en cuanto a desempeño en limpieza del área, que un comportamiento aleatorio, tanto en la simulación como en la implementación física. De igual manera, el comportamiento identificado como más ineficiente, muestra los peores desempeños en limpieza del área comparado con un comportamiento aleatorio y el comportamiento optimizado. Esto permite validar el proceso y metodología utilizados para este proyecto.

La representación genética seleccionada para este proyecto resultó suficiente para la tarea seleccionada y se obtuvieron resultados positivos en los experimentos realizados, sin embargo, no aprovecha al máximo la capacidad de optimización del algoritmo genético implementado dado que un espacio de 512 posibles soluciones puede ser analizado con otros métodos más sencillos, o incluso, probada en su totalidad sin necesidad de algoritmos. Esto resulta una consecuencia de limitar el movimiento de los agentes, que por un lado facilitó el análisis cinemático, pero por otro simplificó demasiado el problema y posibles soluciones.

Por otro lado, fue comprobada y validada la premisa del agente más sencillo posible, dado que el agente implementado resolvió de manera óptima la tarea propuesta. Se comprobó que no es necesario un sistema de comunicación para efectuar la limpieza de un área, y es posible realizarlo de forma óptima, implementando un sistema de movimiento omnidireccional y un sistema simple de detección de colisiones.

Con respecto a la simulación se evidenció diferencias con la implementación física creada como puede observarse en las comparaciones presentadas en la Figura 41, Figura 42 y Figura 43. Se concluye que la simulación presentó un entorno ideal de trabajo en donde no existe fricción ni problemas de alimentación con los agentes, siendo estos los problemas más evidentes en el enjambre físico, los cuales modificaban el comportamiento real. Sin embargo, los resultados obtenidos con la simulación reflejan una buena aproximación de los resultados reales, en esta tarea en específico, permitiendo una optimización fuera de línea aplicable al enjambre físico y obteniendo un desempeño similar al de la simulación.

El algoritmo evolutivo empleado resultó eficiente, logrando converger en un conjunto de soluciones, con una serie de parámetros específicos, pero resulta muy sensible a la variación de los mismos, llegando incluso a la divergencia en algunos casos. Esto impide la experimentación más a fondo con la optimización, y se debe principalmente a la forma de codificar los comportamientos, que permite la existencia de ambigüedades en su interpretación.

6.2. Recomendaciones

Se recomienda, para un trabajo futuro, la completa simulación de los movimientos posibles por el agente, de esta manera será posible tener más opciones en cuanto al comportamiento y se puede explotar al máximo el algoritmo evolutivo.

Es necesario añadir modelos de fricción al movimiento del agente ya que este parámetro resultó crucial en las diferencias entre los resultados obtenidos por simulación y en la implementación física.

Puede ser posible la experimentación con más variaciones en los parámetros del algoritmo genético. Esto puede conducir a una mejor convergencia para el problema además de reforzar los resultados. Si son modificados otros parámetros como la representación genética, es posible que el grupo de comportamientos óptimos se reduzca, encontrando efectivamente una solución única al problema.

Al retirar las limitaciones de movimiento al agente, se puede optar por distintas representaciones genéticas para los comportamientos, pudiendo aumentar exponencialmente las posibles soluciones y justificar así el uso de un algoritmo evolutivo como método de optimización.

Se recomienda trabajar con otras formas de simulación más sencillas, como masas puntuales, para implementar algoritmos de navegación más complejos y no comprometer el desempeño del sistema en general. Además es posible extender las capacidades de los agentes para experimentar con tareas más complejas, a la vez que se desarrolla una plataforma robusta de análisis para sistemas de enjambres robóticos. Pueden implementarse sistemas de comunicación o incluso mejorar el algoritmo de movimiento del agente mediante algoritmos adaptativos.

Para experimentos más complejos resulta crucial cambiar la forma de alimentación del enjambre ya que este entorno de placas polarizadas, limita mucho el desenvolvimiento de los agentes. Otro tipo de tareas resulta imposible de desarrollarse en este tipo de entornos controlados. Es necesario buscar un punto de equilibrio entre costos del agente y capacidades del mismo.

BIBLIOGRAFÍA

- Altshuler, Y., Bruckstein, A. M., & Wagner, I. A. (2005). Swarm robotics for a dynamic cleaning problem. *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, (págs. 209-216).
- Balch, T. (2004). Communication, diversity and learning: Cornerstones of swarm behavior. En *Swarm Robotics* (págs. 21-30). Springer.
- Barash, D. (1999). A genetic search in policy space for solving Markov decision processes. *AAAI Spring Symposium on Search Techniques for Problem Solving under Uncertainty and Incomplete Information*.
- Beni, G. (2004). From swarm intelligence to swarm robotics. *International Workshop on Swarm Robotics*, (págs. 1-9).
- Blum, C., & Li, X. (2008). Swarm intelligence in optimization. En *Swarm Intelligence* (págs. 43-85). Springer.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 1-41.
- Chang, H. S., Lee, H.-G., Fu, M. C., & Marcus, S. I. (2005). Evolutionary policy iteration for solving Markov decision processes. *IEEE Transactions on Automatic Control*, 50(11), 1804-1808.
- Das, P. K., Behera, H. S., & Panigrahi, B. K. (2016). A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation*, 28, 14-28.
- De Silva, V., Ghrist, R., & Muhammad, A. (2005). Blind Swarms for Coverage in 2-D. *Robotics: Science and Systems*, (págs. 335-342).
- Dorigo, M. (2005). SWARM-BOT: An experiment in swarm robotics. *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, (págs. 192-200).
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., & Christensen, A. L. (2016). Evolution of collective behaviors for a real swarm of aquatic surface robots. *PloS one*, 11(3), e0151834.

- Dudek, G., Jenkin, M., Milios, E., & Wilkes, D. (1993). A taxonomy for swarm robots. *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on, 1*, págs. 441-447.
- Fonseca, C. M., & Fleming, P. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 1-16.
- Gbenga, D. E., & Ramlan, E. I. (2016). Understanding the Limitations of Particle Swarm Algorithm for Dynamic Optimization Tasks: A Survey Towards the Singularity of PSO for Swarm Robotic Applications. *ACM Computing Surveys (CSUR)*, 49(1), 8.
- Iima, H., & Kuroe, Y. (2013). Swarm Reinforcement Learning Method for a Multi-robot Formation Problem. *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, (págs. 2298-2303).
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 237-285.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Kong, C. S., Peng, N. A., & Rekleitis, I. (2006). Distributed coverage with multi-robot system. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, (págs. 2423-2429).
- Kumar, V., & Sahin, F. (2003). Cognitive maps in swarm robots for the mine detection application. *Systems, Man and Cybernetics, 2003. IEEE International Conference on, 4*, págs. 3364-3369.
- Lauer, M., & Riedmiller, M. (2000). An algorithm for distributed reinforcement learning in cooperative multi-agent systems. *In Proceedings of the Seventeenth International Conference on Machine Learning*.
- Nolfi, S., Bongard, J., Husbands, P., & Floreano, D. (2016). Evolutionary robotics. En *Springer Handbook of Robotics* (págs. 2035-2068). Springer.
- Nouyan, S., Campo, A., & Dorigo, M. (2008). Path formation in a robot swarm. *Swarm Intelligence*, 2(1), 1-23.

- Payton, D., Estkowski, R., & Howard, M. (2004). Pheromone robotics and the logic of virtual pheromones. *International Workshop on Swarm Robotics*, (págs. 45-57).
- Rothermich, J. A., Ecemis, M. I., & Gaudiano, P. (2004). Distributed localization and mapping with a robotic swarm. *International Workshop on Swarm Robotics*, (págs. 58-69).
- Sahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. *International workshop on swarm robotics*, (págs. 10-20).
- Seyfried, J., Szymanski, M., Bender, N., Estana, R., Thiel, M., & Worn, H. (2004). The I-SWARM project: Intelligent small world autonomous robots for micro-manipulation. *International Workshop on Swarm Robotics*, (págs. 70-83).
- Szymanski, M., Breitling, T., Seyfried, J., & Worn, H. (2006). Distributed shortest-path finding by a micro-robot swarm. En *Ant Colony Optimization and Swarm Intelligence* (págs. 404-411). Springer.
- Wagner, I. A., Altshuler, Y., Yanovski, V., & Bruckstein, A. M. (2008). Cooperative cleaners: A study in ant robotics. *The International Journal of Robotics Research*, 27(1), 127-151.
- Xu, W., & Chen, X. (2008). Artificial moment method for swarm robot formation control. *Science in China Series F: Information Sciences*, 51(10), 1521-1531.

ANEXOS