



**COMPARACIÓN DE ESTRATEGIAS DE NAVEGACIÓN
COLABORATIVA PARA ROBÓTICA MÓVIL**

**JOHN JAIRO MARTÍNEZ PUERTA
MARGARITA MARÍA VALLEJO JIMÉNEZ**

**UNIVERSIDAD AUTÓNOMA DE MANIZALES
MAESTRÍA EN MECATRÓNICA Y CONTROL
MANIZALES
(I COHORTE)
2015**

**ESTRATEGIAS DE NAVEGACIÓN COLABORATIVA
PARA ROBÓTICA MÓVIL**

**JOHN JAIRO MARTÍNEZ PUERTA
MARGARITA MARÍA VALLEJO JIMÉNEZ**

Informe Final para optar al título de Magister en Mecatrónica y Control

Directores

**Ph.D. Oscar Fernando Avilés, Universidad Militar Nueva Granada
M.Sc. Rubén Darío Flórez, Universidad Autónoma de Manizales**

**UNIVERSIDAD AUTÓNOMA DE MANIZALES
MAESTRÍA EN MECATRÓNICA Y CONTROL
MANIZALES
2015**

CONTENIDO

1	PRESENTACIÓN	11
2	RESUMEN	13
3	ÁREA PROBLEMÁTICA	15
3.1	DELIMITACIÓN DEL AREA PROBLEMÁTICA	15
3.2	FORMULACIÓN DEL PROBLEMA	16
3.3	ANTECEDENTES	16
3.4	ESTADO DEL ARTE	19
4	JUSTIFICACIÓN	21
5	OBJETIVOS	23
5.1	OBJETIVO GENERAL	21
5.2	OBJETIVOS ESPECÍFICOS	21
6	MARCO TEÓRICO	24
6.1	BIOMIMÉTICA	24
6.2	SISTEMA COMPLEJO	29
6.3	SISTEMAS BASADOS EN AGENTES (SBA)	32
6.4	INTELIGENCIA ARTIFICIAL DISTRIBUIDA (IAD)	34
6.5	ARQUITECTURAS DE CONTROL	42
6.6	ROBÓTICA MÓVIL	46
6.7	METAHEURÍSTICAS	51
6.8	ALGORITMO ACO (ANT COLONY OPTIMIZATION)	55
7	HIPOTESIS DE INVESTIGACION	61
8	OPERACIONALIZACION DE VARIABLES	62
8.1	VARIABLES INDEPENDIENTES	62
8.2	VARIABLES DEPENDIENTES	62
8.3	PRUEBAS ESTADÍSTICAS	63
9	METODOLOGÍA	65
9.1	METODOLOGÍA MAD-SMART	65
9.2	METODOLOGÍA QFD	91
9.3	METODOLOGÍA DE DISEÑO DE EXPERIMENTOS (DOE)	92
10	RESULTADOS	97
10.1	INTRODUCCIÓN	97
10.2	RESULTADOS ESPERADOS A PARTIR DE LOGRO DE LOS OBJETIVOS PLANTEADOS	97
10.3	ALCANCES Y LIMITACIONES	98
10.4	PRODUCTOS ESPERADOS	99
10.5	RESULTADOS DE LA IMPLEMENTACIÓN DE LOS ALGORITMOS	99
11	DISCUSION DE RESULTADOS	106
11.1	INTRODUCCIÓN	106

11.2	ANÁLISIS DE RESULTADOS	106
12	CONCLUSIONES Y RECOMENDACIONES	126
13	REFERENCIAS	128

LISTA DE FIGURAS

Figura 1	Temas tratados de Biomimética	24
Figura 2	Comportamientos Emergentes a partir de comportamientos sencillos.	29
Figura 3	Temas tratados de Sistema Complejo	30
Figura 4	Temas tratados de Sistemas basados en Agentes	32
Figura 5	Enfoques de la Inteligencia Artificial	34
Figura 6	Tipos de procesamiento característicos de la IA	35
Figura 7	Temas tratados en IA Distribuida.	36
Figura 8	Componentes de los entornos y sus características	37
Figura 9	Planeación y Cooperación en Sistemas Multiagentes Robóticos	39
Figura 10	Inteligencia de enjambre como estrategia de robótica colaborativa	41
Figura 11	Temas tratados en Arquitectura de Control.	42
Figura 12	Esquemas de procesamiento según los paradigmas de la robótica	43
Figura 13	Temas tratados de Robótica Móvil	46
Figura 14	Aproximación clásica de la Robótica	46
Figura 15	Modo general de un control robótico	47
Figura 16	Desempeño de sistemas de robots autónomos descentralizados	50
Figura 17	Temas tratados en Metaheurística	51
Figura 18	Temas tratados en Algoritmo ACO	55
Figura 19	Grafo que describe el camino que una hormiga debe tomar	58
Figura 20	Proceso de Análisis Estadístico recomendado	63
Figura 21	Pasos y Actividades de la Metodología MAD-Smart	66
Figura 22	Objetivos de la Estrategia de Enjambre	67
Figura 23	Objetivos de la Estrategia Multiagente	67
Figura 24	Definición de responsabilidades para la Estrategia de Enjambre	68
Figura 25	Definición de responsabilidades para la Estrategia Multiagente	68
Figura 26	Acciones y comportamientos de los robots para la Estrategia de Enjambre	69
Figura 27	Plantilla del Rol Explorador Reactivo de la Estrategia Enjambre	70
Figura 28	Plantilla del Rol Explorador Seguidor de la Estrategia Enjambre	70
Figura 29	Plantilla del Líder de Autónomos de la Estrategia Multiagentes	71
Figura 30	Plantilla del Líder de Servidores de la Estrategia Multiagentes	71
Figura 31	Plantilla del Rol Servidor de la Estrategia Multiagentes	72
Figura 32	Detalles del Agente Reactivo	72
Figura 33	Detalles del Agente Autónomo	73
Figura 34	Detalles del Agente Servidor	73
Figura 35	Tipos de entidades presentes en el entorno	74

Figura 36	Percepciones y acciones esperadas en el mapeo de entornos	75
Figura 37	Ontología para un sistema de navegación en enjambre y multiagente	75
Figura 38	Comunicación entre agentes según la estrategia implementada	78
Figura 39	Modelo de Conversaciones	79
Figura 40	Lista de software para simulación de sistemas robóticos	80
Figura 41	Relaciones de los Agentes ante los estímulos	82
Figura 42	Relaciones de los Agentes con el entorno	82
Figura 43	Características deseables en robótica de enjambre	83
Figura 44	Características de un sistema de Enjambre	84
Figura 45	Algoritmo de Enjambre implementado	84
Figura 46	Diagrama de flujo del algoritmo de Enjambre	86
Figura 47	Características de un Sistema Multiagente	87
Figura 48	Algoritmo Multiagente Implementado	87
Figura 49	Diagrama de Flujo del algoritmo Multiagente	89
Figura 50	Sistemas de la plataforma robótica propuesta	90
Figura 51	Subsistemas de la plataforma robótica a implementar	90
Figura 52	Diagrama de cubos para la Estrategia de Enjambre	94
Figura 53	Diagrama de cubos para la Estrategia Multiagente	95
Figura 54	Simulación de la Estrategia de Enjambre en NETLOGO	101
Figura 55	Llegadas al objetivo de la estrategia de enjambre en NETLOGO	101
Figura 56	Comportamiento de la estrategia de enjambre en 7 simulaciones	102
Figura 57	Comportamiento de la estrategia de enjambre en intervalo lineal	102
Figura 58	Comportamiento logarítmico de la estrategia de enjambre	103
Figura 59	Simulación de la Estrategia Multiagente en NETLOGO	103
Figura 60	Llegadas al objetivo de la estrategia multiagente en NETLOGO	104
Figura 61	Comportamiento de la estrategia multiagente en 7 simulaciones	104
Figura 62	Comportamiento de la estrategia multiagente en intervalo lineal	105
Figura 63	Comportamiento exponencial de la estrategia multiagente	105
Figura 64	Desempeño de la estrategia de enjambre en cada una de las 16 iteraciones.	107
Figura 65	Efecto de los factores en el tiempo de llegada en la Estrategia de Enjambre	107
Figura 66	Efectos independientes de cada factor en la Estrategia de Enjambre	108
Figura 67	Influencia de la sensibilidad al repulsor sobre sensibilidad al atractor	109
Figura 68	Influencia de la sensibilidad al repulsor sobre difusión de feromona	109
Figura 69	Influencia de la sensibilidad al repulsor sobre evaporación de feromona.	110
Figura 70	Influencia de la sensibilidad al atractor sobre difusión de feromona	110
Figura 71	Influencia de la sensibilidad al atractor sobre evaporación de feromona	111
Figura 72	Influencia de la difusión de feromona sobre evaporación de feromona	111
Figura 73	Desempeño de la estrategia multiagente en cada una de las 16 iteraciones.	112

Figura 74	Efecto de los factores en el tiempo de llegada en la Estrategia Multiagente.	113
Figura 75	Efectos independientes de cada factor en la Estrategia Multiagente	114
Figura 76	Influencia de la Sensibilidad Repulsor sobre Tasa de Autonomía	114
Figura 77	Influencia de Sensibilidad al repulsor sobre Sensibilidad a la luz	115
Figura 78	Influencia de Sensibilidad al repulsor sobre Sensibilidad al sonido	115
Figura 79	Influencia de Tasa de Autonomía sobre Sensibilidad a la luz	116
Figura 80	Influencia de Tasa de Autonomía sobre Sensibilidad al sonido	116
Figura 81	Influencia de Sensibilidad a la Luz sobre Sensibilidad al sonido	117
Figura 82	Diagrama de Caja y Bigotes para establecer la Eficiencia de las Estrategias	121
Figura 83	Diagrama de Caja y Bigotes para establecer la Consistencia de Enjambre	124
Figura 84	Diagrama de Caja y Bigotes para establecer la Consistencia de Multiagente	125
Figura 85	Entorno de simulación de NETLOGO	140
Figura 86	Robótica basada en Comportamientos	142
Figura 87	Aproximaciones de la Robótica Evolutiva	143
Figura 88	Sistemas para generar comportamientos en Robótica Evolutiva	143
Figura 89	Desempeño de las diferentes técnicas de Robótica	145
Figura 90	Comunicación entre sensores y motores en arquitectura de sistema nervioso	147
Figura 91	Inhibición del Sistema Instintivo por parte del Sistema Reflejo	148
Figura 92	Inhibición del Sistema Inteligente por parte del Sistema Reflejo	148
Figura 93	Inhibición del Sistema Reflejo por parte del Sistema Inteligente	149
Figura 94	Señales de inhibición entre Sistema Reflejo e Inteligente	149
Figura 95	Señales de inhibición presentes en la Arquitectura Propuesta	150
Figura 96	Organización de un Sistema Robótico a nivel de Módulos	153
Figura 97	Flujo de señales entre los subsistemas de la arquitectura propuesta.	156
Figura 98	Conexiones Braitenberg	157
Figura 99	Configuraciones cinemáticas para Robots Móviles con Ruedas RMR	158
Figura 100	Diagrama en bloques propuesto para un Microbot	161
Figura 101	Diagrama del Circuito del Sistema Reflejo	162
Figura 102	Diagrama del Circuito del Sistema Instintivo	163

LISTA DE TABLAS

Tabla 1	Configuración de Factores Estrategia de Enjambre y Multiagente	93
Tabla 2	Valores óptimos obtenidos para cada factor en ambas estrategias	94
Tabla 3	Parámetros iniciales para las simulaciones de ambas estrategias	100
Tabla 4	Factores óptimos obtenidos en la simulación para la Estrategia Enjambre.	118
Tabla 5	Factores óptimos obtenidos en la simulación para la Estrategia Multiagente	118
Tabla 6	Costo mínimo para ambas estrategias (Eficiencia)	119
Tabla 7	Estimador p del Test de NOVA para ambas estrategias	120
Tabla 8	Configuración de factores para el caso 1 de Consistencia: Multiagente	122
Tabla 9	Configuración de factores para el caso 1 de Consistencia: Enjambre	122
Tabla 10	Configuración de factores para el caso 2 de Consistencia: Enjambre	123
Tabla 11	Configuración de factores para el caso 2 de Consistencia: Multiagente	123
Tabla 12	Datos obtenidos del Test de ANOVA para evaluar Consistencia	124
Tabla 13	Comparación del desempeño de las estrategias Enjambre y Multiagente	125

LISTA DE ECUACIONES

Ecuación 1	Probabilidad $P_A(P_B)$ de que la hormiga $(i + 1)$ escoja el camino $A(B)$	57
Ecuación 2	Reglas para la dinámica de la elección de la hormiga La probabilidad p_{ij}^k de que la hormiga k escoja la arista que va desde el nodo	57
Ecuación 3	i hasta el nodo j	59
Ecuación 4	Cantidad de feromona $\Delta\tau_{ij}^k$ depositada por la hormiga en un camino	59
Ecuación 5	Cantidad de feromonas en cada arista	59
Ecuación 6	Niveles de evaporación de la feromona	59

LISTA DE ANEXOS

ANEXO 1	SOFTWARE DE SIMULACIÓN	138
ANEXO 2	TÉCNICAS DE ROBÓTICA	142
ANEXO 3	SÍNTESIS DE UN SISTEMA NERVIOSO	146
ANEXO 4	IDENTIFICACIÓN DEL SISTEMA ROBÓTICO	151
ANEXO 5	PRUEBAS ESTADÍSTICAS (Archivo en Excel)	164

Agradecimientos

*Dedico este trabajo a mi familia, por su paciencia, su comprensión
y su apoyo permanente, especialmente a mi sobrina Sarita,
quién me enseña a ver cada día con nuevos ojos
y a no olvidar a la niña que llevo dentro .*

Margarita María Vallejo Jiménez

1 PRESENTACIÓN

Para el estudio, investigación y desarrollo en lo que se refiere a la navegación de robots móviles en ambientes desconocidos se usan 2 tipos de técnicas, las técnicas tradicionales de inteligencia artificial y las técnicas modernas basadas en comportamientos.

En el *enfoque tradicional* en robótica móvil colaborativa se utiliza *generalmente* una aproximación centralizada (Liu & Wu, 2001), y se expresa el entorno bien conocido como una representación simbólica (un modelo) donde los robots navegan; de tal forma que dada la posición inicial de un robot y un destino, el plan construido deberá encontrar la mejor ruta posible (de acuerdo a cierto criterio de optimización) de todas las rutas posibles que unan el inicio con el destino.

Por otra parte, las técnicas *basadas en comportamientos* utilizan sistemas descentralizados que prescinden del uso de modelos del entorno y se basan en reacciones inmediatas (comportamientos) que debe tener un robot cuando se presentan situaciones inesperadas (Arkin, 2000); estas técnicas son ideales para uso de robots en ambientes desconocidos, están inspiradas en la naturaleza e imitan el comportamiento animal ante diferentes situaciones (Bonabeau & Théraulaz, 2000; Brooks, 1990; Mataric, 2002; Martinoli & Mondada, 1997; Webb, 2009).

Adicionalmente, desde hace algún tiempo se han estado implementando soluciones de *robótica colectiva*, para problemas de optimización de tareas con sistemas robóticos móviles: exploración de áreas, recolección de muestras, mantenimiento de ductos etc. En este ámbito hay dos enfoques de trabajo para la robótica: los sistemas *multiagente* y los sistemas de *enjambre*.

La *robótica multiagente* supone que los robots tienen un alto grado de inteligencia y alguna capacidad de comunicación entre ellos (Balch & Arkin, 1994), mientras que la *robótica de enjambre* estudia grupos de robots, generalmente homogéneos, con comportamientos reactivos y pocas capacidades individuales, que utilizan la interacción extensiva con el mundo real para producir el resultado global deseado (Ijspeert, Martinoli, Billard & Gambardella, 2001; Mondada et al., 2003).

Casi todo el trabajo en robótica móvil cooperativa comenzó después de la introducción del *control basado en comportamientos* como un nuevo paradigma de la robótica (Brooks, 1986A). Este paradigma ha tenido una fuerte influencia para mucha de la investigación en el tema, sin embargo en ocasiones se han logrado soluciones más eficientes, mediante el empleo de las técnicas convencionales implementadas en sistemas de múltiples robots.

El trabajo con robótica colectiva implica, entonces, *determinar cuál de las dos tendencias, multiagente o enjambre, es la más adecuada para la tarea especificada*. Por supuesto, no es de sorprender que algunas veces el resultado de este análisis concluya que lo mejor no es utilizar una técnica o la otra, sino una mezcla de ambas; por ejemplo, se pueden utilizar técnicas

subsimbólicas, como los Algoritmos Genéticos, para generar comportamientos básicos de un robot (Liu & Wu, 2001); o bien técnicas simbólicas que permitan mejorar la calidad de los planes de alto nivel; o incluso de las operaciones conjuntas entre diversos agentes heterogéneos.

La ventaja que aporta la robótica colectiva es que se trata de una solución distribuida, la cual evita el cuello de botella presente en controladores centralizados, reduciendo la susceptibilidad a los fallos en robots individuales e incrementando su reactividad a un entorno dinámico. El problema más difícil es predecir el comportamiento global de los sistemas emergentes a partir del diseño de las reglas de control individuales de cada robot. La metodología típica es proponer una regla de control local, que permita a un grupo de robots resolver un problema dado y a continuación estudiar el comportamiento resultante del grupo utilizando una simulación y/o robots reales. Esta metodología suele basarse en resultados matemáticos de convergencia (Bonabeau, 2002).

Una vez que se dispone de un mecanismo para diseñar comportamientos básicos se debe encontrar un modelo para combinarlos, esto es definir la arquitectura de control de acuerdo a los tres paradigmas: Deliberativo, Reactivo o Híbrido; y llevar a cabo una implementación o realización de la misma, diseñando y programando un sistema software con la funcionalidad elegida para la arquitectura propuesta. Finalmente hay que plantear la forma como se generan los comportamientos en los robots, el área de la IA que se dedica a investigar este campo es el Aprendizaje Automático (Matellan, 2012).

Ahora bien, independientemente de la tarea que deba ser efectuada, varios de los problemas críticos en la robótica móvil corresponden a la navegación: exploración, evasión de obstáculos y definición de rutas (Liu & Wu, 2001). Estos problemas presentan varias situaciones que dependen de las características del robot o los robots que sean empleados (cantidad y tipos de sensores, odometría, capacidad de comunicación, aprendizaje, tipo de procesamiento, memoria, etc.); en este sentido generalmente se habla de problemas de localización y mapeado, como el problema clásico de Localización y Mapeado Simultáneos (Bailey & Durrant-Whyte, 2006); de soluciones, como la Navegación por Estima, y de técnicas que asistan a las soluciones, como los Campos Potenciales (Koren & Borenstein, 1991).

Si el robot sabe con precisión su ubicación, la creación de un mapa preciso es algo simple; una vez que un robot tiene un mapa exacto, se puede mover con precisión de forma rápida y con confianza a través del entorno; La dificultad está cuando un robot no tiene un mapa exacto o la ubicación precisa y tiene que moverse a través de una zona desconocida.

La autonomía exige que los robots puedan percibir el entorno mediante sus sensores y que decidan por sí mismos como relacionar estas señales con el cumplimiento exitoso de la tarea (Liu & Wu, 2001). La percepción es el proceso de lograr la sensibilización y comprensión de la información sensorial. Para un robot esto significa el tratamiento de datos sensoriales para obtener datos semánticos. La cognición se refiere a las funciones mentales, los procesos mentales (pensamientos), y los estados de entidades inteligentes y se centra en procesos mentales específicos, tales como comprensión, inferencia, toma de decisiones, planificación y aprendizaje.

2 RESUMEN

En el presente trabajo se comparan dos estrategias de navegación colaborativas, enmarcadas en el contexto de grupos de múltiples robots móviles, aplicados a la búsqueda de rutas óptimas entre un punto de partida y un punto de destino cuya ubicación es desconocida. Se describen las tendencias de la robótica móvil colaborativa y los enfoques de Inteligencia Artificial aplicados, haciendo énfasis en los sistemas de múltiples agentes y las metaheurísticas comúnmente usadas en este ámbito. Posteriormente se hace un análisis de las arquitecturas y se propone una arquitectura inspirada en un sistema nervioso biológico. Finalmente se hace un estudio estadístico que contrasta una estrategia tipo enjambre con una estrategia tipo multiagente, a partir de simulaciones de sistemas multi-robot, con el fin de determinar cuál de estas presenta mejor desempeño para ejecución de tareas colectivas, en ambientes simulados aplicadas a sistemas multi-robots.

3 ÁREA PROBLEMÁTICA

3.1 DELIMITACIÓN DEL AREA PROBLEMÁTICA

En los robots convencionales es característica la integración de muchos sistemas en una misma unidad; debido a esto puede ocurrir que si un componente falla entonces todo el sistema falle, además, si se trata de robots altamente especializados y diseñados para una misión específica, estos pueden ser inefectivos ante situaciones inesperadas; por otra parte, en cuanto a sistemas robóticos de exploración, se han dado casos en los que los vehículos autónomos han dejado de ser precisamente autónomos, ocasionando pérdidas de tiempo en la toma de decisiones.

Esto indica que es necesario recurrir a métodos más eficientes en los procesos de exploración, reconocimiento y búsqueda de rutas; se requiere de sistemas robóticos que puedan tomar decisiones básicas por sí mismos en tiempo real, que no tengan que esperar órdenes para simples tareas de desplazamiento, que se adapten a las condiciones adversas del entorno y que su funcionamiento no comprometa el éxito de la misión.

Se busca, facilitar procesos de exploración y búsqueda en entornos desconocidos, agilizar el desempeño en aplicaciones de navegación y, evitar pérdida de tiempo en toma de decisiones. Para esto, el empleo de robots múltiples con comportamiento colaborativo puede ofrecer la redundancia necesaria para misiones críticas y una solución eficaz para los problemas anteriormente mencionados (Liu & Wu, 2001).

Cuando se refiere particularmente a tareas de búsqueda en sistemas con varios individuos que interactúan entre sí, generalmente se han empleado técnicas basadas en algoritmos deterministas, los cuales no tienen en cuenta la naturaleza no determinista de los sistemas complejos del mundo real, dando como resultado el logro de una meta, pero sin poder modelar realmente el comportamiento o las interacciones entre dichos individuos.

Los algoritmos basados en sistemas con Agentes, apoyados en técnicas heurísticas, entran a describir con mayor precisión los comportamientos tanto individuales como aquellos compuestos por las interacciones entre cada individuo; pero a medida que fueron diseñados para resolver problemas más complejos y más ajustados al mundo real, esta misma complejidad se tradujo en algoritmos mucho más elaborados, difíciles de entender y de simular.

Esta complejidad al describir las interacciones entre los individuos de una población se ha visto solucionada en gran parte, al aplicar algoritmos inspirados en la naturaleza, los cuales, siguiendo algunas reglas (dependiendo del tipo de algoritmo), describen los comportamientos individuales de una población y como resultado, surgen comportamientos e interacciones, que no necesitan ser modelados con anticipación, sino que son el resultado de comportamientos individuales.

Diferentes tipos de algoritmos se han implementado en sistemas de robótica móvil, como los desarrollados en Cao, Fukunaga & Kahng (1997):

- Labores de salvamento, búsqueda de sobrevivientes
- Localización de cajas negras
- Búsqueda de elementos causantes de un siniestro
- Búsqueda y detonación de minas antipersonales
- Monitoreo de volcanes y fallas geológicas
- Búsqueda de grietas en tanques y represas
- Ubicación de fugas en conductos de petróleo o gas
- Localización de sustancias tóxicas, detección de material radioactivo
- Mantenimiento de puentes y de vías férreas
- Recolección de muestras, minería
- Exploración
- Cooperación en tareas industriales

La búsqueda de buenas soluciones a estos problemas requiere de enfoques que abarcan muchos campos diferentes y, en consecuencia, la investigación se basa en áreas como la inteligencia artificial, la teoría de la decisión, y la investigación de operaciones. Las aplicaciones de la investigación incluyen la exploración planetaria, la gestión de la cadena de suministro, la medicina, la gestión de crisis (como la contención de derrames de petróleo), la robótica y los juegos en tiempo real (entretenimiento, juegos serios, entrenamiento y simulación).

3.2 FORMULACIÓN DEL PROBLEMA

¿Qué estrategias de navegación basadas en multiagente y enjambre, pueden ser contrastadas para simulación en ambientes relacionados con robótica móvil colaborativa?

3.3 ANTECEDENTES

Alrededor del mundo se han llevado a cabo diversos experimentos, muchos de ellos enfocados a ejecución de tareas en forma colaborativa mediante agentes robóticos simples. Convencionalmente el estudio de la robótica había sido enfocado al control de un agente único, con pocos esfuerzos para abordar sistemas multi-robot. Sin embargo, las bases de la nueva tendencia tienen su origen en las primeras investigaciones sobre Inteligencia Artificial (Cao et al.,1997).

3.3.1 Primeros robots móviles autónomos: Los primeros trabajos, basados en las teorías de la cibernética (Wiener, 1948) empleaban técnicas de retro-alimentación negativa. El comportamiento de tales sistemas se basa en la diferencia entre el estado presente y el estado deseado. El primer ejemplo de este tipo de sistemas, basados en la emulación del comportamiento animal, son las tortugas de Grey Walter (1950), quien construyó algunos de los primeros robots autónomos electrónicos, basados en electrónica analógica. Walter (1953) quiso demostrar que las conexiones ricas entre un pequeño número de células cerebrales podrían dar

lugar a comportamientos muy complejos. Otro ejemplo es el robot Teseo construido por Claude Elwood Shannon (1951), el cual parece haber sido el primer dispositivo de aprendizaje artificial de su clase.

Pasaron varios años antes de que se produjera un nuevo avance significativo para la robótica móvil; el robot Shakey, desarrollado en 1966, en el Centro de Inteligencia Artificial del Instituto de Investigación de Stanford (ahora SRI International), fue el primer robot móvil de propósito general capaz de razonar acerca de sus propias acciones; Shakey podría analizar la orden y desglosarla en partes básicas por sí mismo. Este fue el primer proyecto que fusionó el razonamiento lógico y la acción física (Nilsson, 1984).

3.3.2 Robots simples reactivos: Muchos han sido los trabajos que se han hecho respecto a las aplicaciones de grupos de robots y a la inclusión de técnicas de inteligencia artificial. En los años 80, Rodney Brooks fue uno de los primeros en darse cuenta de las limitaciones que imponía el enfoque de la IA clásica; Brooks (1987) disputó en favor de construir robots inspirados por la naturaleza, en lugar de construir sistemas inteligentes por resoluciones abstractas y altamente formalizadas de problemas; haciendo esto acentuó la importancia de mecanismos reactivos prescindiendo del uso de modelos del entorno (Brooks, 1991A).

Los robots de Brooks (1991B), fueron entidades determinadas por conductas elementales y mecanismos de activación e inhibición a partir de los cuales construyen sus propios conceptos; estos mecanismos reactivos tienen un acoplamiento muy estrecho entre los datos de los sensores y las activaciones motoras y, además, no tienen control central. Basado en estas ideas, Brooks (1986B) propuso la *Arquitectura de Subsunción*, argumentando que la acción y los comportamientos son más adecuados para la robótica que los cálculos.

En los años 90 Mark Tilden, basándose en ideas similares a las de Brooks, centró sus investigaciones en la creación de una especie de robots que usa transistores como “cerebros”, inspirados en la naturaleza. Tilden (1994) regresó a lo básico, construyendo robots de componentes sencillos con el menor número de transistores para que fueran simples y fuertes. Tilden desarrolló las *Redes Nerviosas*, que proporcionan a los robots un alto grado de robustez para evitar obstáculos, y los hace altamente resistentes; con sus creaciones ha comprobado que unos pocos componentes pueden conformar el cerebro de un robot.

Con el surgimiento del paradigma de robots reactivos, en la década de los 80, apareció el concepto de “*micro-robot*” y un nuevo enfoque que condujo a la búsqueda de autonomía y confiabilidad, en sistemas basados en el comportamiento animal; sin embargo, solo en la década de los 90 se tuvo completo éxito con grupos de robots trabajando cooperativamente en ambientes creados en laboratorio (Cao et al., 1997).

3.3.3 Inspiración en la naturaleza: El campo específico de los robots autónomos ha estado ligado a la simulación de formas de vida animal. En los años 90 se llevaron a cabo muchas investigaciones sobre el comportamiento de los seres vivos, y el papel que ejercen sobre estas las características biológicas (Ijspeert et al., 2001). Las investigaciones más relevantes se basaron

en la ingeniería inversa a partir del estudio de los insectos (Webb, 1996; Holland & Melhuish, 1999), aunque también se destacan investigaciones basadas en otros organismos como la langosta y la lamprea de mar (Ayers, Blidberg & Massa, 1994).

Numerosos investigadores han estudiado e intentado imitar el comportamiento colectivo de ciertos insectos como las hormigas o las avispas (McFarland & Bossert, 1993). También se han propuesto diversos sistemas distribuidos en los cuales emergen comportamientos colectivos como resultado de interacciones locales (Garnier, Gautrais & Theraulaz, 2007; Steels, 1990). Así pues, se tienen formaciones de robots que se mueven uniformemente emulando las bandadas de aves o los bancos de peces; en estos sistemas, cada individuo toma sus decisiones basándose únicamente en las acciones de sus vecinos más próximos (Balch & Arkin, 1994).

3.3.4 Sistemas de múltiples robots: Desde finales de los años 80, la comunidad de investigación de robótica entró muy activamente en la robótica cooperativa. Un trabajo inicial importante es el sistema CEBOT, de Toshio Fukuda (Fukuda, 1988), que demuestra el *comportamiento auto-organizante* de un grupo de agentes robóticos heterogéneos. La investigación sobre enjambres de robótica, de Beni & Wang (1993) demuestra cooperación a gran escala a nivel de una simulación.

Steels (1990) afirmó que el interés de los micro-robots se basa en la flexibilidad de los mismos. En caso de que una unidad falle en poco tiempo debe aparecer su reemplazo, de forma que el sistema alcance un alto grado de seguridad y robustez. Steels demostró que es posible hacer que los micro-robots de un grupo cumplan sus propias sub-tareas y las de sus compañeros inutilizados. En el peor de los casos el sistema “cae”, cuando todas sus unidades lo hacen simultáneamente.

Fusionando los conceptos de robótica colectiva y micro-robots, se han desarrollado muchos trabajos con sistemas de múltiples robots; estos hacen énfasis en distribución, interacciones directas o indirectas entre robots relativamente simples. Los trabajos de Brooks (1986A, 1986B) y Mataric (2001), muestran el desarrollo de grupos de agentes basados en la *arquitectura de subsunción*. Los experimentos incluyen desde pequeños grupos de micro-robots, hasta simulaciones de enjambres con cientos de robots virtuales. Los trabajos más significativos destacan el empleo de capacidades de aprendizaje como la clave para lograr un desempeño eficiente.

En la década del 90 se realizaron importantes investigaciones que sirvieron como base del estudio en robótica colectiva. Así por ejemplo se incluyeron soluciones a los problemas de navegación (Millán & Torras, 1992), evasión de obstáculos (Borenstein & Koren, 1991), algoritmos de cooperación (Beckers, Holland & Deneuborug, 1994; Kube & Zhang; 1992), métodos de comunicación multiagente (Kelly, Keating & Warwick, 1997), aprendizaje por refuerzo y seguimiento de trayectorias de feromona artificial (Russell, Thiel & Mackay-Sim, 1994). En cuanto a los sistemas de múltiples robots, donde cada robot tiene acceso a parte de la información la cual es compartida con los demás para lograr el objetivo, los experimentos comprobaron que la técnica y el trabajo en equipo pueden llegar a ser más importantes que la inteligencia (Martinoli & Mondada, 1997).

3.4 ESTADO DEL ARTE

3.4.1 Robots “hormiga”: Durante la década del año 2000, algunos de los trabajos más significativos se centraron en la robótica de enjambre y la aplicación de los principios de *Inteligencia de Enjambre* para investigar cómo a través de la explotación de interacciones locales puede obtenerse la colaboración en un grupo de robots reactivos simples; en este enfoque se destaca el logro de metas a partir de información local y reacciones simples, en lugar de emplear mapas y modelos del entorno para desplegar acciones elaboradas (Garnier et al., 2007; Kube & Bonabeau, 2000; Trianni & Nolfi, 2009).

El trabajo se centró en algunas de las más complejas capacidades de insectos, incluyendo integración multimodal, navegación y aprendizaje. Se estudiaron robots biomiméticos basados en modelos neurobiológicos simples, con controladores basados en circuitos neuronales establecidos a partir de la neurofisiología (Melhuish, Sendova-Franks, Scholes, Horsfield & Welsby, 2006). Los mayores avances se dieron en el creciente campo de investigación de la optimización por colonia de hormigas, un conjunto de métodos inspirados en la comunicación basada en feromonas de las hormigas biológicas; a partir de estas ideas, Marco Dorigo desarrolló la metaheurística de optimización por colonia de hormigas (Dorigo & Gambardella, 1997).

Las combinaciones de hormigas artificiales y algoritmos de búsqueda local se convirtieron en un método de elección para la solución de numerosos problemas de optimización, que implican algún tipo de grafo, tal como el bien conocido problema del viajante. Recientemente Dorigo se involucró con la investigación en robótica de enjambre.

3.4.2 Investigaciones actuales: Actualmente se llevan a cabo en todo el mundo, labores de investigación referentes a la robótica cooperativa y el comportamiento emergente. La investigación se centra alrededor de las técnicas para decisión (planeación y aprendizaje) que les permite a los robots actuar inteligentemente en sus ambientes y exhibir una conducta dirigida a metas en tiempo real (Rutishauser, Correl & Martinoli, 2009), aun cuando tengan sólo conocimiento incompleto de su ambiente, percepción limitada o ruidosa, habilidades imperfectas de manipulación, o velocidad del razonamiento insuficiente (Koenig & Likhachev, 2002). Hay un especial interés en los sistemas inteligentes que tienen que operar en dominios grandes, no deterministas, no estacionarios o sólo parcialmente conocidos.

Se hace especial hincapié en las técnicas de control escalables que dependen de la auto-organización como mecanismo de coordinación (Ijspeert et al., 2001), en los comportamientos colectivos e inteligencia colectiva (Garnier et al., 2007), en el diseño de micro-robots que se inspiran en los organismos biológicos (Dorigo et al., 2005), en métodos que permitan que varios robots cooperen para cumplir las misiones imposibles para un solo robot (Ducatelle, Di Caro & Gambardella, 2010), en la evolución artificial para sintetizar nuevas formas de organismos electro-mecánicas (Trianni, Nolfi & Dorigo, 2008; Tuci, Ampatzis, Vicentini & Dorigo, 2006) y en hacer ingeniería inversa de circuitos biológicos complejos (Floreano & Mattiussi, 2008).

3.4.3 Tendencias de la investigación: Las tendencias de investigación se encuentran en la intersección entre la robótica, la neurociencia computacional, los sistemas dinámicos no lineales, y el aprendizaje de máquina aplicado. Se destaca el interés en el uso de simulaciones numéricas y robots para tener una mejor comprensión de la locomoción animal y el control de movimiento, y en el uso de la inspiración de la biología para el diseño de nuevos tipos de robots y controladores de locomoción. Los temas de mayor interés son la Robótica Evolutiva, el comportamiento adaptativo (Trianni, Nolfi & Dorigo, 2008), la Robótica Cognitiva (Iocchi, Nardi & Cesta, 2003), la Robótica Epigenética y el aprendizaje por refuerzo.

Una tendencia sobresaliente se utiliza en el concepto de "*Ecología de Sistemas Inteligentes Físicamente Embebidos*" o "*PEIS (Physically Embedded Intelligent Systems) Ecology*", un nuevo enfoque desarrollado por Saffiotti y Broxvall (2005, 2008) el cual intenta incluir tecnologías robóticas en la vida cotidiana, basado en la combinación de la robótica cooperativa y la *Inteligencia Ambiental*.

4 JUSTIFICACIÓN

Las aplicaciones de robots móviles están más allá del alcance de la tecnología existente y muestran lo inadecuado de las metodologías del enfoque tradicional. Una de las características que se persigue en los robots es la autonomía o capacidad para interactuar con el entorno de forma independiente, por lo que debe tenerse en cuenta que las reacciones deben ser rápidas y que pensar mucho puede tomar mucho tiempo, además para pensar se necesita información precisa lo cual no es práctico de obtener (Brooks et al., 1998; Cao et al., 1997; López, Mata & García, 2001; Silva et al, 2007; Toal, Jones & Strunz, 1996).

Una solución eficaz, ante este problema, la brindan los sistemas de robótica colectiva, los cuales presentan varias ventajas en cuanto a la eficiencia y el desempeño. Actualmente la tendencia es poner énfasis en la comunicación de experiencias entre varios robots, ligadas a sus reacciones dentro del entorno en el cual se desenvuelven; tal comunicación deberá permitir, a los demás robots, aprender y decidir la acción a realizar (D'Angelo & Menegatti, 2007; Iocchi, Nardi & Cesta, 2003; Pagello, D'Angelo & Ferrari, 2003; Stone & Veloso, 2000; Vallejo, Ochoa & Jiménez, 2009).

Ahora bien, al observar los sistemas robóticos de exploración convencionales, se aprecian deficiencias que podrían solucionarse aplicando un método similar al usado por una colonia de hormigas. El uso de robots está ampliamente difundido en labores de exploración y reconocimiento, sin embargo, si se trata de un robot altamente especializado y programado para una misión específica este es, de todas formas, susceptible a sufrir fallas y averías en medio de la exploración.

Se requiere, entonces, de sistemas robóticos eficientes que puedan tomar decisiones básicas por sí mismos en tiempo real, y no tengan que esperar órdenes desde una base para efectuar tareas simples de desplazamiento; que se adapten a las condiciones adversas del entorno y que la misión no dependa de su funcionamiento.

Un sistema multi robot tiene propiedades tales como un rango grande de tareas que pueden realizar, gran eficiencia, posee mejor rendimiento, mayor tolerancia a los fallos, robustez, facilidad de implementación y bajo costo, detección y actuación distribuida, paralelismo inherente y mediante sus interacciones se puede echar un vistazo a las ciencias sociales y biológicas (Liu & Wu, 2001).

Es de resaltar que las acciones esperadas en los robots deben ser cercanas a las encontradas en la naturaleza, así pues, la observación de los animales sociales inspira la investigación en robótica colectiva y el desarrollo de algoritmos que permitan lograr optimización en cualquiera de los aspectos involucrados. Los investigadores en comportamiento animal han propuesto muchos modelos para explicar algunos aspectos interesantes tales como auto-organización y agrupaciones.

Se hace necesario, imitar este tipo estrategias colaborativas; por lo tanto se han propuesto algoritmos inspirados en los modelos de la naturaleza, para resolver problemas computacionales de alta complejidad, lo que los hace apropiados para su simulación e implementación (Cen, Song & Wang, 2008; El-Abd, 2010; Koenig, Szymanski & Liu, 2001; Kumar, Sahin & Cole, 2004; Saffari & Mahjoob, 2009; Sato, 2007; Svennebring & Koenig, 2004).

En cuanto a los algoritmos de exploración, evasión de obstáculos y planificación de rutas, estos tratan de encontrar soluciones óptimas en tiempo, distancia, costo y otros factores que al ser aplicados en la vida real, toman en cuenta muchas otras variables como la velocidad y la interacción con otros robots, además de la cantidad de posibles rutas a elegir. Mediante algoritmos basados en inteligencia de enjambre es posible encontrar una solución óptima para problemas complejos, de gran cantidad de variables, en un período razonable de tiempo (Dorigo et al., 2005; Iocchi, Nardi & Cesta, 2003; Rybski et al., 2008).

5 OBJETIVOS

5.1 OBJETIVO GENERAL

Simular y contrastar el desempeño de estrategias de navegación, basadas en inteligencia colaborativa, en entornos estáticos para robots móviles.

5.2 OBJETIVOS ESPECÍFICOS

- Desarrollar una estrategia base de navegación para robótica móvil aplicable a sistemas multi-robot.
- Contrastar el desempeño de la técnica multiagente y de la técnica de enjambre, en la solución del problema de navegación en entornos estáticos, usando el entorno virtual.
- Establecer los resultados derivados de la observación de las acciones e interacciones simuladas y generadas a partir del comportamiento de los robots.

6 MARCO TEÓRICO

Los conceptos y definiciones aquí expuestos inician con la biomimética, seguido de sistema complejo, enfoques de la inteligencia artificial, sistemas basados en agentes, sistemas multiagente, sistemas de enjambre, arquitecturas de control, robótica móvil y metaheurísticas. Se presentan mapas conceptuales, que relacionan cada uno de los conceptos expuestos con algoritmos multiagentes o con algoritmos de enjambre.

6.1 BIOMIMÉTICA

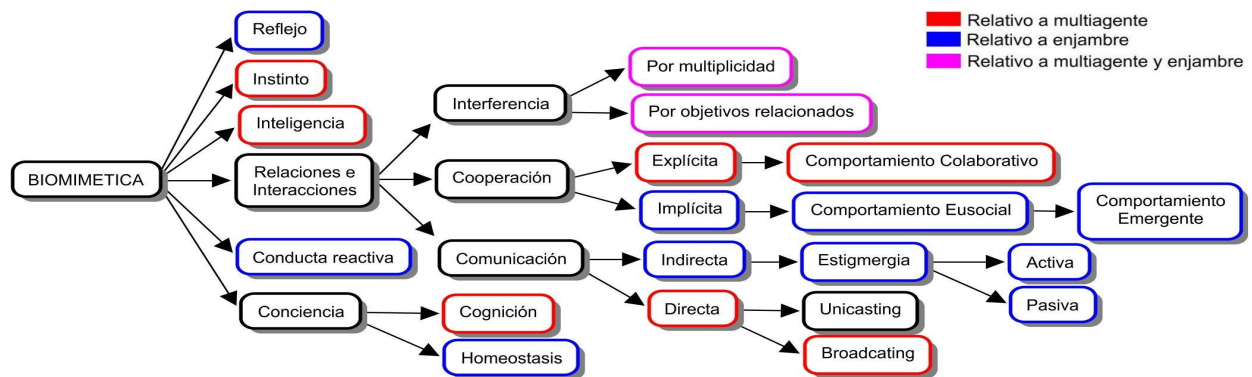


Figura 1. Temas tratados de Biomimética.

Cualquier robot móvil que nade, vuele, o camine se inspira en la locomoción animal, y puede decirse que es *biológicamente inspirado* (Delcomyn, 2007). El concepto actual de la robótica de inspiración biológica se originó en las últimas décadas del siglo XX con las ideas de Beer, Quinn, Chiel & Ritzmann (1997). Últimamente, este enfoque se ha llamado *robótica biomimética* (Ayers & Witting, 2007).

Debido a que los animales son extremadamente complejos no es posible obtener una reproducción completa en hardware o software; sin embargo algunos investigadores sugieren que hay que incorporar en el robot el máximo de características de un animal como sea posible (Ritzmann, Quinn, Watson & Zill, 2000), incluso si la ventaja funcional de una característica particular no es clara (Cham et al., 2004; Dillmann et al., 2007); el argumento es que muchas de estas características confieren atributos útiles al robot, incluso si no son evidentes, además algunos investigadores sostienen que el estudio de los robots puede contribuir a la comprensión de los principios biológicos (Beer et al., 1997; Ritzmann et al., 2000; Webb, 2006). Otros consideran que incluir demasiadas características animales a un robot perjudica su rendimiento (Yoneda & Ota, 2003).

Los animales que más han inspirado a los investigadores de la robótica han sido los insectos, no solo por la sencillez de sus comportamientos, sino también por su éxito evolutivo, muchas especies lograron tal perfección que han permanecido sin cambio alguno a lo largo de millones de años. (Newman, 1970).

6.1.1 Conducta Reactiva: La conducta puede definirse simplemente como la respuesta de un organismo al medio ambiente. Toda forma de conducta, simple o compleja, es una reacción a estímulos, un estímulo es un cambio en un aspecto determinado del medio ambiente. La conducta depende del desarrollo de los órganos de los sentidos y puede variar en un mismo individuo. El estado interno de un animal no solo produce tendencia hacia ciertas clases de conducta, sino que también cambia las respuestas usuales a los estímulos (Carthy, 1970).

Casi todo el comportamiento de los animales es automático, produciendo la respuesta apropiada, incluso ante una situación nueva; no se requiere aprendizaje alguno, porque cuando se capta el estímulo, se produce un flujo de impulsos a lo largo de fibras nerviosas adaptadas justamente para cada situación y como resultado surge una respuesta concreta (Boorer, 1980). Una característica principal de los organismos vivos es la reacción, un movimiento del individuo respondiendo y ajustándose al entorno y a su propio estado interno. La mayoría de los animales tienen un tiempo de respuesta limitado (Carthy, 1970), los individuos que reaccionan más rápidamente tienen más posibilidad de supervivencia que los individuos lentos (Sparks, 1971).

La respuesta de un animal a una situación depende de su estructura física y el grado de desarrollo de su sistema nervioso (Carthy, 1970). En cuanto a los insectos su comportamiento depende casi por completo de la reacción a estímulos y en general se pueden predecir sus acciones; los insectos de cada especie siguen los mismos modelos de conducta (Carthy, 1970). El sistema nervioso de un insecto constata un cordón nervioso doble que recorre el cuerpo; en cada segmento del cuerpo hay un engrosamiento llamado ganglio, del que parten nervios hacia músculos y órganos; el ganglio mayor está en la cabeza y se considera como cerebro, pero los menores pueden actuar, en gran proporción, independientemente de él; por esto los insectos siguen vivos y con cierta actividad cuando les falta la cabeza; el cerebro *inhibe* y limita muchos movimientos, si desaparece, los movimientos son exagerados y atáxicos (Newman, 1970).

6.1.2 Instinto: Designa la conducta innata que no se aprende ni se modifica por la experiencia (Carthy, 1970); es específico para cada especie, gracias a este el organismo se comporta de la manera más adecuada ante las situaciones más comunes enfrenta. Sus características son:

- Es común a una misma especie.
- Su finalidad es la *adaptación* del individuo.
- Es de carácter *global*, compromete a todo el individuo.
- *Permanece* aún después de haber desaparecido el estímulo que lo motivó.

El instinto consta de una serie de pasos para su producción:

- Percepción de la necesidad.
- Búsqueda del objeto.
- Percepción del objeto.
- Utilización del objeto.
- Satisfacción y cancelación del estado de necesidad.

6.1.3 Reflejo: Designa las acciones simples que se deben al arco reflejo, una conexión nerviosa establecida entre los órganos sensoriales y el músculo efector; este un atajo elimina el tiempo que necesitaría un impulso para ir hasta el cerebro y producir como respuesta una serie de impulsos que conduzcan la acción (Carthy, 1970). Los reflejos cumplen una función protectora y ofrecen una respuesta acertada ante situaciones peligrosas; su puesta en marcha es automática cuando un estímulo sobrepasa cierto umbral, y no depende de la especificación de un objeto para producirse. Sus características son:

- Es común a una misma especie.
- Su finalidad es la *protección* del individuo.
- Es de carácter *local*, compromete solo a una pequeña parte del individuo.
- Obra sólo *mientras* se presenta el estímulo.

Entre las propiedades de los reflejos se encuentran:

- *Invariabilidad:* Un estímulo determinado causa siempre el mismo reflejo.
- *Integridad:* Una interrupción en el arco suprime el reflejo.
- *Velocidad:* Incluye el recorrido de las señales en los nervios; el tiempo de respuesta y el tiempo de reacción del músculo.
- *Ecuación Personal:* El tiempo de la reacción es variable según los individuos.
- *Fatigabilidad:* Si se provoca repetidas veces el mismo reflejo, el tiempo de reacción aumenta progresivamente y la intensidad del movimiento disminuye hasta que la reacción acaba por no producirse. La fatiga aparece al nivel de las sinapsis de las neuronas.

Las acciones reflejas obedecen las siguientes leyes:

- *Ley de la intensidad:* la intensidad de la reacción es directamente proporcional a la intensidad de la excitación.
- *Ley de la localización:* Un estímulo de baja intensidad causa reacciones unilaterales.
- *Ley de simetría:* Un estímulo de mediana intensidad causa reacciones bilaterales.
- *Ley de irradiación:* Un estímulo de alta intensidad causa reacciones en efectores cercanos.

6.1.4 Inteligencia: Si bien parte del comportamiento es rígido y se desarrolla automáticamente, los animales poseen una considerable aptitud para aprender (Sparks, 1971), y aunque el instinto es la mejor clase de comportamiento para la mayoría de animales, pues sus vidas no dan tiempo para el aprendizaje, el instinto presenta el inconveniente de ser inflexible; por otra parte un comportamiento inteligente permite mayor flexibilidad, en el sentido de que se pueden hallar soluciones más matizadas a los problemas que se presentan (Boorer, 1980). Las características de la conducta inteligente son:

- Es propia de cada individuo.
- Se aprende por medio de la experiencia.
- Es flexible, puede modificarse y adaptarse.

Para que un individuo altere su conducta debe poder retener experiencias pasadas; esa capacidad de aprender permite seleccionar entre una serie de métodos posibles para solucionar un problema. Si ocurren cambios en el entorno, el individuo puede alterar su conducta para acomodarse a estos cambios, en lugar de responder siempre de un modo idéntico (y quizá ineficaz). Cuando un individuo es capaz de aprender, cada vez ejecuta la acción con más perfección, cometiendo cada vez menos errores (Carthy, 1970).

6.1.5 Cognición y Homeostasis: *Cognición* se refiere a la facultad de procesar información a partir de la percepción, la experiencia y características subjetivas que permiten valorar la información; la investigación en el campo aborda capacidades de los sistemas tales como abstracción, generalización, concreción/especialización y razonamiento, en las cuales se involucran conceptos subjetivos como *creencias, conocimiento, estados mentales y preferencias*. Por otra parte, *Homeostasis* es la característica de un sistema mediante la cual se regula el ambiente interno para mantener una condición estable y constante. En robótica es la capacidad para mantener ciertas variables en estado estacionario, de equilibrio dinámico o dentro de ciertos límites, cambiando parámetros de su estructura interna. La homeostasis responde a cambios en:

- **El medio interno:** Eliminación de sustancias de desecho o liberación de mensajeros químicos, como neurotransmisores y hormonas que regulan múltiples funciones fisiológicas.
- **El medio externo:** Percepción de estímulos externos o captura y conservación de la energía.

6.1.6 Relaciones e interacciones:

- **Interferencia:** Influencia que impide a un individuo alcanzar un objetivo; una manera de disminuir las interferencias es definir reglas sociales que permitan optimizar la distribución de recursos. Hay dos tipos de interferencia:
 - **Por multiplicidad:** Se da en sociedades con iguales objetivos para todos; los individuos compiten por recursos, si el número de individuos crece aumenta la interferencia y disminuye el desempeño del equipo.
 - **Por conflictos de objetivos relacionados:** Cuando se distribuyen objetivos globales en sub-equipos, estos objetivos pueden ser diferentes, pero pueden coincidir a nivel de sub-objetivos, de manera que se sigue presentando competencia por recursos, esta vez para alcanzar objetivos globales diferentes. En este caso se presentan problemas mayores como puntos muertos y oscilaciones.
- **Cooperación:** Interacción basada en alguna forma de *comunicación*, con objeto de alterar el estado de los individuos para que trabajen en función de un objetivo relevante para el sistema. Hay dos modos de cooperación (McFarland, 1994):
 - **Cooperación explícita:** Corresponde al *Comportamiento Cooperativo*, definido como las interacciones entre individuos “egoístas”, con el fin de maximizar el *beneficio individual*.
 - **Cooperación implícita:** Corresponde al *Comportamiento Eusocial* (Batra, 1966), definido como las interacciones entre individuos “altruistas”, con intención de maximizar el *beneficio grupal* (Wilson, 1971).
- **Comunicación:** Envío de información a otros individuos, es la forma más común de interacción entre individuos inteligentes, y un aspecto muy importante para el alcance efectivo de acciones en comunidades. Aún se encuentra bajo investigación la cantidad óptima de individuos para que un sistema garantice la distribución efectiva de la información (Vallejo, Ochoa & Jiménez, 2009). Se identifican dos tipos de comunicación:
 - **Comunicación directa:** Con el único propósito de transmitir información; puede ser de tipo uno-a-uno (unicasting) o uno-a-varios (broadcasting), en cualquier caso es dirigida a receptores identificados.

- **Comunicación indirecta:** Basada en el comportamiento observado de otros agentes y no en comunicación con ellos; ocurre como efecto lateral de otras acciones que generan información sin un remitente explícito.

6.1.7 Estigmergia: Es un mecanismo de comunicación indirecta, a través de un medio físico; que produce estructuras complejas, aparentemente inteligentes, sin necesidad de planificación o control; apoya la colaboración eficiente entre individuos simples, que carecen de memoria, inteligencia o conocimiento de la existencia de otros (Marsh & Onof, 2008). El concepto fue introducido por Pierre-Paul Grassé, al estudiar el comportamiento de las termitas, quien lo define como: "*Estimulación de los trabajadores por el desempeño que han logrado*" (Grassé, 1959, p. 50). El principio es que la huella dejada en el medio por una acción, estimula una acción siguiente, del mismo individuo o de otro; las acciones posteriores tienden a reforzarse, dando lugar a la aparición espontánea de actividad coherente y aparentemente sistemática.

En robótica se destacan dos tipos de Estigmergia:

- **Estigmergia Activa:** Se produce cuando un individuo altera su entorno para afectar la información sensorial de otro; un ejemplo típico es soltar artefactos que otros puedan recoger.
- **Estigmergia Pasiva:** Se produce cuando *la acción* de un individuo altera su entorno, de tal manera que los cambios medioambientales realizados por un individuo diferente también se modifican; un ejemplo típico es cuando un individuo intenta eliminar todos los objetos de un recipiente, mientras que otro intenta llenar el recipiente completamente.

El concepto de Estigmergia deja abierta la cuestión de cómo los estímulos deben ser organizados en el tiempo y en el espacio para una coordinación perfecta. A pesar de la vaguedad de la formulación de Grassé (1959), estigmergia es un concepto profundo, cuyas consecuencias aún no se han explorado en totalidad (Garnier et al., 2007). Actualmente el concepto ha sido tomado y extendido a una serie de algoritmos que forman parte de la inteligencia artificial

Las características principales de un sistema con estigmergia son:

- Los individuos no confían en las instrucciones.
- Cada individuo reúne información propia y decide por sí mismo lo que debe hacer.
- La información se obtiene de un entorno compartido.
- No hay toma de decisiones centralizada.
- Los individuos sólo necesitan unas pocas reglas elementales.

6.1.8 Comportamiento Emergente: Muchas especies de seres vivos con conductas individuales, *simples* y rudimentarias, tienen la capacidad de generar conductas colectivas *complejas*, cuyas características y consecuencias *superan a la simplicidad* de la conducta individual. Este fenómeno se denomina *comportamiento emergente*; una forma de explicar el concepto es asumiendo que *el todo es mayor que la suma de las partes*. El comportamiento emergente se da cuando *la interacción entre objetos en un nivel da origen a objetos en otro nivel*; un comportamiento es emergente si para describirlo se requieren nuevas categorías, las cuales no son requeridas para describir el comportamiento de los componentes que lo

constituyen. Se pueden observar comportamientos globales muy complejos partiendo de comportamientos locales sencillos (Miceliet al., 2005).

Un comportamiento emergente inicia con el procesamiento de información *local* por parte de los individuos y surge cuando cierto número de individuos designados para comportarse de determinada forma, se involucra en interacciones locales con otros individuos, formando patrones *globales* de procesamiento de información que pueden percibirse cuando se observa a un nivel macroscópico (Reynoso, 1998).

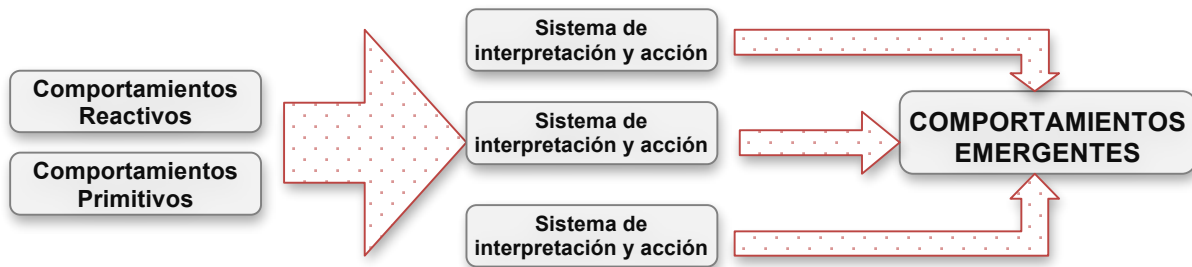


Figura 2. Comportamientos Emergentes a partir de comportamientos sencillos. Los sistemas de interpretación y acción se conocen como “motor schemas”. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

En todos los casos en los que se da este tipo de comportamiento hay presentes tres condiciones:

- **Objetivo prioritario:** Proporciona una meta y dicta un comportamiento en ausencia de información local, y en caso de que esta exista se utiliza para tomar decisiones, que se espera que conduzcan al logro del objetivo prioritario en forma más rápida.
- **Reglas de interacción local:** La búsqueda de soluciones por parte de los *agentes* independientes deposita y modifica la información en el universo, y es a través de éste método indirecto que un *agente* comunica a los demás los resultados de sus esfuerzos.
- **Universo para interactuar:** Es el depositario de la información y el que la proporciona a los agentes; en ningún caso un agente tiene conocimiento global del mismo, por el contrario, se limita a la información local que, en su caso, está depositada en la geografía por la que transita en el momento.

6.2 SISTEMA COMPLEJO

Un sistema complejo es un sistema formado por un gran número de elementos, que interactúan localmente y que dan lugar a algún tipo de *comportamiento emergente*.

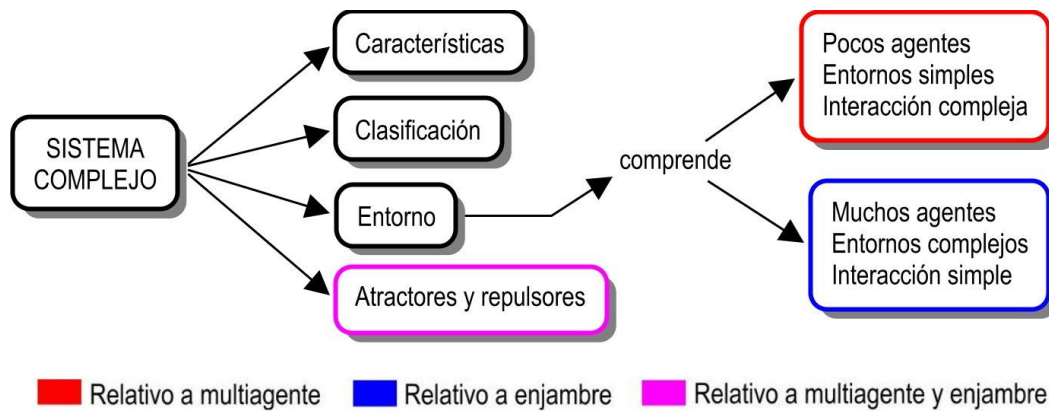


Figura 3. Temas tratados de Sistema Complejo.

6.2.1 Características: Las características fundamentales son: gran número de individuos idénticos, interacción local, y Comportamiento Emergente (Miceli et al, 2005).

- La noción de complejidad se refiere a la aparición de propiedades emergentes, provenientes de la interacción local de un número de componentes, no necesariamente alto.
- No necesariamente un modelo complicado se aproxima más a la verdad que uno sencillo.
- Sistemas ontológicamente distintos no se comportan de manera distinta, ya que para comprender el funcionamiento de un sistema no hay que tener en vista todos sus detalles.
- Operar con un modelo implica, por lo tanto, no considerar a todas las variables que lo conforman como igualmente relevantes.
- El comportamiento complejo representa una situación intermedia entre un estado ordenado y otro totalmente desordenado.
- No importa cuánto se modifiquen las magnitudes escalares del sistema, siempre el comportamiento del mismo mostrará la misma pauta autoorganizativa (autosimilaridad).

6.2.2 Clasificación:

- **No-lineales:** Las entradas son a veces función de las salidas, y las salidas incluyen efectos que se realimentan al sistema. Explican las características que adopta un subsistema por la influencia del sistema donde anida.
- **Adaptativos:** Se amoldan al ambiente y muestran capacidad de absorber fluctuaciones.
- **Auto-organizados:** Las regularidades detectadas se comprimen en un esquema interno, independiente de toda regla externa. La auto-organización es un conjunto de mecanismos dinámicos por lo que las estructuras aparecen en el nivel global de un sistema a partir de las interacciones de sus componentes de nivel inferior (Garnier et al., 2007).
- **Evolutivos:** Muestran una cierta lucha por el esquema más apto, amoldándose a la supervivencia del mejor esquema frente a esquemas rivales.
- **Ubicados entre el orden y el caos:** Permiten la emergencia de nuevas soluciones por nuevas ligaduras en el campo de grados de libertad del sistema. Muchos de estos fenómenos se han estudiado usando simulaciones por computador.

6.2.3 Entorno: Los entornos pueden influenciar y restringir la conducta del individuo, además presentan propiedades que permiten categorizar la dificultad de adaptación a la que este se enfrentará. Se pueden diferenciar varias clases de entornos, de acuerdo a la influencia que ejercen sobre las acciones del agente:

- **Accesibles:** Si el agente puede tener el conocimiento total del entorno.
- **Deterministas:** Si el estado siguiente del entorno se determina solamente como consecuencia del estado actual y de la acción del agente.
- **Episódicos:** Si la calidad de la decisión de un agente depende solamente del estado del sistema y de dicha decisión, y no se afecta por decisiones anteriores.
- **Estáticos:** Si el entorno no cambia *durante la deliberación* del agente.
- **Dinámicos:** Si el entorno cambia sin esperar la decisión del agente.
- **Semi-dinámicos:** Si el tiempo afecta la calidad de la decisión del agente pero no al estado.
- **Continuos:** Si la cantidad de percepciones y acciones posibles es infinita.
- **Discretos:** Si es finita la cantidad de percepciones y acciones posibles.

Sin embargo, si se busca implementar sistemas de agentes robóticos en el mundo real, se deben tener en cuenta las siguientes consideraciones (Russell & Norvig, 2009):

- **El mundo real no es accesible:** Los sensores no son perfectos y siempre percibirán exclusivamente aquellos estímulos que estén cercanos al agente.
- **El mundo real no es determinista:** Al menos desde el punto del robot, por lo que no existe certeza de que una acción determinada llegue a funcionar, por lo tanto, es necesario que los robots tengan capacidad para desenvolverse adecuadamente en medio de la incertidumbre.
- **El mundo real no es episódico:** Los efectos producidos por las acciones cambian con el tiempo, los robots deben lidiar con problemas de decisión secuenciales y con el aprendizaje.
- **El mundo real es dinámico:** Por lo tanto, es importante que un robot sepa cuando conviene deliberar y cuando es preferible actuar de inmediato.
- **El mundo real es continuo:** en tanto que los estados y las acciones proceden de un continuo de configuraciones físicas y movimientos, esto impide numerar en su totalidad el conjunto de sus posibles acciones.

En cuanto a las interacciones que se pueden presentar, se pueden dar varias combinaciones de los factores implicados; las interacciones que se abordan con más frecuencia son:

- **Interacción, relativamente compleja entre pocos agentes e entornos muy sencillos y entre sí,** es simbólica en su mayor parte y depende poco de los cuerpos de los agentes; el énfasis se pone en la estructura lógica de la interacción.
- **Interacción relativamente sencilla entre muchos agentes en entornos ligeramente más complejos,** depende de que los agentes tengan cuerpo, estos pueden ser físicos o simulados; el énfasis se pone en la emergencia de orden a partir de formas simples de interacción.

6.2.4 Atractores y repulsores: *Atractor* es una singularidad en el entorno hacia la cual convergen las trayectorias de una determinada dinámica, que encuentran allí una condición *local* de máxima energía. Los atractores representan el comportamiento asintótico del sistema, es decir, un atractor es el conjunto al cual el sistema evoluciona después de un tiempo suficientemente largo; el ejemplo típico es el punto más bajo de un péndulo. El concepto opuesto es el *repulsor*, y un ejemplo apropiado es el punto más bajo de un péndulo invertido.

El atractor de algunos sistemas complejos coincide con su estado de auto-organización. El atractor de algunos sistemas orientados a satisfacer una meta, es la meta misma. Para que un conjunto sea un atractor, las trayectorias que le sean suficientemente próximas han de permanecer próximas incluso si son ligeramente perturbadas. Geométricamente, un atractor puede ser un punto, una curva o un conjunto complicado de estructura fractal. La trayectoria del sistema dinámico en el atractor no tiene que satisfacer ninguna propiedad especial excepto la de permanecer en el atractor; puede ser periódica, caótica o de cualquier otro tipo.

Los atractores pueden clasificarse según Martínez (2003) en:

- **Puntuales:** Puntos fijos, que señalan un estado de equilibrio final del sistema.
- **Cíclicos:** Orbitas periódicas que señalan que la configuración del sistema evoluciona de forma periódica repitiendo sus estados.
- **Caóticos o Extraños:** Cuya sensibilidad a las condiciones iniciales impide la predictibilidad. Son producto de auto-oscilaciones (perturbaciones que se mantienen, en sistemas no lineales, mediante retroalimentación) que dan lugar a inestabilidades locales, producto de las condiciones de mezcla en que se encuentran los sistemas.

6.3 SISTEMAS BASADOS EN AGENTES (SBA)

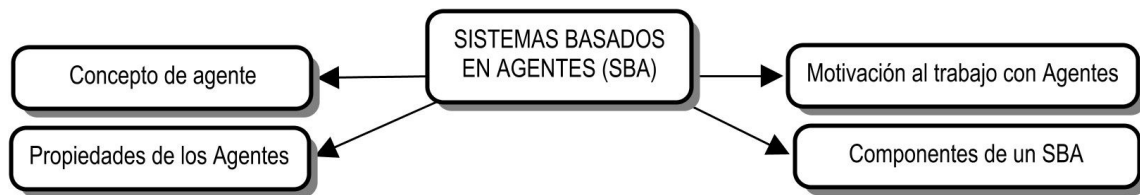


Figura 4. Temas tratados de Sistemas basados en Agentes.

6.3.1 Concepto de “Agente”: Un agente es un *solucionador de problemas* (D’Inverno & Luck, 2007) que percibe estímulos del entorno mediante sensores, y ejecuta acciones mediante efectores (Russell & Norvig, 2009); las acciones son definidas como eventos discretos que pueden cambiar el estado del entorno al ser ejecutados.

El trabajo con agentes se basa en construir sistemas inteligentes con énfasis en la interacción, en lugar de en los componentes. Una forma de estudiarlos se funda en las dinámicas de interacción entre componentes simples que llevan a una complejidad emergente. La teoría de agentes sostiene que la *interacción social local* es determinante para el comportamiento de los individuos ya que este es discreto, no lineal y simultáneo respecto al resto de los agentes. Los modelos basados en agentes adoptan una postura de *Adaptabilidad*, pues la información que tienen los actores está acotada a lo que observa de sus vecinos; del mismo modo, su influencia únicamente alcanza a su vecindario (Miceli et al., 2005).

6.3.2 Motivación al trabajo con agentes: Generalmente han existido dos formas opuestas de hacer ciencia: el método hipotético *deductivo* que parte de una serie de reglas a priori, para deducir patrones de conductas; el método *inductivo* que parte de la recolección de datos para establecer generalizaciones empíricas. Los *modelos basados en agentes* establecen una vía intermedia, en la que los modelos propuestos son construcciones del investigador, el cual establece reglas de conducta a investigar acercándolas al método deductivo, pero como modelo, pues no significan que tengan valor de verdad, sino que son una evidencia más de la validez de las reglas establecidas. Los acerca así a los modelos inductivos ya que se convierten en una muestra más aplicable al objeto de estudio (Miceli et al., 2005).

Si existe un problema demasiado complejo o grande, puede convenir dividirlo en componentes modulares enfocados en partes específicas del problema; el objetivo final puede conseguirse mediante la colaboración entre tales módulos. Un sistema basado en agentes facilita esta descomposición del problema, en componentes especializados en partes de dicho problema; cada agente utiliza la técnica más apropiada, en lugar de tener que adoptar una técnica global, que podría no ser óptima para cada caso particular. De esta forma se puede concebir el sistema como una *sociedad cooperativa de elementos autónomos especializados en la resolución de problemas concretos y sencillos*.

6.3.3 Componentes de un Modelo Basado en Agentes (Miceli et al, 2005):

- **Los Agentes**, quienes poseen estados internos y reglas de conducta.
- **Los estados internos** de los agentes, que pueden ser fijos o cambiantes.
- **El entorno**, sobre el que los agentes operan y con el que interactúan.
- **Las reglas de conducta**, que se aplican a los agentes entre sí, a la interacción de los agentes con el entorno y al propio entorno.

Puede definirse comportamiento como una ley de control para alcanzar y/o mantener una meta particular; luego, un comportamiento es un tipo de operador que garantiza una meta, cualquiera sea su tipo. Un agente puede tener múltiples metas concurrentes, incluyendo al menos una *meta a objetivo* (meta que al ser alcanzada hace finalizar un comportamiento particular del agente) y una o más *metas de mantenimiento* (metas de diagnóstico y funcionamiento corriente del agente) (Vallejo, Ochoa & Jiménez, 2009).

6.3.4 Propiedades de los agentes: Dependiendo del tipo de agentes estos pueden presentar una o varias de las muchas características existentes, a continuación solo se mencionan las características relevantes para este proyecto:

- Orientación a metas: Cuando las metas condicionan la selección de acciones, en la mayoría de los casos la meta requiere una secuencia de acciones.
- Orientación a la utilidad: Si además de lograr una meta, se debe incrementar el desempeño.
- Memoria: Si el comportamiento se basa en la historia de las percepciones y un modelo del ambiente, que suple en parte las deficiencias perceptivas.
- Autonomía: El agente opera sin intervención directa de otros, su conducta se basa en su propia existencia y tiene algún *control sobre sus acciones y su estado interno*.
- Capacidad Social: Interacción con otros agentes mediante algún tipo de comunicación.
- Reactividad: Percibir el ambiente y responder a cambios que ocurren en este.
- Proactividad: Tomar la iniciativa y elegir las acciones que conducen a la meta, la cual se intenta lograr prioritariamente.
- Movilidad: Capacidad de moverse de un entorno a otro por decisión propia.
- Adaptabilidad o aprendizaje: Reaccionar ante una misma situación de formas diferentes, con objeto de hallar comportamientos óptimos orientados a la resolución de cada problema.
- Flexibilidad: Un agente flexible es un agente con capacidad social, proactividad, movilidad y adaptabilidad.
- Inteligencia: Según Wooldridge y Jennings (1995), un agente inteligente debe presentar las características de autonomía y flexibilidad.

6.4 INTELIGENCIA ARTIFICIAL DISTRIBUIDA (IAD)

6.4.1 Enfoques de la Inteligencia Artificial: El trabajo en Inteligencia Artificial (IA) se puede dirigir hacia varios enfoques, en general hay dos escuelas de pensamiento, *IA Convencional* e *IA Computacional*; sin embargo también se habla de *IA Fuerte* e *IA Débil* y de *IA Distribuida* (y su contraparte *IA No Distribuida*).

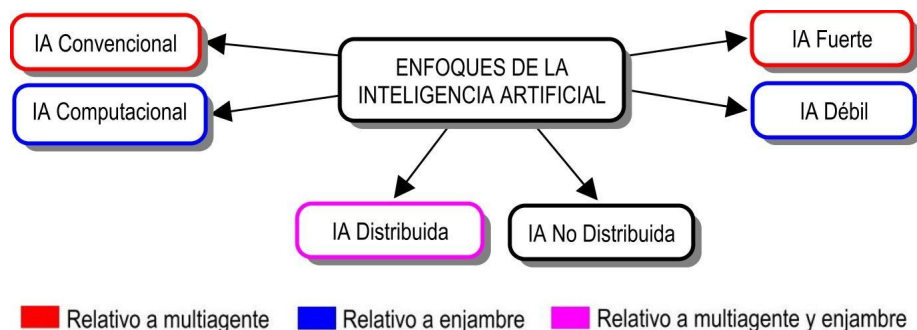


Figura 5. Enfoques de la Inteligencia Artificial.

- **IA Fuerte e IA Débil:** La IA Fuerte, *IA Completa* o *IA General* es la inteligencia de una (hipotética) máquina que podría realizar con éxito cualquier tarea intelectual de un ser humano. Los problemas de IA Completa no se pueden resolver solo con la tecnología actual, sino que requieren también computación humana. Por otro lado, la IA Débil, *IA estrecha* o *IA aplicada*, pretende ser aplicable a un tipo determinado de problemas y no intenta simular toda la gama de capacidades cognitivas humanas; la hipótesis de la IA Débil es que las máquinas pueden demostrar inteligencia, pero no necesariamente tienen una mente, estados mentales o conciencia.
- **IA Convencional e IA Computacional:** La *IA Convencional*, también llamada *IA simbólico-deductiva* o *IA Simbólica*, supone que muchos aspectos de la inteligencia humana se logran mediante símbolos (conceptos) que se usan para categorizar el entorno y razonar acerca de él; estudia máquinas capaces de contener modelos *simbólicos* de su entorno y efectuar en ellos manipulaciones sofisticadas. Dentro de la IA Convencional se encuentran técnicas como: *Sistemas Expertos*, *Razonamiento Basado en Casos*, *Redes Bayesianas* y *Administración Inteligente de Procesos*. Por otra parte, la IA Computacional, también llamada *IA subsimbólica-inductiva* o *IA Conexionista*, se basa en que las *conexiones* neuronales permiten asociar respuestas apropiadas a los estímulos que se perciben; intenta imitar el sustrato emergente de la inteligencia reproduciendo solamente sus propiedades observables. Dentro de la IA Computacional se encuentran técnicas como: *Redes Neuronales*, *Computación Evolutiva*, *Inteligencia de Enjambre*, *Sistemas Inmunes Artificiales*, *Sistemas Difusos*, *Fractales*, *Teoría del Caos*, *Wavelets* y *Autómatas Celulares*.

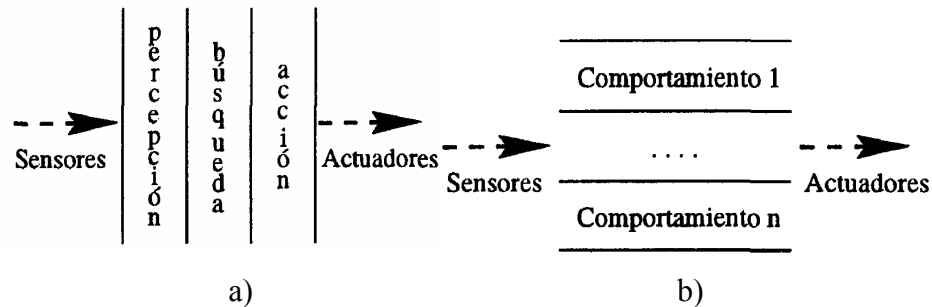


Figura 6. a) Procesamiento por pasos sucesivos característico de la IA Convencional. b) Procesamiento paralelo característico de la IA Computacional. Fuente: Matellan, V. (2012) ABC2: Un Modelo Para el Control de Robots Autónomos (tesis doctoral). Facultad de Informática de la Universidad Politécnica de Madrid.

En la IA Convencional las acciones son el resultado de pasos sucesivos: primero se genera un modelo interno del entorno; luego se busca una solución en ese modelo, y por último se aplican las acciones que en el modelo parezcan llevar a la solución. En la IA Computacional cada acción se ve como un enlace directo entre las entradas y las salidas del sistema.

6.4.2 Concepto de Inteligencia Artificial Distribuida (IAD): Es la rama de la IA que estudia sistemas que pueden ser visualizados como una *sociedad de agentes*; intenta resolver problemas en donde una conducta colectiva es más eficiente que una conducta individual.

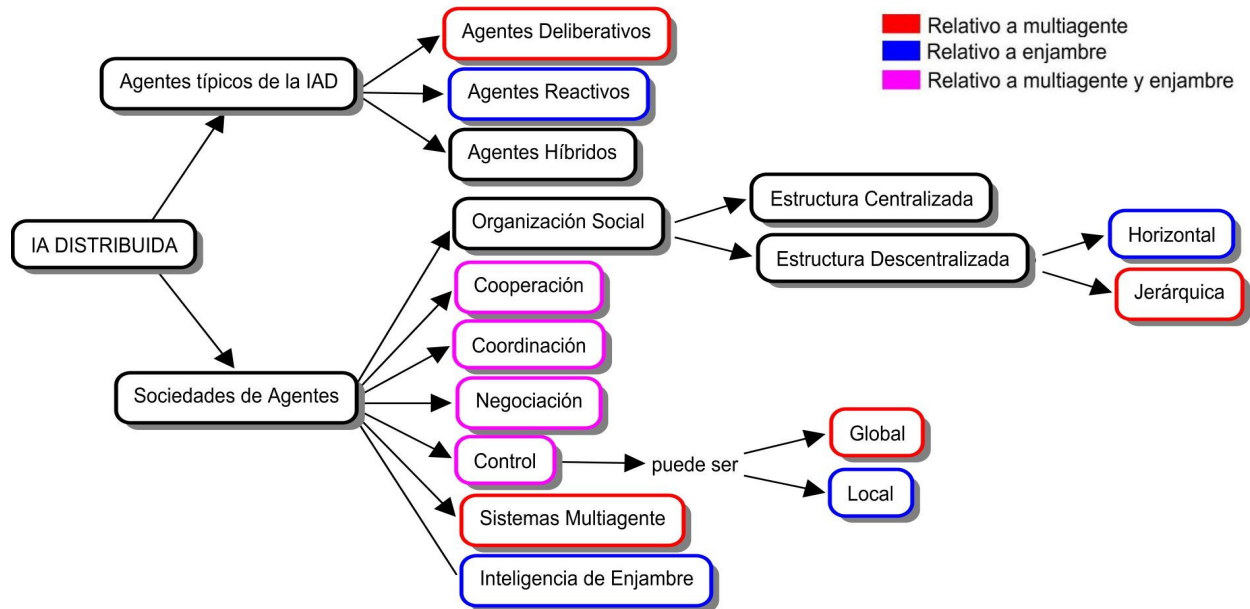


Figura 7. Temas tratados en IA Distribuida. Fuente: Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

La IAD presenta las siguientes ventajas:

- Incremento de la flexibilidad, se permite la adición de nuevos agentes.
- Mejor seguridad y efectividad, los agentes se pueden especializar en una tarea específica.
- Mejor tiempo de respuesta, puede resolver sus problemas particulares al mismo tiempo.
- Reducción de la complejidad, una tarea puede ser descompuesta en varias subtareas.
- Reutilización, la solución de un agente en un sistema puede ser incorporada en otro sistema.

En el contexto de la IAD, la investigación se efectúa en el entorno donde se encuentran todas las entidades de estudio. A continuación se definen los componentes del entorno (D'Inverno & Luck, 2007):

- *Entorno*: Colección de entidades.
- *Entidad*: Colección de atributos, definidos como características perceptibles del entorno.
- *Objeto*: Una entidad con capacidades, acciones que se pueden ejecutar en el entorno.
- *Agente*: Es un objeto con metas, definidas como estados de asuntos que deben ser logrados en el entorno; los demás objetos son llamados objetos neutrales.
- *Agente Autónomo*: Es un agente con motivaciones, definidas como deseos o preferencias que pueden conducir a la generación de metas; un agente Autónomo adopta metas de forma

voluntaria, los demás agentes son llamados *agentes servidores* y sus metas son asignadas por otros agentes (*agentes observadores*).

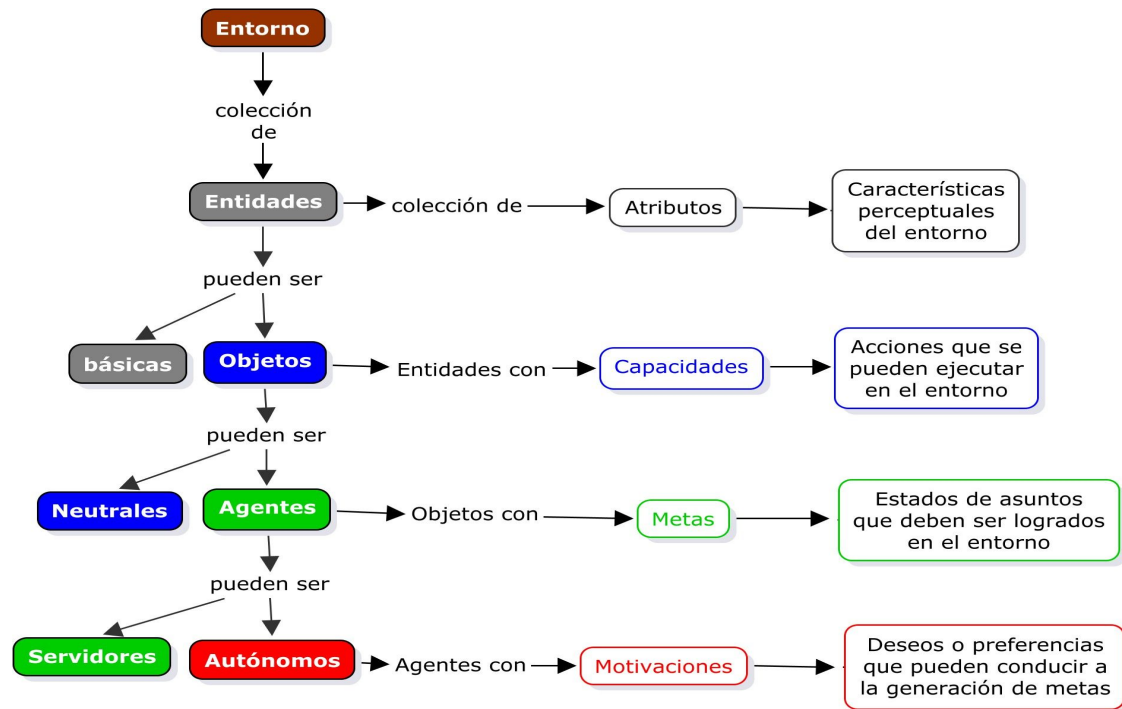


Figura 8. Componentes de los entornos y sus características. Fuente: Adaptado de D’Inverno, M., &Luck, M. (2007). Understanding Agent Systems. Springer-Verlag Berlin.

6.4.3 Agentes típicos de la IAD:

Según Russell & Norvig (2009), existen tres clases típicas de agentes:

- **Agente Deliberativo:** En este tipo de agentes se enfatiza la autonomía y la cooperación para ejecución de tareas, su operación se basa en modelos del entorno en el que se encuentra. Las críticas a estos agentes son por su uso del paradigma de *razonamiento deliberativo*, con el cual cada acción se determina tras un análisis o procesamiento de la información disponible por el agente, esto puede provocar tiempos de respuesta excesivos ante ciertas situaciones.
- **Agente Reactivo:** Es un agente que percibe las características del entorno como estímulos, que le permiten elaborar una respuesta directa y rápida, sin necesidad de representaciones internas del entorno. Las ideas que caracterizan a estos agentes son:
 - El mejor modelo del entorno es el entorno mismo (Brooks, 1990).
 - Descomposición de tareas, el agente puede verse como una colección de módulos que actúan de forma independiente en tareas específicas.
 - Comportamiento emergente

Las ventajas de utilizar agentes reactivos son:

- Mayor robustez.

- Mejor tolerancia a fallos (un agente puede fallar en el sistema sin resultados catastróficos).
- Mayor flexibilidad y adaptabilidad.
- Menor tiempo de respuesta.
- **Agentes Híbridos:** Combinación de dos o más filosofías de agentes en uno solo. La tendencia es implementar estos agentes para aprovechar las ventajas que ofrece cada una de las filosofías de agentes, dependiendo de las situaciones específicas para las que son diseñados.

6.4.4 Sociedades de Agentes: Las sociedades de agentes presentan características particulares: *organización social, cooperación, coordinación, negociación y control.*

- **Organización social:** Se relaciona con la estructura de los componentes del sistema, sus características, responsabilidades, necesidades y forma de comunicación:
 - Estructura Centralizada: Un agente controla los demás, se requiere un conocimiento global del entorno y un diseño preciso, para considerar todos los posibles estados del sistema.
 - Estructura Descentralizada: Cada agente percibe el ambiente y puede tomar decisiones de manera autónoma sin instrucciones de un control central; esta estructura puede ser:
 - *Estructura Horizontal:* Todos los agentes están al mismo nivel; cada uno divide el problema en sub-problemas que puede resolver, con la cooperación de los demás.
 - *Estructura Jerárquica:* Los agentes trabajan diferentes niveles de abstracción de un problema; cada agente divide el problema en sub-problemas así: los que puede resolver él solo, los que puede resolver con cooperación de agentes del mismo nivel, y los que pueden resolver los agentes de niveles inferiores.
- **Cooperación:** Para cooperar de manera eficiente, cada agente debe:
 - Comunicarse de alguna manera con los demás agentes.
 - Integrar información de otros agentes con la suya, para formar conceptos globales o conocimiento conformado por varios agentes.
 - Interrumpir un plan que se esté llevando a cabo, para atender a otros agentes y cooperar cuando sea necesario.
- **Coordinación:** Mecanismo mediante el cual se disponen metódicamente los componentes del sistema para unir esfuerzos que contribuyan al objetivo; permite considerar las tareas a realizar para no ejecutar acciones no deseables. La coordinación implica sincronización de acciones e intercambio de información; esto depende de los objetivos del sistema, las características de los agentes y el ambiente en el que actúan (Chaimowicz, Kumar, & Campos, 2004). La coordinación está relacionada con la planificación de acciones para la resolución de tareas, ya que estos planes permiten:
 - Conocer a alto nivel y predecir el comportamiento de otros agentes.
 - Intercambiar resultados intermedios que contribuyan a la tarea global.
 - Evitar acciones redundantes, si ellas no son deseables.
- **Negociación:** Mecanismo por medio del cual los agentes se ponen de acuerdo cuando cada uno defiende sus propios intereses. Los procesos de negociación tienen como resultado la modificación o confirmación de las creencias de cada agente involucrado, en lo relacionado

con los demás agentes y con el mundo en el que se desenvuelve. La negociación se caracteriza por tener los siguientes elementos:

- Agentes: Un número adecuado de agentes involucrados en el proceso.
- Lenguaje: Un conjunto mínimo de acciones que se llevan a cabo en el proceso, como: proponer, evaluar, refutar, contraproponer, aceptar, rechazar, modificar, etc.
- Protocolo: Establece un conjunto de pasos que debe seguir un proceso de negociación, así como las posibles respuestas de un agente, a las acciones de otro agente.



Figura 9. Planeación y Cooperación en Sistemas Multiagentes Robóticos. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

- **Control:** Es el mecanismo básico que provee apoyo para la implementación de mecanismos de coordinación en un SMA; puede ser considerado desde dos puntos de vista:
 - Control global: Se relaciona con toma de decisiones basada en datos consolidados a partir de la información de todos los agentes del sistema.
 - Control local: Se relaciona con toma de decisiones basadas solo en datos locales disponibles por cada agente.

Estos dos controles deben ser balanceados por varias razones:

- El aumento del poder de control local lleva a rapidez en la toma de decisiones y en su ejecución, pero se puede llegar a la pérdida de mecanismos de cooperación efectivos y como consecuencia a la realización de tareas no deseables.
- El aumento del poder del control global lleva a aumentar los costos de computación del sistema, porque la información de los agentes cambia dinámicamente, como consecuencia para mantener información global coherente, se deben revisar e incorporar constantemente los cambios en cada uno de los agentes.

Para mantener el balance entre el control global y el control local, los componentes del control local de cada agente deben:

- Proveer al control cooperativo una descripción de su estado actual y actividades esperadas.
- Interactuar con el control cooperativo para modificar las actividades locales y estar más "inline" con las necesidades de los otros agentes.

6.4.5 Sistemas Multiagente (SMA): Son una rama de la IAD que intenta resolver problemas mediante interacciones entre agentes, que poseen sus propios intereses y objetivos (Stone &

Veloso, 2000; Cao et al., 1997) y que además exhiben un *comportamiento cooperativo* (Cao et al., 1997).

Un SMA se caracteriza por (D'Inverno & Luck, 2007):

- Está compuesto por varios agentes, no necesariamente iguales.
- Debe tener al menos un agente Autónomo, con el objetivo de que surjan metas.
- Se presenta un comportamiento Colaborativo.
- Se usa comunicación directa.
- Se establece cooperación explícita, necesaria al no haber metas globales.

Debido a que estos sistemas deben contener al menos un agente autónomo, se deduce que un Sistema Multiagente presenta:

- Complejidad Individual de algunos agentes o todos: estos pueden tener información sensorial global, realizan acciones complejas, tienen algún grado de memoria; capacidad de localización, capacidad de aprendizaje, conocimiento del estado del grupo o de su objetivo.
- Ejecución recurrente de tareas: los agentes ejecutan sus tareas de acuerdo a las metas generadas o asignadas; se realiza el trabajo de forma síncrona.

6.4.6 Inteligencia de Enjambre (*Swarm Intelligence, SI*): Con este término se denomina el comportamiento inteligente que proviene de la unificación (más que de una aglomeración) de agentes que por sí solos no tienen más que un comportamiento limitado, pero que en conjunto son capaces de desplegar un comportamiento dinámico (Russell & Norvig 2009). Inteligencia de Enjambre es quizás la técnica más representativa de la *Resolución de Problemas Distribuida (RPD)*, que es la rama de la IAD que se centra en conjuntos de agentes con *Comportamiento Eusocial*¹. El objeto de estudio de la RPD son los sistemas conformados por un gran número de agentes, relativamente simples, que al interactuar entre ellos y con su entorno logran generar comportamientos complejos (Dorigo & Sahin 2004). El concepto fue introducido por Beni, Hackwood y Wang (1993), cuando estaban investigando las propiedades de agentes auto-organizables simulados, en sistemas de robots móviles. Eric Bonabeau, Marco Dorigo, y Guy Theraulaz extendieron el concepto de inteligencia de enjambre para incluir "cualquier intento de diseñar algoritmos o dispositivos distribuidos para resolución de problemas inspirados en el comportamiento colectivo de las colonias de insectos sociales y otras sociedades animales" (Garnier et al., 2007). Los sistemas robóticos basados en este tipo de comportamiento hacen parte de la denominada *Robótica de Enjambre*.

Un Sistema de Inteligencia de Enjambre posee las siguientes características (Martinoli, 2001A; Floreano & Mattiussi, 2008):

- Está compuesto por muchos agentes idénticos.
- Todos los agentes deben ser reactivos.
- Se presenta un comportamiento Eusocial.
- Se usa comunicación indirecta (estigmergia).
- Se establece cooperación implícita, no hay metas globales sino intenciones.

¹Ver "cooperación implícita" en la sección 6.1.6 Relaciones e interacciones, pág 25

Debido a que en estos sistemas todos los agentes son reactivos, se deduce que un Enjambre presenta:

- *Simplicidad Individual de todos los agentes*: Solo tienen información sensorial local, realizan acciones sencillas, tienen poca o ninguna memoria; capacidades limitadas de localización, y de aprendizaje, no tienen conocimiento del estado del enjambre o de su objetivo.
- *Ejecución concurrente de tareas*: Cada agente ejecuta sus tareas sin depender de la ejecución de las tareas de otros individuos; es decir, realiza su trabajo de forma asíncrona.
- *Comportamiento Emergente*: El enjambre puede resolver problemas complejos que un solo individuo con habilidades simples (computacionales o físicas) no puede resolver.

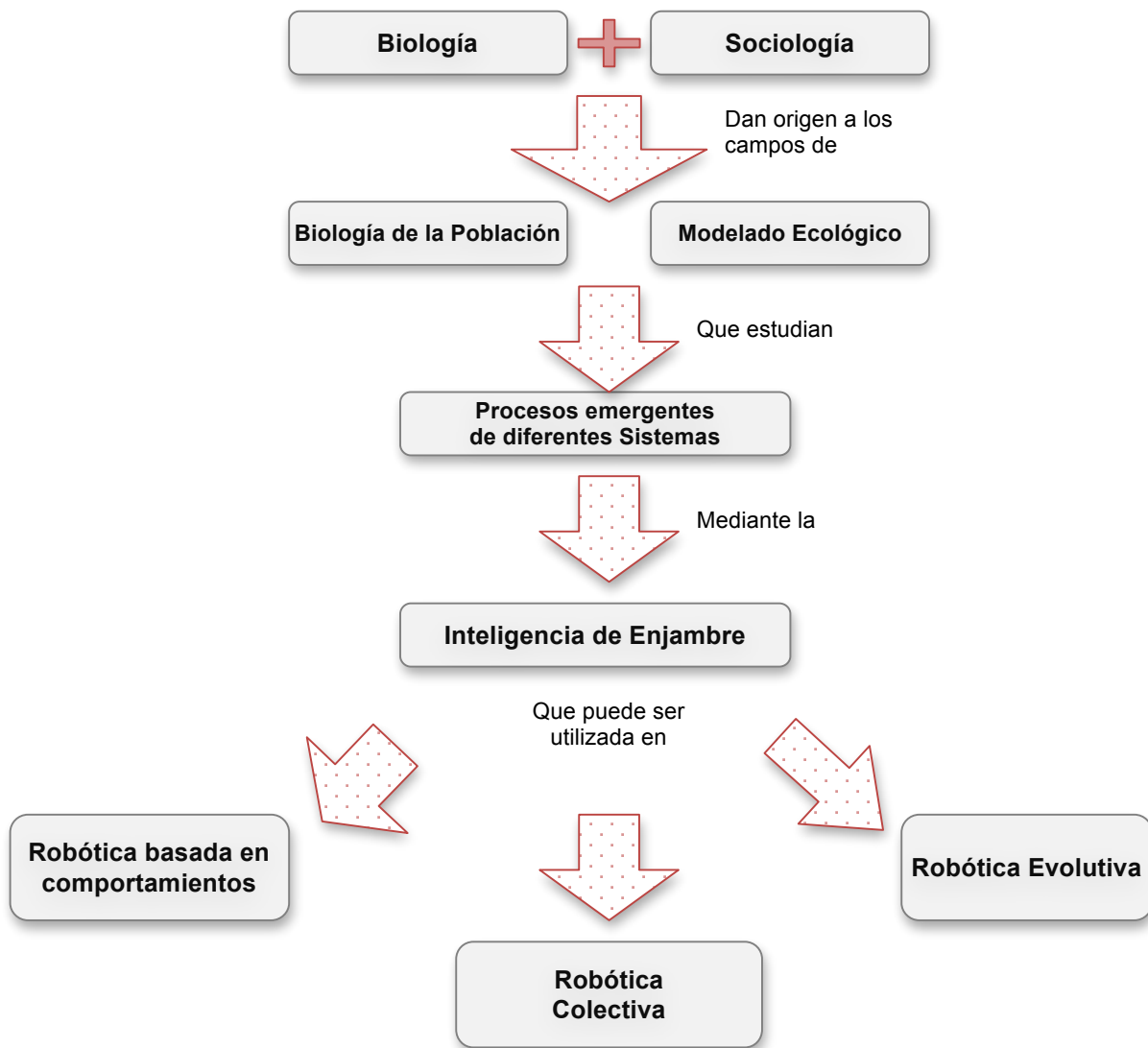


Figura 10. Inteligencia de enjambre como estrategia de robótica colaborativa. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

6.5 ARQUITECTURAS DE CONTROL

En el contexto de la robótica, arquitectura se refiere a la organización, función e interacción de los componentes (hardware y software) que integran un robot (Aldana, 2009); es el marco de trabajo sobre el que estos se encajan para alcanzar el comportamiento previsto (Angulo, Ponsa & Raya, 2006). La arquitectura marca qué estímulos aceptar o buscar, organiza las capacidades de procesamiento, y determina los comportamientos más apropiados de un robot autónomo en cada momento, de acuerdo con su percepción, sus objetivos y las condiciones del entorno. En síntesis, una arquitectura ha de incluir capacidad de percepción selectiva y de actuación adecuada.

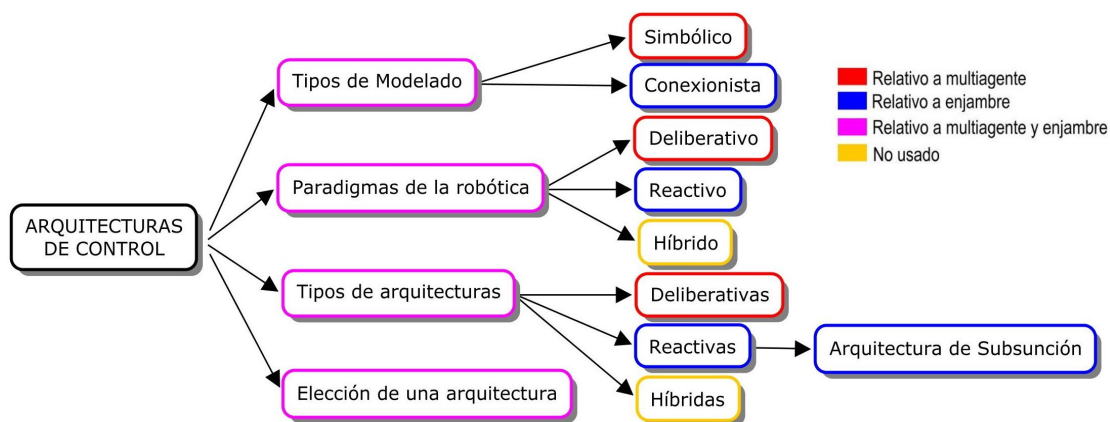


Figura 11. Temas tratados en Arquitectura de Control.

6.5.1 Tipos de modelado: Los modelos computacionales requieren una representación formal matemática y lógica de un problema y son utilizados en la simulación y verificación experimental de diferentes propiedades específicas y generales de la inteligencia.

- **Modelado simbólico:** Se centra en las funciones mentales abstractas de una mente inteligente y opera con símbolos. Fue desarrollado por los primeros investigadores cognitivos y posteriormente utilizado en los sistemas expertos; usa las tecnologías de los sistemas basados en el conocimiento, así como una perspectiva filosófica.
- **Modelado conexionista:** Se ajusta a las propiedades neuronales y asociativas del cerebro humano; se basa en la idea de que el cerebro se compone de nodos simples y que la potencia del sistema proviene de la existencia y la forma de las conexiones entre los nodos.

6.5.2 Paradigmas de la robótica: Describen las formas de relacionar las tres primitivas de la robótica: *percibir*, *planear* y *actuar* (Arkin, 2000) e indican cómo se procesan y distribuyen los datos sensoriales a través del sistema, y donde se toman las decisiones (Asada & Slotine, 1986).

- **Paradigma Jerárquico/Deliberativo:** Predominante hasta 1985, siendo sus características:
 - El robot funciona de manera jerárquica, desde el punto de vista de la planificación.
 - El robot percibe el entorno, planea la siguiente acción y actúa; a cada paso planea el próximo movimiento.
 - Los los datos de percepción son reunidos en un solo modelo global del entorno.
- **Paradigma Reactivo:**
 - Utiliza una organización de tipo “Planear – Actuar”.
 - El robot maneja varias instancias de la dupla “Planear – Actuar”.
 - Estas duplas son procesos concurrentes, llamados comportamientos, que toman los datos locales de los sensores y calculan la mejor acción a tomar con independendia de lo que los otros procesos que estén haciendo.
 - El robot efectúa una combinación de comportamientos.
- **Paradigma Híbrido (Deliberativo / Reactivo):**
 - El robot planea (delibera) cómo descomponer mejor una tarea en subtareas y determina los comportamientos adecuados para lograr cada subtarea.
 - Ejecuta el comportamiento de acuerdo con el paradigma reactivo.
 - La organización de las percepciones es una mezcla de los estilos deliberativo y reactivo; los datos del sensor son enrutados a cada comportamiento que necesita ese sensor, pero también están disponibles para la construcción de un modelo global del entorno, orientado a tareas.

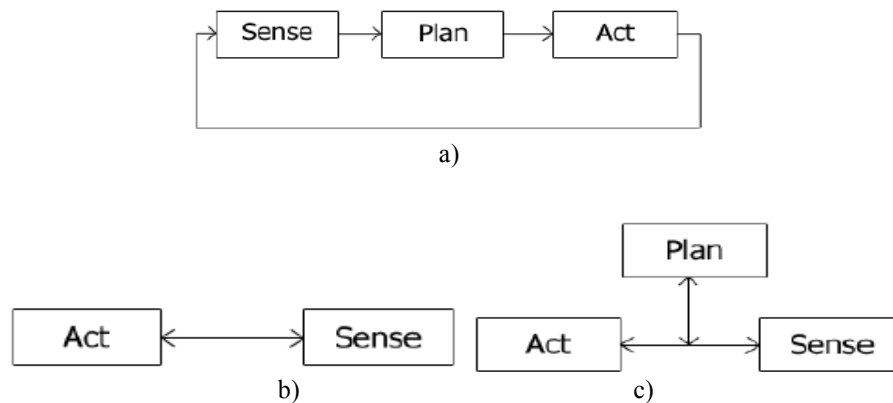


Figura 12. Esquemas de procesamiento según los paradigmas de la robótica: a) Paradigma Deliberativo, b) Paradigma Reactivo, c) Paradigma Híbrido. Fuente: Arkin, Ronald. (2000). Behavior Based Robotics. Lecture Notes in Computer Science, 1790, 103–116.

6.5.3 Tipos de arquitecturas: Existen tres tipos de arquitecturas basadas en los paradigmas de la robótica (Angulo et al., 2006):

- **Arquitecturas Deliberativas:** Basadas en el paradigma deliberativo, Percibir-Planificar-Actuar (Nilsson, 1980), cuentan con una unidad central de proceso que contenga un modelo *simbólico* del entorno, reciba las entradas sensoriales y genere las salidas de los actuadores. Con base en el modelo y las condiciones del entorno, el robot planifica la toma de decisiones.

Ventajas:

- Usan modelos muy exactos y completos del entorno (idealmente esto es una fortaleza).
- Permiten hacer planeaciones, sobre el entorno, a cualquier nivel deseado.

Desventajas:

- Necesitan modelos muy exactos y completos (en términos prácticos es un problema).
- Requieren altas capacidades de cómputo y de memoria.
- Son inadecuadas para tiempo real, la ejecución de acciones puede tomar mucho tiempo.
- No es posible conocer a priori las acciones correspondientes a una percepción.
- Proporcionan dificultad de operación en entornos dinámicos o desconocidos.

- **Arquitecturas Reactivas:** Basadas en el paradigma reactivo (Brooks, 1986A), no requieren procesador central, en su lugar cuentan con un *acople* muy cercano entre sensores y actuadores (Brooks, 1991A); el robot ejecuta acciones con base en las condiciones del entorno; su capacidad es responder a los estímulos por lo que no requiere representación del ambiente ni planeación de acciones (Brooks, 1990). Este tipo de arquitectura es denominada “*Basada en Comportamientos*” (Behavior Based).

Ventajas:

- Adecuadas para operación en tiempo real, altas velocidades de respuesta.
- Las acciones son predecibles para cada percepción.
- Utilizan bajos recursos computacionales.
- Brooks (1990) afirma que la información proviene del entorno (el mejor modelo de este él mismo).
- Confiables en entornos desconocidos y de alto desempeño en entornos dinámicos.

Desventajas:

- Dependen de la calidad de las señales provenientes de los sensores
- No permiten el desarrollo de sistemas para tareas complejas.
- No acumulan experiencias, no es posible el aprendizaje o la adaptación.

- **Arquitecturas Híbridas:** Presentan una mezcla entre los paradigmas reactivo y deliberativo (Arkin, 1990); han tenido gran acogida ya que combinan la rápida respuesta de las arquitecturas reactivas con la toma de decisiones basada en modelos internos de las arquitecturas deliberativas.

Las arquitecturas híbridas son propicias para una estructuración por capas, que puede ser:

- **Estructuración vertical:** Sólo una capa tiene acceso a los sensores y actuadores.
- **Estructuración horizontal:** Todas las capas tienen acceso a sensores y actuadores.

Las capas se organizan jerárquicamente con información sobre el entorno, a diferentes niveles de abstracción; el comportamiento global del agente viene definido por la interacción entre estos niveles:

- **Nivel Reactivo:** El más bajo nivel, en él se toman decisiones acerca de lo que se debe hacer, con base en los estímulos recibidos del entorno en tiempo real.
- **Nivel de Conocimiento:** El nivel intermedio, se centra en el conocimiento que el agente posee del entorno, normalmente con ayuda de una representación simbólica del mismo.
- **Nivel Social:** Es el más alto nivel, maneja los aspectos sociales del entorno, incluyendo tanto información de otros agentes, como los deseos, intenciones, etc.

6.5.4 Elección de una arquitectura: Cuanto más complejo es un sistema más relevancia cobra la organización de sus componentes; para lograr sistemas repetibles y adaptables su organización interna ha de ser clara, accesible y fácilmente modificable; desafortunadamente, los sistemas robóticos son complejos y difíciles de controlar, y por lo tanto el diseño de arquitecturas robóticas convenientes es aún un campo importante de investigación.

El problema principal que debe ser resuelto en la investigación de agentes es llegar a una *arquitectura* para agentes autónomos, de manera que el agente demuestre un comportamiento con las siguientes características:

- **Adaptativo:** El agente mejora su competencia en resolución de problemas con el tiempo.
- **Robusto:** El sistema nunca colapsa completamente, sino que demuestra una degradación lenta cuando fallan componentes del agente o cuando suceden imprevistos.
- **Efectivo:** El agente es exitoso al cumplir sus metas.

No existe la arquitectura de control válida para todos los entornos y comportamientos, incluso un animal con escasa capacidad de aprendizaje tiene un repertorio de comportamientos limitado, aunque amplio; su arquitectura le provee de un comportamiento y no de otro, aunque puede tener el “hardware” necesario para ambos. Los criterios que pueden seguirse para comparar y evaluar diferentes arquitecturas son los siguientes (Arkin, 2000):

- **Modularidad:** Acerca de la independencia de los distintos componentes de la arquitectura así como su capacidad de reutilización y escalabilidad.
- **Adaptabilidad:** Cómo se adapta la arquitectura en la aplicación a la cual ha sido pensada.
- **Portabilidad:** Facilidad de portabilidad a otros dominios, es decir, cómo se adapta la arquitectura a otras aplicaciones.
- **Robustez:** Acerca de la vulnerabilidad de la arquitectura.

6.5.5 Arquitectura de Subsunción (*Subsumption Architecture*). Fue propuesta por Rodney Brooks (1986B), basándose en teorías que explican el comportamiento animal; teniendo en cuenta que la actividad de los seres vivos puede modelarse como un conjunto de comportamientos elementales caracterizados por una reacción a un conjunto de estímulos que vienen del medio ambiente.

Con base en esto, Brooks (1986A) propone ciertos aspectos que debía cubrir la IA:

- No puede haber una noción tradicional de planeación
- No es necesaria una representación central
- Las nociones de modelación del mundo son imprácticas e innecesarias
- La biología y la evolución son buenos modelos a seguir.
- Es necesaria la construcción de sistemas completos que existan en el mundo real.

Con la Arquitectura de Subsunción (Brooks, 1986B), la cual es representante del *paradigma reactivo*, los comportamientos del micro-robot se presentan en capas, donde las capas inferiores son comportamientos básicos. Un comportamiento de una capa superior puede inhibir uno de una capa inferior de acuerdo a la acción resultado de una sensación, pero no necesariamente un nivel superior suprime a todos los inferiores a él. Cada capa se vale de la información proporcionada por el grupo de sensores que requiera y proporciona información a los actuadores para la ejecución de su tarea; la advertencia principal es que las tareas separadas pueden suprimir (o anular) las entradas o inhibir salidas.

La Arquitectura de Subsunción es una manera de descomponer el complicado comportamiento inteligente en muchos módulos "simples" de comportamiento. Cada capa implementa un objetivo particular del agente, y las capas más altas son cada vez más abstractas. El objetivo de cada capa *subsume* el de las capas subyacentes; de esta manera, las capas más bajas pueden funcionar como mecanismos de rápida adaptación, mientras que las capas superiores trabajan para lograr el objetivo general. La realimentación se da principalmente a través del entorno.

6.6 ROBÓTICA MÓVIL

Se puede definir robot móvil como un sistema electromecánico controlado, capaz de trasladarse de forma *autónoma* a una meta preestablecida en un determinado espacio de trabajo (Ollero, 2006).

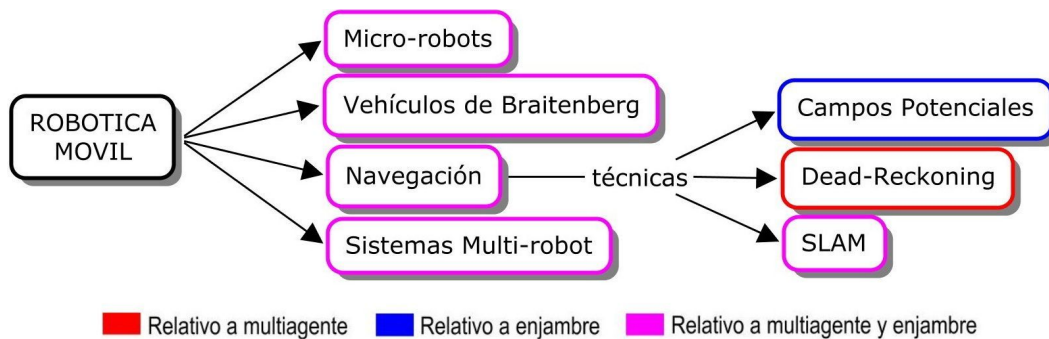


Figura 13. Temas tratados de Robótica Móvil. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

Se entiende por autonomía de un robot móvil, a la capacidad para determinar su curso de acción mediante un razonamiento propio, en lugar de seguir una secuencia fija de instrucciones; para esto debe poseer 3 sistemas fundamentales (Ortega, 1994; Murphy, 2000; Posadas, 2003):

- **Sistema de percepción**, que permite tener conocimiento de las características de su entorno.
- **Sistema de planificación**, que define las acciones que deben ser llevadas a cabo, en función de la tarea y de la información sensorial.
- **Sistema de ejecución o acción**, que le permite ejecutar las tareas mediante la interacción con los actuadores.

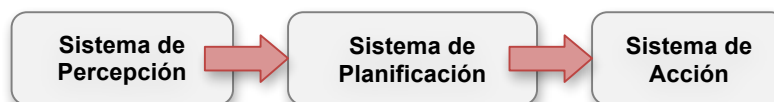


Figura 14. Aproximación clásica de la Robótica. Fuente: Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

El robot móvil autónomo se caracteriza por una conexión *inteligente* entre las operaciones de percepción y acción, que define su comportamiento y le permite llegar a la consecución de los objetivos programados sobre entornos con cierta incertidumbre (Jaramillo, 2010). De este modo, las dos grandes características que diferencian a un robot móvil de cualquier otro tipo de vehículo son (Lozano-Pérez, 1990):

- **Percepción:** Determina la relación del robot con su entorno mediante el uso de los sensores.
- **Razonamiento:** Determina las acciones que se han de realizar en cada momento, según el estado del robot y su entorno, para alcanzar las metas asignadas.



Figura 15. Modo general de un control robótico. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

Los robots móviles se pueden clasificar, de acuerdo al medio en que se desplacen, en terrestres, acuáticos y aéreos (Ollero, 2006); siendo los terrestres los más estudiados y dentro de los cuales se puede hacer una nueva clasificación de acuerdo al tipo de locomoción utilizado: por ruedas (Muir & Newmanl, 1987), por patas (Todd, 1985) y por orugas (Grasnosik & Borenstein, 2005). El mayor desarrollo se presenta en los Robots Móviles con Ruedas (RMR), pues su construcción es más sencilla y presentan mayor eficiencia en cuanto a energía en superficies lisas y firmes.

6.6.1 Micro-robots: Son la tendencia más sobresaliente en la robótica móvil colaborativa, emplean técnicas *basadas en comportamientos* combinadas con técnicas tradicionales de inteligencia artificial; esto permite *reacciones inmediatas* ante *situaciones inesperadas*. Los micro-robots no son necesariamente pequeños en tamaño sino en capacidad de procesamiento, sin embargo se ha comprobado su funcionalidad y en ocasiones se les ha observado un desempeño más eficiente que los sistemas expertos. Las razones son principalmente:

- El control se basa en la superposición de los controles individuales, por lo que no es necesario un control tan complejo.
- Son unidades de estudio muy adecuadas para los investigadores, ya que estos últimos pueden concentran sus esfuerzos en aplicar sistemas de control reducidos pero sofisticados.

La tecnología actual ha permitido incorporar técnicas de control muy amplias en los micro-robots, sin que estos dejen de serlo. Su simplicidad no tiene nada que ver con las técnicas de implementación pues el avance tecnológico hace que hoy sea sencillo lo que antes no lo era.

6.6.2 Vehículos de Braitenberg: Concepto ideado por Valentino Braitenberg (1986) para ilustrar las habilidades de un agente inteligente simple. El vehículo representa la forma más

simple de inteligencia artificial, basada en un comportamiento inteligente que emerge de la interacción sensorial-motora entre el agente y su medio, sin necesidad de una memoria interna, representación del ambiente, o inferencia.

Un vehículo de Braitenberg puede moverse en su entorno de manera autónoma. Tiene sensores para percibir estímulos y ruedas dirigidas por sus propios motores. Dependiendo de cómo los sensores y las ruedas están conectados, los vehículos exhiben diferentes comportamientos, de esta forma parecen esforzarse por alcanzar determinadas situaciones y evitar otras, cambiando de rumbo cuando la situación cambia. En un entorno con varias fuentes de estímulos, los vehículos de Braitenberg exhiben un comportamiento complejo y dinámico. Dependiendo de las conexiones entre sensores y actuadores, un vehículo de Braitenberg presenta un comportamiento que está indudablemente dirigido a un objetivo, es flexible y adaptativo.

6.6.3 Navegación: Todo robot móvil que necesite alcanzar un destino debe ser capaz de conocer su posición global y poder definir una trayectoria libre de colisiones con los obstáculos presentes del ambiente. Para esto, es necesario programar al robot de forma de utilizar una o más estrategias de acuerdo a las necesidades que se tengan sobre dichas trayectorias (Holland, 2004).

- **Campos Potenciales (Force Fields):** Considera al robot como una partícula bajo la influencia de un campo potencial artificial, producido por la configuración final y los obstáculos, y cuyas variaciones modelan el espacio libre. La configuración final genera un potencial de atracción que lleva al robot hacia la meta, mientras que los obstáculos producen un potencial de repulsión que aleja al robot de ellos (Espitia & Sofrony, 2012).

El concepto de los campos se puede extender para utilizarse como atractores a un determinado objeto. La función potencial en un punto del espacio consiste en la composición del potencial atractivo y del potencial repulsivo. El gradiente negativo del potencial total se trata como una fuerza artificial aplicada al robot, por lo que en cada configuración la dirección de esta fuerza se considera como la dirección más oportuna de movimiento. El potencial de atracción disminuye conforme el robot se acerca a la configuración final, mientras que el potencial de repulsión es un valor no cero a cierta distancia del obstáculo y tiende a infinito conforme el robot se acerca a este. La ruta encontrada, entre las configuraciones inicial y final, se genera siguiendo el gradiente negativo del potencial total.

El método es eficiente, pero tiene la desventaja de que la función potencial puede caer en un mínimo local en lugar de que en la configuración final, es decir la fuerza de atracción y repulsión en algún punto son de igual intensidad provocando que el robot no se mueva.

- **Dead Reckoning:** (Navegación por Estima) es una expresión derivada del término náutico *deduced reckoning* (cálculo basado en inferencia), un procedimiento matemático para calcular la ubicación actual de un barco con base en el rumbo y la velocidad de navegación en un período, sin usar puntos de referencia. Es una técnica usada hoy por la mayoría de sistemas de robots móviles terrestres. La utilización adecuada de *Dead Reckoning* en este sentido sería proporcionar un porcentaje conocido de energía a los motores del robot, en una cantidad determinada de tiempo, desde un punto de partida general. La técnica no es totalmente exacta, lo que puede conducir a errores en las estimaciones de distancia, con base en duración del recorrido, velocidad del robot, longitud de la trayectoria y otros factores.

Dead Reckoning es utilizada en algunas aplicaciones de gama baja, con misiones no críticas o fuertemente restringidas por el tiempo o por el peso.

- **SLAM**, *Simultaneous Localization And Mapping (Localización y Mapeado Simultáneos)*, es una técnica usada por vehículos autónomos para construir un mapa de un entorno desconocido en el que se encuentra, a la vez que estima su trayectoria al desplazarse dentro del mismo. SLAM busca resolver los problemas que plantea el colocar un robot móvil en un entorno y posición desconocidos, para que él mismo sea capaz de construir incrementalmente un mapa del entorno, al tiempo que lo utiliza para determinar su propia localización. SLAM investiga los problemas que plantea la construcción de modelos matemáticos, geométricos o lógicos de entornos físicos, empleando como herramienta uno o varios robots móviles, y los sensores y actuadores que los conforman. El ruido presente en los sistemas sensoriales, los inevitables errores y aproximaciones cometidos en los modelos empleados, y la dificultad representativa de los entornos a medida que éstos aumentan en complejidad, hacen que la tarea de resolver el problema de SLAM sea ardua (Bailey & Durrant-Whyte, 2006). Tal complejidad tiene una doble vertiente:
 - **Desde un punto de vista conceptual** se impone la necesidad de razonar en un mundo a veces confuso, en ocasiones dinámico y cambiante, aprehendido mediante sensores que distan mucho de ser perfectos; en estas condiciones se busca obtener y manipular datos acerca del entorno, extraer el conocimiento que sea sustancial para la tarea de su representación, e integrar la información así obtenida del modo más conveniente.
 - **Desde un punto de vista computacional** de las soluciones planteadas al problema del SLAM, indisolublemente ligada a la anterior, el modo en que el robot perciba su entorno, la cantidad de información disponible, las técnicas empleadas en su procesamiento, interpretación y combinación, determinarán los recursos computacionales necesarios para la construcción del mapa. Estos recursos son limitados; más aún si el objetivo es ceñirse a los disponibles a bordo de la máquina.

Así pues, en la base de cualquier solución al problema del SLAM siempre se ve la necesidad de trabajar con cantidades progresivamente crecientes de información, contaminada en mayor o menor medida por ruido, y manipulada mediante modelos que, la mayoría de las veces, son solo meras aproximaciones a la realidad. No es de extrañar que las soluciones más exitosas hasta el momento hayan estado basadas en la *utilización de técnicas probabilísticas*.

6.6.4 Sistemas Multi-Robot (SMR): Son grupos de robots homogéneos o heterogéneos² que tiene las características de coordinación, comunicación, colaboración y conciencia (Verret, 2005). La coordinación es el aspecto más delicado a la hora de implementar un SMR, no solo por el número de robots sino por la necesidad de que trabajen juntos para cumplir eficientemente sus objetivos; a esto se suman múltiples requerimientos que adicionan complejidad (Aldana, 2009). Los sistemas multi-robot frecuentemente aparecen mencionados en la literatura como *sistemas robóticos multiagente*, se tiende a identificar o asemejar los términos “*multi-robot*” y “*multiagente*” (Aldana, 2009); sin embargo los sistemas multi-robot no siempre

²Un grupo de robots es *homogéneo* si los robots comparten características y capacidades similares como, capacidad de procesamiento, mecanismos de comunicación, dispositivos sensoriales y de actuación etc.; en caso contrario el grupo es *heterogéneo* (Aldana, 2009).

están basados en Sistemas Multiagente (*robótica multiagente*), pues también pueden estar basados en Inteligencia de Enjambre (*robótica de enjambre*); en ambos casos, sin embargo, siempre se asume al robot como un *agente*; en este sentido puede tomarse la definición dada por Russell y Norvig (2009) quienes proponen definir robot simplemente como un *agente artificial, activo, cuyo entorno es el mundo físico*. El término *activo* descarta de esta definición a entes inanimados, el término *artificial* descarta a los seres vivos, y el término *físico* descarta a los agentes de software puros (llamados *softbots*).

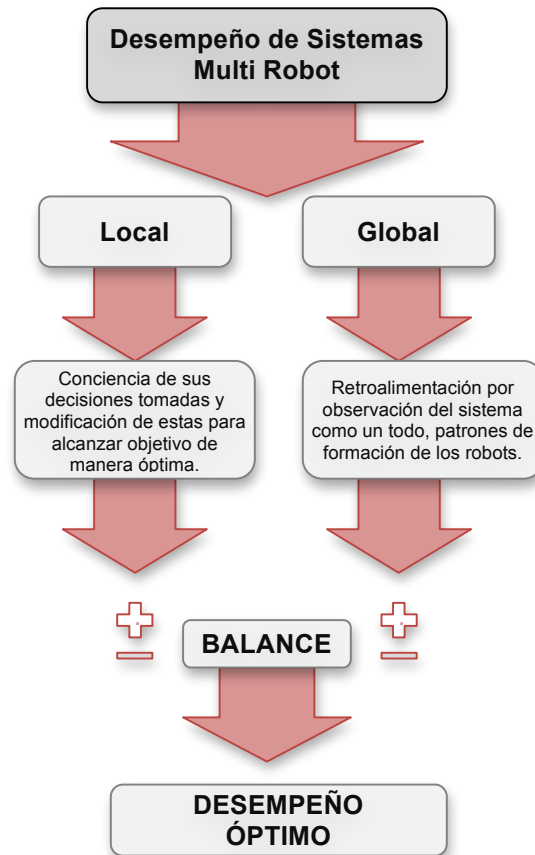


Figura 16. Desempeño de sistemas de robots autónomos descentralizados.
 Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press.

Ventajas de los sistemas multi-robot (Días, 2004; Liu & Wu, 2001):

- Diseñar un solo robot capaz de ejecutar tareas complejas, puede ser difícil y costoso.
- A través de la división del trabajo se puede completar una tarea más eficientemente.
- Puede tener robots diseñados para un solo propósito que permiten cierto grado de especialización en el sistema.
- Robustez, debida a la redundancia de unidades, y al diseño minimalista de las mismas.
- Pueden ser usados para monitoreo y exploración en lugares peligrosos para el ser humano y de difícil acceso para robots convencionales.

- Utilizan tecnología de fácil manipulación, su mantenimiento no requiere de especialización.
- Escalabilidad, pues es la misma arquitectura de control para unos pocos o para unos miles.
- Flexibilidad, porque las unidades individuales pueden añadirse o eliminarse sin necesidad de una reorganización explícita por parte del operador.

Problemas relativos a sistemas SMR:

- Aún no está totalmente claro como un sistema de control de SMR puede ser descompuesto en partes más sencillas.
- Las interacciones entre subsistemas separados no se limita exclusivamente a la interacción entre éstos, sino que también éstas se dan con el medio ambiente y viceversa.
- A medida que el sistema crece en complejidad, el número de interacciones potenciales entre los componentes del sistema y el ambiente, aumenta exponencialmente. (Liu & Wu, 2001).

6.7 METAHEURÍSTICAS

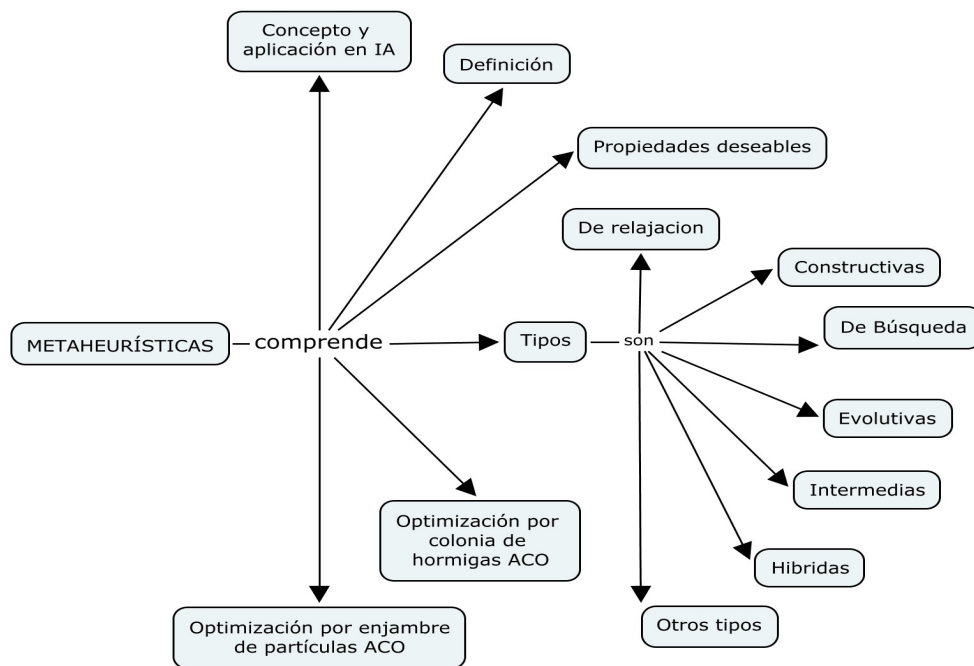


Figura 17: Temas tratados en Metaheurística.

6.7.1 Concepto de Heurísticas y aplicación en la Inteligencia Artificial: En general se puede definir *Heurística* como un conjunto de reglas metodológicas no necesariamente formalizadas, positivas y negativas, que sugieren o establecen cómo proceder y cuáles problemas evitar a la hora de generar soluciones y elaborar hipótesis. Según el matemático George Pólya (2008) la base de la heurística está en la experiencia de resolver problemas y en

ver cómo otros lo hacen. Consecuentemente se dice que hay *búsquedas ciegas*, *búsquedas heurísticas* (basadas en la experiencia) y *búsquedas racionales*.

En Inteligencia Artificial (IA) se emplea el calificativo heurístico, en un sentido muy genérico, para aplicarlo a todos aquellos aspectos que tienen que ver con el empleo de conocimiento en la realización dinámica de tareas. Se habla de heurística para referirse a una técnica, método o procedimiento inteligente de realizar una tarea que no es producto de un riguroso análisis formal, sino de conocimiento experto sobre la tarea. En especial, se usa el término heurístico para referirse a un procedimiento que trata de aportar soluciones a un problema con un buen rendimiento, en lo referente a la calidad de las soluciones y a los recursos empleados.

Una heurística, aplicada a la computación, es un algoritmo que abandona uno o ambos de los objetivos fundamentales en este ámbito (encontrar algoritmos con buenos tiempos de ejecución y llegar a buenas soluciones, usualmente las óptimas); por ejemplo, encuentra buenas soluciones, aunque no hay pruebas de que la solución no pueda ser errónea en algunos casos; o se ejecuta razonablemente rápido, aunque no existe tampoco prueba de que siempre será así.

Las heurísticas generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas (tiempo, espacio, etc.), o cuando no existe del todo. A menudo, pueden encontrarse instancias concretas del problema donde la heurística producirá resultados muy malos o se ejecutará muy lentamente. Aún así, estas instancias concretas pueden ser ignoradas porque no deberían ocurrir nunca en la práctica por ser de origen teórico. Por tanto, el uso de heurísticas es muy común en el mundo real.

Para problemas de búsqueda del camino más corto el término tiene un significado más específico. En este caso una *heurística* es una función matemática $h(n)$, definida *en los nodos de un árbol de búsqueda*, que sirve como una estimación del coste del camino más económico de un nodo dado hasta el nodo objetivo. En cualquier problema de búsqueda donde hay b opciones en cada nodo y una profundidad al nodo objetivo, un algoritmo de búsqueda ingenuo deberá buscar potencialmente entre b^d nodos antes de encontrar la solución. Las heurísticas mejoran la eficiencia de los algoritmos de búsqueda reduciendo el factor de ramificación de b a (idealmente) una constante b^* .

Aunque cualquier heurística admisible devolverá una respuesta óptima, una heurística que devuelve un factor de ramificación más bajo es computacionalmente más eficiente para el problema en particular. Puede demostrarse que una heurística $h_2(n)$ es mejor que otra $h_1(n)$, si $h_2(n)$ domina $h_1(n)$, esto quiere decir que $h_1(n) > h_2(n)$ para todo n .

La tendencia en la IA es extraer la esencia de los procedimientos heurísticos exitosos, para aplicarla a problemas o contextos más extensos, esto ha contribuido al desarrollo del campo de las heurísticas y a extenderla aplicación de sus resultados. De esta forma se han obtenido, tanto técnicas y recursos computacionales específicos, como estrategias de diseño *generales* para procedimientos heurísticos de resolución de problemas. Estas estrategias generales para construir algoritmos, se denominan *metaheurísticas*.

6.7.2 Definición de Metaheurística: Es un método heurístico para resolver un tipo de problema computacional general, usando parámetros dados por el usuario sobre procedimientos genéricos y abstractos, de una manera que se espera eficiente. El término metaheurística lo utilizó por primera vez Fred Glover (1986), en un artículo sobre búsqueda tabú. El nombre

combina el prefijo griego "meta" ("más allá", aquí con el sentido de "nivel superior") y "heurístico" (de εὑρίσκειν, *heuriskein*, "encontrar").

Las metaheurísticas generalmente se aplican a problemas que no tienen un algoritmo o heurística específica que dé una solución satisfactoria; o bien cuando no es posible implementar ese método óptimo. La mayoría de ellas se aplican a problemas de optimización combinatoria, pero pueden aplicarse a cualquier problema que se pueda formular en términos heurísticos. Las metaheurísticas no son la panacea y suelen ser menos eficientes que las heurísticas específicas, en varios órdenes de magnitud, en problemas que aceptan este tipo de heurísticas crudas.

Los conceptos generales relacionados al trabajar con metaheurísticas son:

- **Estado:** Es un objeto matemático finito que maximice (o minimice) una función especificada por el usuario de la metaheurística.
- **Espacio de búsqueda:** El conjunto de todos los estados candidatos. La naturaleza de los estados y del espacio de búsqueda son usualmente específicos del problema.
- **Función objetivo:** Es la función que se desea optimizar, y se da al usuario como un procedimiento '*caja negra*' que evalúa el estado actual o la función.
- **Transición o actualización de estado:** Algunas metaheurísticas mantienen a cada instante de ejecución un único *estado actual*, y lo cambian en cada iteración por uno nuevo. Este paso se conoce como, *movimiento del estado*; el movimiento es *colina arriba* o *colina abajo* dependiendo de si los valores que da la función objetivo se incrementan o se decrementan.
- **Generadores y mutadores:** El nuevo estado puede estar construido desde la nada por un *generador de estados* dado por el usuario. Alternativamente, el nuevo estado puede derivar del estado actual por un *mutador* proporcionado por el usuario. Generadores y mutadores son habitualmente procedimientos probabilísticos.
- **Estado vecino:** El resultado obtenido al aplicar un mutador a un estado.
- **Vecindario:** El conjunto de todos los nuevos estados dados por el mutador.

Algunas metaheurísticas mantienen, en vez de un único estado actual, un *conjunto de varios estados candidato*. Así, el paso básico añade o elimina estados de este conjunto. En este caso, los procedimientos dados por el usuario seleccionan estados para ser descartados, y generan nuevos estados a añadir. El último estado puede ser generado como *combinación* o *cruce* de dos o más estados del conjunto.

Una metaheurística puede guardar información del *óptimo actual*, escogiendo el estado óptimo entre todos los óptimos actuales obtenidos en varias etapas del algoritmo. Dado que el número de candidatos puede ser muy grande, normalmente, las metaheurísticas están diseñadas de manera que puedan ser interrumpidas por un *tiempo máximo* especificado por el usuario. Si no se interrumpen, algunas *metaheurísticas exactas* examinarán todos los candidatos, y usarán métodos heurísticos solo para escoger el orden de la enumeración; de hecho, siempre devolverán un óptimo real, si el *tiempo máximo* es lo suficientemente grande. En cambio, otras metaheurísticas dan solo una garantía probabilística pobre de poder alcanzar el óptimo, de manera que cuando el *tiempo máximo* se aproxima a infinito, la probabilidad de examinar cada candidato tiende a 1.

6.7.3 Propiedades deseables de las metaheurísticas: Son todas las propiedades que favorecen el interés práctico y teórico, indican direcciones a las que dirigir los esfuerzos, algunas son parcialmente contrapuestas y varias apuntan en la misma dirección:

- **Propiedades dirigidas a la comprensión:**

- *Simplicidad:* Debe estar basada en un principio sencillo y claro, fácil de comprender.
- *Precisión:* Sus pasos y fases deben estar formulados en términos concretos.
- *Coherencia:* Sus elementos deben deducirse naturalmente de sus principios.

- **Propiedades dirigidas al rendimiento:**

- *Efectividad:* Debe proporcionar soluciones de alta calidad; óptimas o muy cercanas a estas.
- *Eficacia:* La probabilidad de alcanzar soluciones óptimas de casos realistas debe ser alta.
- *Eficiencia:* Debe realizar un buen aprovechamiento de recursos computacionales; tiempo de ejecución y espacio de memoria.

- **Propiedades dirigidas a la aplicabilidad:**

- *Generalidad:* Debe ser utilizable con buen rendimiento en amplia variedad de problemas.
- *Adaptabilidad:* Debe ser capaz de adaptarse a diferentes contextos de aplicación o modificaciones importantes del modelo.
- *Robustez:* El comportamiento debe ser poco sensible a pequeñas alteraciones del modelo.

- **Propiedades dirigidas a la utilidad:**

- *Interactividad:* Debe permitir que el usuario pueda aplicar sus conocimientos para mejorar el rendimiento del procedimiento.
- *Multiplicidad:* Debe suministrar diferentes soluciones alternativas de alta calidad entre las que el usuario pueda elegir.
- *Autonomía:* Debe permitir un funcionamiento autónomo, libre de parámetros o que se puedan establecer automáticamente.

6.7.4 Tipos de metaheurísticas: Los tipos de metaheurísticas se establecen, en primer lugar, en función del tipo de procedimientos a los que se refiere.

- **De relajación:** Se refieren a procedimientos de resolución de problemas que utilizan modificaciones del modelo original, que hacen más fácil de resolver el problema.
- **Constructivas:** Se orientan a los procedimientos que tratan de la obtención de una solución a partir del análisis y selección paulatina de las componentes que la forman.
- **Debúsqueda:** Guían procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explotar estructuras de entornos asociadas.
- **Evolutivas:** Están enfocadas a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones.
- **Tipos intermedios:** Otras metaheurísticas que aparecen en varias clasificaciones corresponden a tipos intermedios entre los anteriores, entre ellas destacan las metaheurísticas de *descomposición* y las metaheurísticas de *memoria a largo plazo*.

- **Tipos híbridos:** Surgen combinando metaheurísticas de distinto tipo, como la metaheurística *GRASP* (Greedy Randomized Adaptive Search Procedure), que combina una fase constructiva con una fase de búsqueda de mejora, o la metaheurística de *Arranque Múltiple* que incluye una fase de generación de soluciones y una fase de mejora.
- **Otras metaheurísticas:** Se centran en el uso de algún tipo de recurso computacional o formal especial y no se incluyen claramente en ninguno de los cuatro tipos anteriores como las *redes neuronales*, los *sistemas de hormigas* o la *programación por restricciones*.

Cuando se trata de exploración de entornos desconocidos, el uso de algoritmos basados en comportamientos de enjambre ha demostrado incrementar la eficiencia (Kumar, Sahin & Cole, 2004), específicamente las metaheurísticas más empleadas, y que han servido como base para desarrollar otras similares son:

- ABC, Artificial Bee Colony Algorithm (Karaboga, 2005).
- ACO, Ant Colony Optimization (Dorigo, 1992).
- PSO, Particle Swarm Optimization (Kennedy & Eberhart, 1995).
- AIS, Artificial Immune System (Kephart, 1994).
- Métodos de Monte Carlo.

6.8 ALGORITMO ACO (ANT COLONY OPTIMIZATION)

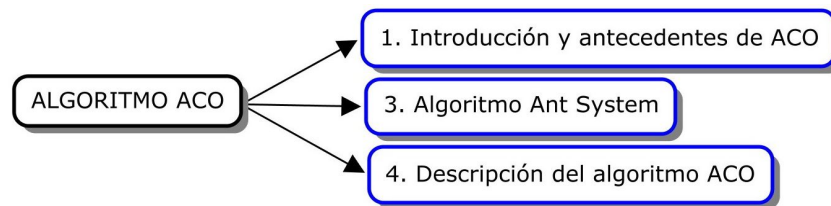


Figura 18: Temas tratados en Algoritmo Ant Colony Optimization.

Los algoritmos basados en colonias de hormigas son útiles en problemas que necesitan encontrar caminos hacia metas con la simulación de hormigas artificiales, a través de agentes, para localizar soluciones óptimas moviéndose en un espacio de parámetros que representa todas las posibles soluciones.

ACO puede funcionar continuamente y adaptarse a los cambios en tiempo real, está basado en dos componentes fundamentales:

- **Retroalimentación positiva:** El rastro de feromonas atrae hormigas que lo fortalecen.
- **Retroalimentación negativa:** Disipación de la feromona, evita la caer en óptimos locales.

6.8.1 Introducción y antecedentes de ACO: Propuesto por Marco Dorigo en 1992 en su tesis de doctorado (Dorigo & Stützle, 2004), usa técnicas probabilísticas para solucionar problemas de cómputo que pueden reducirse a buscar los mejores caminos o rutas en grafos. Este algoritmo está inspirado en el comportamiento que presentan las hormigas para encontrar las trayectorias desde la colonia hasta una fuente de alimento, ya que estas son prácticamente ciegas, sin embargo, los biólogos observaron que las hormigas tienden a usar la ruta más corta que lleva hasta la fuente de alimento, entre rutas con distancias diferentes (Deneubourg, Aron, Goss & Pasteels, 1990).

Las hormigas se desplazan dejando un rastro de feromona que es reconocido por las demás hormigas de la colonia y las estimula a seguirlo, ninguna de ellas sabe donde hay alimento, pero cuando por azar lo encuentran, lo toman y lo llevan al hormiguero por el mismo camino, reforzando así el rastro de feromona. Las demás hormigas se sienten atraídas por la feromona y tienden a seguir el camino con mayor concentración de esta, reforzando aún más dicha concentración (Dorigo & Stützle, 2004).

Teóricamente, si la cantidad de feromona es la misma en todas las rutas durante todo el tiempo, ninguna ruta es elegida. Sin embargo, debido a la retroalimentación, cualquier variación en la concentración de feromona es amplificada y permite la elección de una ruta. El algoritmo pasará de un estado inestable, en el que ninguna ruta es más atractiva que otra, a un estado estable, donde una ruta presenta la atracción más fuerte. Con el paso del tiempo el rastro de la feromona comienza a evaporarse y se reduce su fuerza atractiva. Desde el punto de vista algorítmico, la evaporación de la feromona tiene la ventaja de provocar la convergencia a una solución localmente óptima. Si no hubiera evaporación, todas las trayectorias posibles serían igualmente atractivas para las hormigas.

El proceso por el cual una hormiga es influenciada para ir hacia una fuente de alimento en particular, por otra hormiga o por un rastro químico, es llamado reclutamiento y si este está basado solamente en rastros químicos, se conoce como reclutamiento en masa. Estos fenómenos han inspirado la mayoría de los algoritmos ACO que se han implementado, iniciando primero por el más simple, el Ant System (Dorigo, 1992).

6.8.2 Algoritmo Ant System: Se ha demostrado que la selección de una ruta determinada por parte de una hormiga hacia una fuente de alimento está basada en la auto-organización. Inicialmente, no hay feromona en ninguno de los caminos hacia el alimento y por lo tanto, cada uno de ellos tiene la misma probabilidad de ser elegido por las hormigas. Sin embargo, algunas interferencias y fluctuaciones al azar podrían causar que unas pocas hormigas más, escogieran al azar uno de los caminos, el cual a su vez, estimulará que muchas más hormigas escojan este mismo camino, y así sucesivamente.

Deneubourg, Aron, Goss y Pasteels (1990), desarrollaron un modelo de este fenómeno, basados en el comportamiento que más se acercaba a las observaciones experimentales. Se asume que la cantidad de feromona en un camino es proporcional al número de hormigas que lo han recorrido para llegar a la fuente de alimento. Con esta presunción, la evaporación de las feromonas no fue tomada en cuenta, debido a la corta duración de los experimentos.

En el modelo, la probabilidad de escoger un determinado camino en determinado tiempo depende de la cantidad de hormigas que lo han transitado, hasta ese momento. Más

precisamente, sean A_i y B_i el número de hormigas que han transitado los caminos A y B , después de que i número de hormigas han pasado por allí. La probabilidad $P_A(P_B)$ de que la hormiga $(i + 1)$ escoja el camino $A(B)$ es:

$$P_A = \frac{(k + A_i)^n}{(k + A_i)^n + (k + B_i)^n} = 1 - P_B \quad (1)$$

La Ecuación 1 cuantifica la forma mediante la cual, si una gran cantidad de hormigas han transitado el camino A , existe una probabilidad más alta de que la siguiente escoja el camino A , dependiendo de los valores absolutos y relativos de A_i y B_i . El parámetro n determina el grado de no linealidad de la función de elección: si n es grande, significa que un camino tiene un poco más de feromonas que el otro y así, la próxima hormiga que pasa tiene una alta probabilidad de escogerlo.

El parámetro k cuantifica el grado de atracción de un camino sin marcar: cuanto más grande sea k , mayor es la cantidad de feromona para que la elección no sea al azar.

La forma particular de P_A fue obtenida mediante experimentos con hormigas, basados en su seguimiento del rastro de feromonas. Los valores de los parámetros k y n que permiten que la función se acerque más a los resultados obtenidos mediante experimentación son $n \approx 2$ y $k \approx 20$. Si $A_i \gg B_i$ y $A_i \gg 20$, $P_A \approx 1$; Si $A_i \gg B_i$ pero $A_i < 20$, entonces $P_A \approx 0.5$. Lo mismo aplica para P_B . La dinámica de la elección, según la Ecuación 1, sigue las siguientes reglas:

$$\begin{aligned} A_{i+1} &= \begin{cases} A_i + 1 & \text{si } \delta \leq P_A; \\ A_i & \text{si } \delta > P_A, \end{cases} \\ B_{i+1} &= \begin{cases} B_i + 1 & \text{si } \delta > P_A; \\ B_i & \text{si } \delta \leq P_A, \end{cases} \quad (2) \\ A_i + B_i &= i, \end{aligned}$$

donde δ es una variable aleatoria, uniformemente distribuida en el intervalo $[0,1]$. Los modelos expresados mediante las ecuaciones anteriores fueron analizados mediante simulación por el método de Monte Carlo y los resultados están en perfecta concordancia con los experimentos realizados.

En estos experimentos se encontró que el rastro de feromonas que dejan las hormigas usualmente persiste por un período de tiempo considerable, desde muchas horas hasta muchos meses (dependiendo de la especie, el sustrato, el tamaño de la colonia, las condiciones climáticas, etc.), indicando que la vida útil de las feromonas deben ser medidas en escalas de tiempo considerablemente mayores.

Optimización es definida con respecto a muchas restricciones, incluidas restricciones ecológicas, como la depredación o la competencia con otras colonias. El algoritmo que se describen a menudo en ACO se formulan teniendo en cuenta técnicas similares a la evaporación de feromonas, el cual es un poderoso método para la optimización.

6.8.3 Descripción del algoritmo ACO (Ant Colony Optimization): Los algoritmos de Optimización por Colonia de Hormigas (ACO) realmente corresponde a una familia de algoritmos de optimización, inspirados en la estrategia de feromonas que utilizan las hormigas forrajeras. Estos algoritmos fueron concebidos inicialmente para encontrar la ruta más corta en el problema del viajante (Travel Salesman Problem). En ACO, varias hormigas viajan a través de las aristas que conectan los nodos en un grafo, mientras depositan feromonas virtuales (Floreano & Mattiussi, 2008).

Las hormigas que viajan por el camino más corto están en capacidad de hacer más viajes de regreso y depositar más feromonas, en un tiempo determinado. Por lo tanto, el camino puede “atraer” mas hormigas, convirtiéndose en un ciclo de realimentación positiva. En la naturaleza, sin embargo, si muchas hormigas escogen el camino más largo durante la exploración inicial, este camino será reforzado, así no sea el más corto. Para solucionar este inconveniente, ACO asume que las feromonas virtuales se evaporan, reduciendo de esta manera la probabilidad de que los caminos más largos sean seleccionados. Varios tipos de algoritmos ACO basados en esos principios están siendo desarrollados con variaciones, según el tipo de problema que deban resolver; pero todos se basan en el principio básico que se describe a continuación.

Inicialmente, las hormigas se distribuyen al azar en los nodos del grafo. Cada hormiga artificial debe escoger una arista o camino a seguir, desde donde se encuentre, haciendo uso de una regla de probabilidad que toma en cuenta la longitud de esa arista y la cantidad de feromonas presentes en ella. Las aristas o caminos que llegan a los nodos y que han sido visitadas antes por la misma hormiga, no son considerados en la elección.

Una vez todas las hormigas han completado un tour o recorrido completo por el grafo, cada una de ellas vuelve a trazar o a seguir su propia ruta, mientras deposita en el camino una cantidad de feromonas que es inversamente proporcional a la longitud de la ruta. Antes de que las hormigas vuelvan a iniciar otra búsqueda, desde diversos sitios al azar, las feromonas en todas las aristas del grafo o caminos se evaporan en una pequeña cantidad. Esta evaporación, combinada con la elección probabilística de la ruta, asegura que las hormigas eventualmente transiten por uno de los caminos más cortos, aunque existirán algunas otras que continúen viajando por caminos un poco más largos.

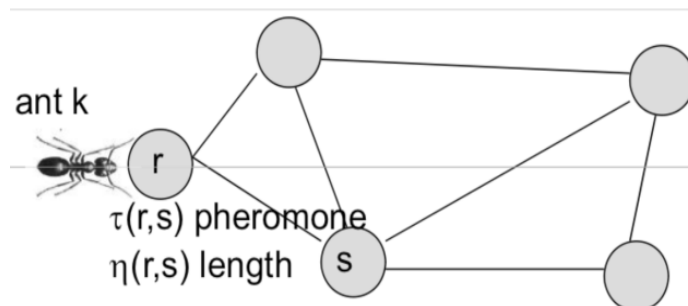


Figura 19. Grafo que describe el camino que una hormiga debe tomar, dependiendo de la longitud y la cantidad de feromona presente. Fuente: Dorigo, M. & Stützle, T. (2004) Ant Colony Optimization. USA. MIT Press.

Consideremos entonces un número de hormigas M , que usualmente es igual al número de nodos N en el grafo. Una pequeña cantidad de feromonas virtuales es depositada en todas la aristas al inicio de la búsqueda.

La probabilidad p_{ij}^k de que la hormiga k escoja la arista que va desde el nodo i hasta el nodo j está dada por:

$$p_{ij}^k = \frac{\tau_{ij}^a \eta_{ij}^b}{\sum_{h \in J^k} \tau_{ih}^a \eta_{ih}^b} \quad (3)$$

donde τ_{ij} es la cantidad de feromonas virtuales en la arista y η_{ij} es la visibilidad del nodo, que corresponde a la inversa de la longitud de la arista o camino $1/l_{ij}$. Las constantes a, b corresponden a 2 factores importantes. Si $a = 0$, las hormigas solamente escogen la distancia más corta y si $b = 0$ solamente basan su elección en la cantidad de feromonas. El divisor de la ecuación corresponde a la sumatoria de los valores de visibilidad y de feromonas para todas las aristas H que están conectadas al nodo donde la hormiga se encuentra y a su vez, conectan a otros nodos que pertenecen al conjunto J^k de nodos que la hormiga k aún no ha visitado. Tan pronto como la hormiga visita un nodo, este es borrado de la lista de nodos J^k .

Una vez todas las hormigas han completado un tour o un recorrido del grafo, cada hormiga k vuelve a trazar su propio camino y deposita una cierta cantidad de feromona $\Delta\tau_{ij}^k$ en las aristas o caminos por las que ha pasado, de acuerdo con la Ecuación 4:

$$\Delta\tau_{ij}^k = \frac{Q}{L^k} \quad (4)$$

donde L^k es la longitud total del camino encontrado por la hormiga k y Q es una constante, la cual se ajusta para que sea la longitud del camino más corto, estimado mediante un simple método heurístico. La cantidad de feromonas en cada arista, después de que M hormigas han vuelto a trazar su propio camino, es igual a la descrita en la Ecuación 5:

$$\Delta T_{ij} = \sum_k^M \Delta\tau_{ij}^k \quad (5)$$

Antes de que las hormigas nuevamente empiecen a buscar el camino más corto hacia una fuente de alimento, los niveles de feromona se evaporan de acuerdo a la Ecuación 6:

$$\tau_{ij}^{t+1} = (1 - \rho)\tau_{ij}^t + \Delta T_{ij} \quad (6)$$

donde $0 \leq \rho < 1$ es el coeficiente de evaporación de la feromona.

Toda esta descripción matemática corresponde a una iteración del algoritmo. El proceso se repite por varias cientos de iteraciones, hasta que se encuentre un camino relativamente corto hasta la fuente de alimento. Este algoritmo no garantiza encontrar siempre el camino más corto, pero para grafos de tamaño moderado (aproximadamente 30 nodos) proporciona igual o mejores soluciones que los mejores algoritmos diseñados anteriormente para resolver, por ejemplo, el problema del viajante.

El desempeño del algoritmo puede ser mejorado, dejando que aquellas hormigas que encuentren el camino más corto en cada iteración, vuelvan a trazar su propio camino muchísimas veces, mientras depositan la feromona virtual.

La mayor ventaja de los algoritmos ACO sobre otros tipos de algoritmos de búsqueda es que las hormigas virtuales descubren y mantienen varios caminos cortos, adicionalmente al mejor, debido a que eligen las aristas o caminos mediante probabilidad. Si una arista no se puede recorrer o un nodo ya no está disponible, las hormigas podrán rápidamente usar y trazar nuevamente un segundo camino, mientras otros algoritmos deben realizar la búsqueda nuevamente sobre este grafo modificado.

Este es el caso del enrutamiento en redes de comunicación en la vida real, donde los datos deben ser enviados entre dos puntos, en el menor tiempo posible y con la mejor calidad, sin importar la congestión o el fallo de algunos nodos.

7 HIPOTESIS DE INVESTIGACION

El uso de la técnica de enjambre (Swarm) presenta mejor desempeño, para ejecución de tareas colectivas en ambientes simulados de sistemas multirobots, frente al uso de la técnica multiagente.

8 OPERACIONALIZACION DE VARIABLES

Algunos de los tipos de variables que se ven involucradas en la medición del desempeño de un algoritmo multirobot y que se utilizaron, son las siguientes:

8.1 VARIABLES INDEPENDIENTES

8.1.1 Variables independientes relacionadas con el desempeño computacional:

- Tiempo: Medido como el número y tipo de pasos de ejecución de un algoritmo.
- Espacio: Referido a la cantidad de memoria utilizada para resolver el problema.

8.1.2 Variables independientes relacionadas con el sistema multi-robot:

- Trayectoria más corta hasta el objetivo final.
- Tiempo empleado por el agente en llegar a su objetivo final.
- Tasa de evaporación de la feromona
- Tasa de difusión de la feromona

8.2 VARIABLES DEPENDIENTES

- Desempeño:

Se realiza su medición con base en tres criterios:

- **Efectividad**: Mide cuán buena es la solución. Corresponde al número de llegadas exitosas del agente a su objetivo. Se establece que las medidas de eficiencia y consistencia se tomarán con una efectividad del 80%, es decir, el tiempo en ticks se medirá cuando el 80% de los agentes robóticos lleguen a su objetivo.
- **Eficiencia**: Mide cuán rápido converge el algoritmo. La función de costo empleada será el tiempo, medido en ticks del programa. Debe tenerse en cuenta que ambas estrategias (enjambre y multiagente) son colaborativas, por lo tanto, los agentes deben interactuar entre sí y llegar al objetivo mancomunadamente, de acuerdo a una efectividad del 80%.
- **Consistencia**: Mide como se comporta la variabilidad del resultado. Se medirá a través de las comparaciones de ambos algoritmos de acuerdo a sus medias, cuando la efectividad sea del 80%.

8.3 PRUEBAS ESTADÍSTICAS

Para determinar si se acepta o rechaza la hipótesis nula existen pruebas estadísticas que se pueden clasificar como:

- *Paramétricas*: Están basadas en suposiciones sobre los datos.
- *No paramétricas*: No requieren suposiciones para su aplicación.

Las pruebas *paramétricas* son más robustas; en el caso de ser empleadas es necesario comprobar previamente las suposiciones mediante pruebas estadísticas las cuales suelen ser de:

- *Normalidad*: Determina si los datos siguen una distribución normal
- *Homocedasticidad*: Consiste en establecer si los grupos de datos a comparar presentan la misma varianza.

En el caso de tener normalidad y homocedasticidad lo recomendado es realizar un *análisis de varianza (ANOVA)*; si se presenta la normalidad de los datos pero no la homocedasticidad, se puede emplear la prueba de *Welch* o de *Kruskal Wallis*, finalmente, si no se presenta normalidad la prueba recomendada es la de *Kruskal Wallis*.

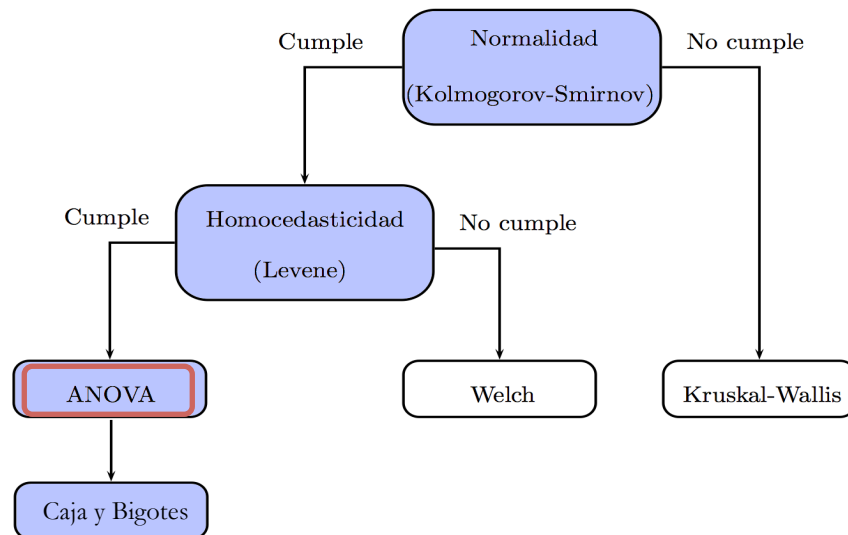


Figura 20. Proceso de análisis estadístico recomendado: Fuente: Adaptado de Espitia, H.E. & Sofrony, J. I. (2012). Algoritmo para Planeación de Trayectorias de Robots Móviles Empleando Campos Potenciales y Enjambres de Partículas Brownianas. Ciencia e Ingeniería Neogranadina, 22(2), 75 – 96.

Si luego de esto se rechaza la hipótesis nula, se concluye que existen diferencias significativas entre los grupos de resultados, por lo cual, el siguiente paso consiste en realizar pruebas de comparaciones múltiples para establecer qué grupos presentan diferencias entre sí.

Si se cumplen los supuestos de normalidad y homocedasticidad, las comparaciones múltiples se pueden realizar empleando los contrastes de: *Duncan*, *Newman-Keuls*, *Bonferroni*, *Scheff'e* o *HSD* de *Tukey*. En el caso de no cumplirse los supuestos de normalidad y homocedasticidad se pueden emplear las pruebas no paramétricas para comparaciones múltiples de: *Nemenyi*, *Holm*, *Bonferroni-Dunn*, o la prueba de Bonferroni como método complementario de la prueba de Kruskal-Wallis.

Como resultado de aplicar el método no paramétrico de Bonferroni se tienen intervalos de confianza los cuales permiten determinar si los grupos a comparar no presentan diferencias significativas. Una representación aproximada del resultado de esta prueba se realiza de forma gráfica donde se muestra el ranking promedio de cada grupo y un intervalo equivalente. Dos grupos de resultados se consideran con diferencias significativas si sus intervalos no se traslapan.

8.3.1 Pruebas estadísticas utilizadas

Con base en los resultados obtenidos, se utilizaron sucesivamente las siguientes pruebas:

1. Normalidad (Kolmogorov – Smirnov)
2. Homocedasticidad (Levene)
3. ANOVA
4. Diagrama de Caja y Bigotes

El análisis ANOVA corresponde a **Tipos de algoritmos vs. Tiempo de ejecución** (ticks). La prueba de la caja y bigotes se uso para determinar el método más eficiente.

En el capítulo de Análisis de Resultados se hace una descripción del procedimiento y en el ANEXO 5. PRUEBAS ESTADÍSTICAS se presentan los datos obtenidos, las tablas y los gráficos utilizados para cada una de las pruebas.

9 METODOLOGÍA

Para el logro de los objetivos se implementaron las siguientes metodologías:

- **MAD-Smart**, para la conceptualización, el análisis y diseño de sistemas multiagentes robóticos.
- **QFD**, para el diseño del hardware y los requerimientos asociados a la plataforma robótica colectiva.
- **DOE** (Diseño de Experimentos), como técnica de recolección y análisis de la información arrojada por las simulaciones, para llevar a cabo la comparación entre los algoritmos utilizados, de manera cuantitativa, utilizando las pruebas estadísticas antes descritas.

El tipo y el método de investigación fue observacional y analítico, pues se realizó la comparación entre algoritmos, lo que permitió establecer como válido el criterio de desempeño, según las variables y las métricas planteadas.

9.1 METODOLOGÍA MAD-SMART:

El enfoque de la metodología para el análisis y diseño de sistemas multiagente robóticos MAD-Smart está fundamentado en:

- Independencia de las técnicas de implementación.
- Proceso metodológico ascendente iniciando en la determinación de los requerimientos del proyecto para llegar a las estrategias de solución de éstos de manera cooperativa.

La metodología recoge elementos de trabajos existentes para la construcción de Sistemas Multi-Agente como son GAIA, Mas-CommonKADS y MaSe, a la vez que introduce otros nuevos que surgen desde la propia área de los Sistemas Multi-Agentes Robóticos (Jiménez, 2007).

Sirve como herramienta de documentación y guía de proyectos, a la vez que permite conservar conocimiento que puede ser reutilizado en desarrollos posteriores.

Los pasos que la constituyen se aprecian en la figura xxx y están orientados a capturar la definición de mecanismos para la comunicación y manejo de conceptos entre los agentes, la descripción de los agentes individuales –incluyendo características hardware y software–, la descripción de los mecanismos de interacción entre los miembros del grupo, y la distribución de tareas entre los agentes.

Se divide en tres partes principales:

Conceptualización: La finalidad de esta etapa es permitir al desarrollador tener una mejor comprensión de cuál es el problema a resolver, el alcance y limitaciones del sistema a desarrollar. A este nivel la definición de los agentes no es importante.

Análisis: Al finalizar esta fase se definirán o se reutilizarán los agentes con sus respectivas características físicas, de comunicación, razonamiento, y las estrategias para la formación de equipos y la solución cooperativa de tareas para el alcance de objetivos.

Diseño: El propósito en esta fase es acercar lo más posible los agentes definidos en la fase anterior a su posterior implementación. En este punto se tiene en consideración para cada agente su parte hardware y software en una misma plantilla. El nivel de abstracción manejado se busca que sea lo más alto posible ya que las posibilidades de implementación en el campo de la robótica son muy amplias tanto a nivel de los sensores y efectores disponibles como a nivel de procesamiento y control (Acosta, 2010).

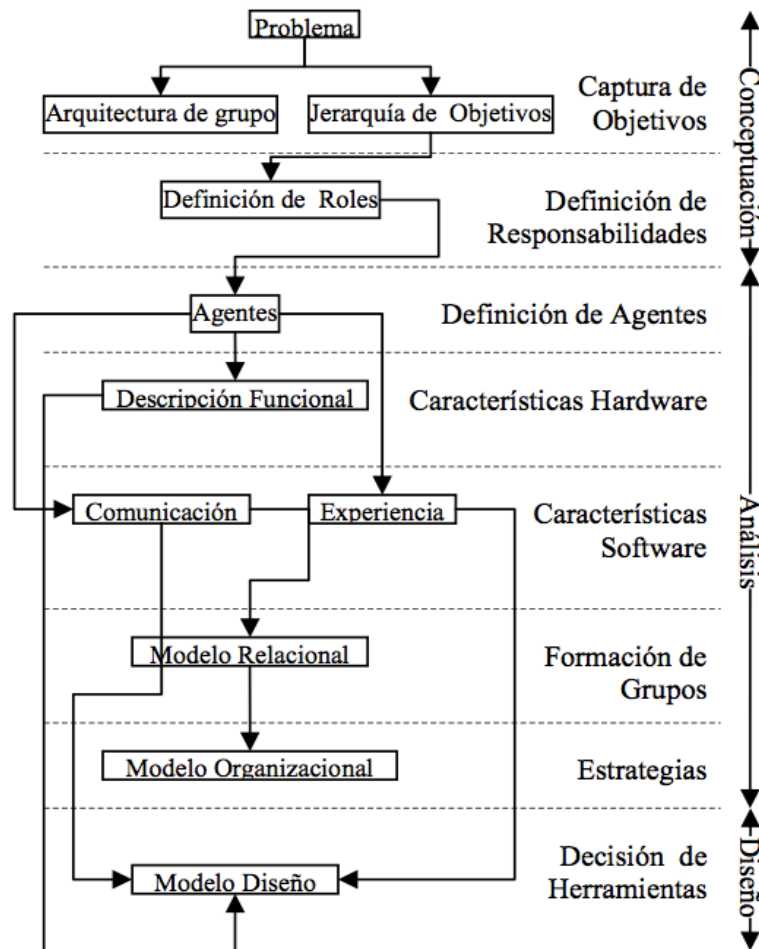


Figura 21. Pasos y actividades de la metodología MAD-Smart. Jiménez, J., Vallejo, M., & Ochoa, J. (2007). Metodología para el Análisis y Diseño de Sistemas Multi-Agente Robóticos: MAD-Smart. Segundo Congreso Colombiano de Computación.

9.1.1 Conceptualización

9.1.1.1 Captura De Objetivos

Estrategia de Enjambre

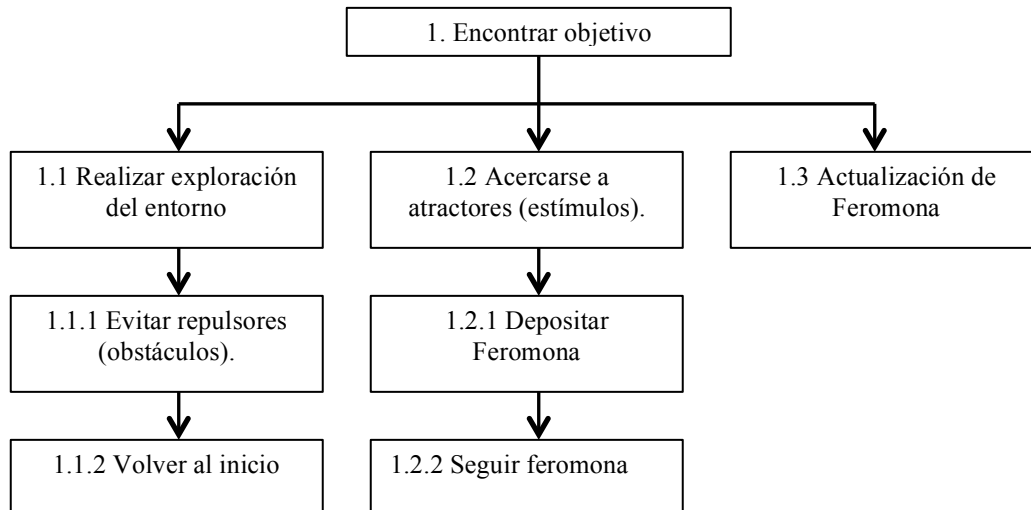


Figura 22. Objetivos de la Estrategia de Enjambre.

Estrategia Multiagente

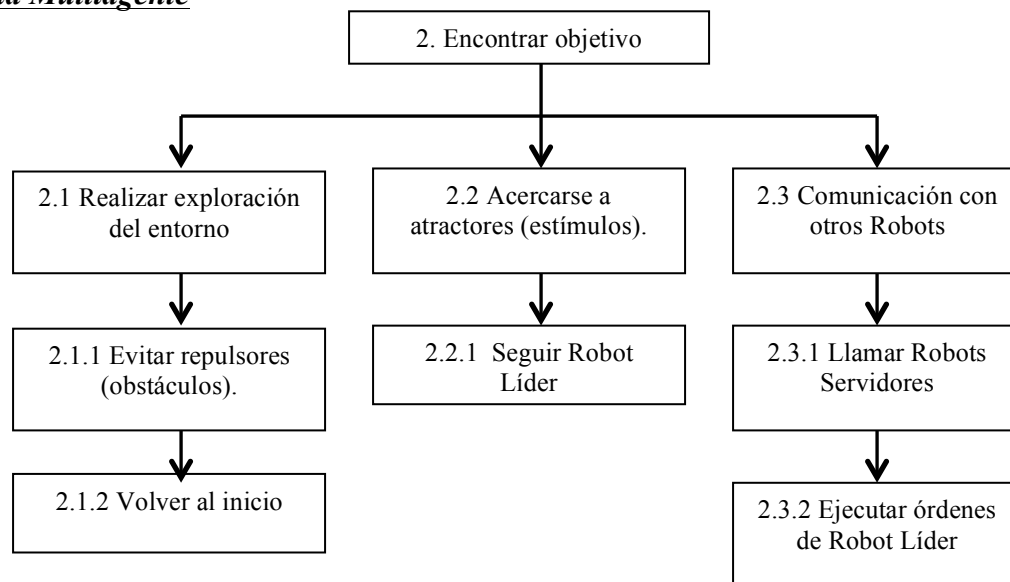


Figura 23. Objetivos de la Estrategia Multiagente.

9.1.1.2 Definición De Responsabilidades

Ítem	Decisión	Descripción
<i>Nombre del Sistema</i>		
<i>Número de Agentes Robóticos</i>	1	Existe solo un tipo de agente, el hormiga.
<i>Número de Agentes de Software</i>	30	Se realizará la simulación con un mínimo de 30 agentes.
<i>Toma Decisiones</i>	Distribuido	Cada agente determina las acciones que llevará a cabo teniendo en cuenta su estado actual y negociaciones con otros agentes y el entorno.
<i>Distribución del Conocimiento</i>	Distribuido	
<i>Característica del grupo</i>	Homogéneo	Los agentes poseen las mismas características
<i>Comunicación</i>	Indirecta	Comunicación no explícita mediante feromona y tasa de difusión.
<i>Topología de comunicación</i>	Broadcast	Los mensajes están dirigidos a todos los agentes, no a alguno en particular.
<i>Redundancia</i>	Si	Ante el fallo de un agente reactivo, el sistema sigue en operación, pues cualquier otro puede desempeñar sus mismos roles.

Figura 24. Definición de responsabilidades para la estrategia de Enjambre.

Ítem	Decisión	Descripción
<i>Nombre del Sistema</i>		
<i>Número de Agentes Robóticos</i>	2	Existen dos tipos de agentes, el autónomo y el servidor.
<i>Número de Agentes de Software</i>	30	Se realizará la simulación con un mínimo de 30 agentes (cuantos autónomos y cuantos servidor?)
<i>Toma Decisiones</i>	Centralizado	El logro del objetivo está definido por las metas dadas por los agentes autónomos.
<i>Distribución del Conocimiento</i>	Centralizado	Quienes conocen el objetivo y su localización son los robots autónomos.
<i>Característica del grupo</i>	Heterogéneo	Los agentes poseen diferentes características, según su rol. Existen 2 tipos de roles: autónomo y servidor
<i>Comunicación</i>	Directa	La comunicación se realiza punto a punto entre el agente autónomo y el que las sigue (servidor).
<i>Topología de comunicación</i>	Multicasting	Los mensajes están dirigidos a ciertos tipos de agentes en particular, según su rol.
<i>Redundancia</i>	No	Los agentes no pueden cambiar su comportamiento, pues este está dado por su rol específico por lo tanto si todos los agentes con un mismo rol llegasen a fallar, el sistema fallaría.

Figura 25. Definición de responsabilidades para la estrategia Multiagente.

Definición de Roles:

Las acciones de los agentes son desatadas por las capacidades de los objetos del entorno; de acuerdo a los conceptos expuestos por Braitenberg (Braitenberg, 1986) y la arquitectura de Subsunción (Brooks, 1987), los micro-robots de este proyecto presentan comportamientos cuyas características dependen de las percepciones de las entidades del entorno, liberan

comportamientos e inhiben otros, las prioridades dependen de las probabilidades de éxito que estos proporcionen a la misión del robot.

Para caracterizar el comportamiento del robot en la estrategia de enjambre, y con miras a que su conducta sea adecuada ante las distintas condiciones que puedan presentarse, se ha establecido que sus acciones se agrupan en conjuntos, dependiendo de los estímulos percibidos en un instante dado; cada grupo o conjunto de acciones se ha denominado “*modo de operación*”; de acuerdo a los algoritmos implementados, correspondientes a las estrategias adoptadas, se han definido 8 modos de operación diferentes, como se aprecia en la Figura 26:

- Modo Reposo
- Modo Búsqueda
- Modo Seguir sonido
- Modo Seguir robot
- Modo Seguir luz
- Modo Seguir feromona
- Modo Regresar al punto inicial
- Modo Evadir obstáculo

		Estímulo percibido						
		Ningún Estímulo	"Auditivo" (escucha el objetivo)	Señal de Robot	"Visible" (ve el objetivo)	"Olfativo" (encuentra feromona)	"Palpable" (toca el objetivo)	Obstáculo
Modo actual	1. Reposo	Detenido	Modo: Seguir "sonido"	Modo: Seguir robot	Modo: Seguir "luz"	Modo: Seguir "feromona"	Detenido	Modo: Evadir obstáculo
	2. Búsqueda	Movimiento aleatorio	Modo: Seguir "sonido"	Modo: Seguir robot	Modo: Seguir "luz"	Modo: Seguir "feromona"	Modo: Regresar	Modo: Evadir obstáculo
	3. Seguir "sonido"	Modo: Búsqueda	Modo: Seguir "sonido"	Modo: Seguir robot	Modo: Seguir "luz"	Modo: Seguir "feromona"	Modo: Regresar	Modo: Evadir obstáculo
	4. Seguir robot	Modo: Búsqueda	Ignorar	Modo: Seguir robot	Modo: Seguir "luz"	Modo: Seguir "feromona"	Modo: Regresar	Modo: Evadir obstáculo
	5. Seguir "luz"	Modo: Búsqueda	Ignorar	Ignorar	Modo: Seguir "luz"	Modo: Seguir "feromona"	Modo: Regresar	Modo: Evadir obstáculo
	6. Seguir "feromona"	Modo: Búsqueda	Ignorar	Ignorar	Ignorar	Modo: Seguir "feromona"	Modo: Regresar	Modo: Evadir obstáculo
	7. Regresar	Modo: Regresar	Ignorar	Ignorar	Ignorar	Ignorar	Modo: Regresar	Modo: Evadir obstáculo
	8. Evadir obstáculo	Modo: Búsqueda	Ignorar	Ignorar	Ignorar	Ignorar	Modo: Regresar	Modo: Evadir obstáculo

Figura 26. Acciones y comportamientos de los robots para la estrategia de enjambre.

En la estrategia de enjambre todos los agentes poseen los mismos dos roles dentro de la misma estrategia elegida, dependiendo del estímulo externo; esto sucede porque se trata de que los individuos tengan las mismas características.

Plantilla del Rol: Explorador Reactivo. Estrategia: Enjambre.
<p><i>Descripción:</i> Los robots reactivos salen de la base explorando en direcciones aleatorias, una vez perciben el sonido del objetivo se dirigen a él, cuando están más cerca de este perciben su luz; cada estímulo captado aumenta la prioridad de encaminamiento hacia el objetivo; cuando un robot hace contacto por primera vez con este, el robot regresa a la base dejando un rastro de feromona que puede ser percibido por los demás.</p> <p><i>Condición de Activación:</i> El rol siempre está activo, sus comportamientos se derivan de la interacción con el entorno y de los estímulos positivos o negativos que en el encuentro.</p> <p><i>Actividades:</i></p> <ul style="list-style-type: none"> • Modo Reposo • Modo Búsqueda • Modo Seguir Sonido • Modo Seguir luz • Modo Regresar al punto inicial • Modo Evadir obstáculo <p><i>Objetivos que satisface:</i> (ver figura 22).</p> <p><i>Esquema de ejecución del Rol:</i> (Reposo, Búsqueda, Seguir Sonido, Seguir Luz, Regresar al punto inicial, Evadir obstáculo).</p>

Figura 27. Plantilla del Rol Explorador Reactivo de la Estrategia Enjambre.

Plantilla del Rol: Explorador Seguidor Estrategia: Enjambre.
<p><i>Descripción:</i> Los robots reactivos salen de la base explorando en direcciones aleatorias, una vez perciben el sonido dirigen a él, cuando están más cerca de este perciben su luz; cada estímulo captado aumenta la prioridad de encaminamiento hacia el objetivo, al hacer contacto con este, y al percibir un rastro de feromona, el robot regresa a la base siguiendo el camino que dejó el robot explorador reactivo y a su vez, dejando un rastro de su propia feromona, que puede ser percibido por los demás.</p> <p><i>Condición de Activación:</i> El rol siempre está activo, sus comportamientos se derivan de la interacción con el entorno y de los estímulos positivos o negativos que en el encuentro.</p> <p><i>Actividades:</i></p> <ul style="list-style-type: none"> • Modo Reposo • Modo Búsqueda • Modo Seguir Sonido • Modo Seguir luz • Modo Seguir feromona • Modo Regresar al punto inicial • Modo Evadir obstáculo <p><i>Objetivos que satisface:</i> (ver cuadro de objetivos).</p> <p><i>Esquema de ejecución del Rol:</i> (Reposo, Búsqueda, Seguir Sonido, Seguir Luz, Seguir Feromona, Regresar al punto inicial, Evadir obstáculo)</p>

Figura 28. Plantilla del Rol Explorador Seguidor de la Estrategia Enjambre.

En la estrategia multiagente, existen tres tipos de roles, el Líder autónomos, el Líder servidores y el Robot servidor.

Plantilla del Rol: Líder de Autónomos.	Estrategia: Multiagente.
<p><i>Descripción:</i> Los robots autónomos salen de la base explorando en direcciones aleatorias; una vez perciben el sonido del objeto, se acercan a él; el que esté más cerca del objetivo percibe su luz, en este momento, se convierte en líder de los autónomos y envía una señal a los demás robots autónomos para que se dirijan hacia él.</p>	
<p><i>Condición de Activación:</i> El rol se activa en el momento en que perciba el sonido del objeto y su luz.</p>	
<p><i>Actividades:</i></p> <ul style="list-style-type: none"> • Modo Reposo • Modo Búsqueda • Modo Seguir Sonido • Modo Seguir Luz • Modo Llamar Autónomos • Modo Evadir obstáculo 	
<p><i>Objetivos que satisface:</i> (ver cuadro de objetivos).</p>	
<p><i>Esquema de ejecución del Rol:</i> (Reposo, Búsqueda, Seguir Sonido, Seguir Luz, Llamada a Autónomos, Evadir obstáculo).</p>	

Figura 29. Plantilla del Rol Líder de Autónomos de la Estrategia Multiagente.

Plantilla del Rol: Líder de Servidores.	Estrategia: Multiagente.
<p><i>Descripción:</i> Los robots autónomos salen de la base explorando en direcciones aleatorias; una vez perciben el sonido del objeto, se acercan a él, pero como ya existe un Líder autónomo, este acude a su llamado y se convierte en Líder Servidor, quien se comunica con los Robot Servidores para que lo siga y así asignarles sus tareas.</p>	
<p><i>Condición de Activación:</i> El rol se activa cuando se percibe el sonido y se reciba la llamada del Líder Autónomo.</p>	
<p><i>Actividades:</i></p> <ul style="list-style-type: none"> • Modo Reposo • Modo Búsqueda • Modo Seguir Sonido • Modo Seguir Robot (Líder Autónomos) • Modo Llamada a Servidores • Modo Seguir Luz • Modo Evadir obstáculo 	
<p><i>Objetivos que satisface:</i> (ver cuadro de objetivos).</p>	
<p><i>Esquema de ejecución del Rol:</i> (Reposo, Búsqueda, Seguir Sonido, Seguir Robot, Llamada a Servidores, Seguir Luz, Evadir obstáculo).</p>	

Figura 30. Plantilla del Rol Líder de Servidores de la Estrategia Multiagente.

Plantilla del Rol: Servidor.	Estrategia: Multiagente.
<i>Descripción:</i> Los robots servidores deben estar siempre en modo escucha; cuando un el Robot Lider Servidor le envía una comunicación, este debe seguirlo y cumplir sus órdenes.	
<i>Condición de Activación:</i> El rol siempre está activo y debe esperar la información proveniente del Líder Servidor para realizar su labor.	
<i>Actividades:</i>	
<ul style="list-style-type: none"> • Modo Reposo • Modo Seguir Sonido • Modo Seguir Robot (Líder Servidores) • Modo Seguir Luz • Modo Evadir obstáculo 	
<i>Objetivos que satisface:</i> (ver cuadro de objetivos).	
<i>Esquema de ejecución del Rol:</i> (Reposo, Seguir Sonido, Seguir Robot, Seguir Luz, Evadir obstáculo).	

Figura 31. Plantilla del Rol Servidor de la Estrategia Multiagente.

9.1.2 Análisis

9.1.2.1 Definición De Agentes

Los agentes (robots) usados en este trabajo dependen de las estrategias utilizadas.

Agente: Reactivo
<i>Capacidades de Sensado/Acción:</i> Movimiento a través del entorno, determinación de presencia/ausencia de objetos, comunicación indirecta con otros agentes mediante deposición de feromona, detección de feromona de otros agentes.
<i>Capacidades Razonamiento/Experiencia:</i> Procesamiento de información procedente del entorno (estímulo luz y feromona) y comportamiento eusocial que conduce al objetivo, al surgir un comportamiento emergente.
<i>Roles:</i> Explorador Reactivo y Explorador Seguidor
<i>Grupo al que pertenece:</i> Enjambre

Figura 32. Detalles del Agente Reactivo.

Agente: Autónomo
<i>Capacidades de Sensado/Acción:</i> Movimiento a través del entorno, determinación de presencia/ausencia de objetos, comunicación directa con otros agentes de tipo multicasting.
<i>Capacidades Razonamiento/Experiencia:</i> Capacidad de desplazamiento, recolección y transmisión de información. Generan metas que asignan a los agentes autónomos y servidores.
<i>Roles:</i> Líder Autónomo y Líder Servidor.
<i>Grupo al que pertenece:</i> Multiagente

Figura 33. Detalles del Agente Autónomo.

Agente: Servidor
<i>Capacidades de Sensado/Acción:</i> Movimiento a través del entorno, determinación de presencia/ausencia de objetos.
<i>Capacidades Razonamiento/Experiencia:</i> Capacidad de desplazamiento, comportamiento colaborativo para llegar al objetivo. Siguen las metas que asignan los agentes autónomos.
<i>Roles:</i> Servidor
<i>Grupo al que pertenece:</i> Multiagente

Figura 34. Detalles del Agente Servidor.

9.1.2.2 Características de Software

9.1.2.2.1 Modelo de la Experiencia

Corresponde a las herramientas de razonamiento de las cuales harán uso los robots para la consecución del objetivo.

Sistema de Navegación:

Para la simulación de las estrategias de enjambre y multiagente se plantea una situación, en donde el objetivo principal consiste en que el robot llegue a la meta, la cual se encuentra en unas coordenadas preestablecidas, pero desconocidas para el robot. Cada uno de los comportamientos necesarios para llegar a la meta, está influenciado por diferentes estímulos provenientes del mundo exterior.

El método de navegación utilizado en ambos algoritmos es el de *Campos Potenciales*; se supone a todas las entidades del entorno como partículas, los obstáculos ejercen una fuerza de repulsión sobre el móvil, mientras que su punto de arribo ejerce una fuerza atractiva; se consigue entonces establecer un campo potencial que representa el ambiente en el que el robot debe seguir su trayectoria. Los resultados de esta estrategia de navegación están condicionados a las siguientes variables:

1. *Sensibilidad al obstáculo*: Especifica la influencia de los repulsores (obstáculos) en la navegación del robot, determinando la sensibilidad de este ante la presencia del obstáculo.
2. *Sensibilidad al atractor*: Especifica la influencia del atractor (objetivo) en la navegación del robot, determinando la sensibilidad de este para detectar y dirigir sus movimientos hacia el objetivo, en concreto este parámetro determina la sensibilidad del robot ante un estímulo positivo provisto por el objetivo.

La atracción y repulsión pueden definirse como capacidades de los objetos del entorno, las cuales se analizarán a continuación.

Mapeo de Entornos:

Para la simulación de las estrategias de enjambre y multiagente se plantea una situación, en donde el objetivo principal consiste en que el robot llegue a la meta, la cual se encuentra en unas coordenadas preestablecidas, pero desconocidas para el robot, que se encuentra presente en un entorno determinado.

Este entorno puede definirse como un conjunto de entidades de diferentes clases, que según el tipo de estrategia, tienen una interacción entre sí, afectando la percepción de los agentes robots.

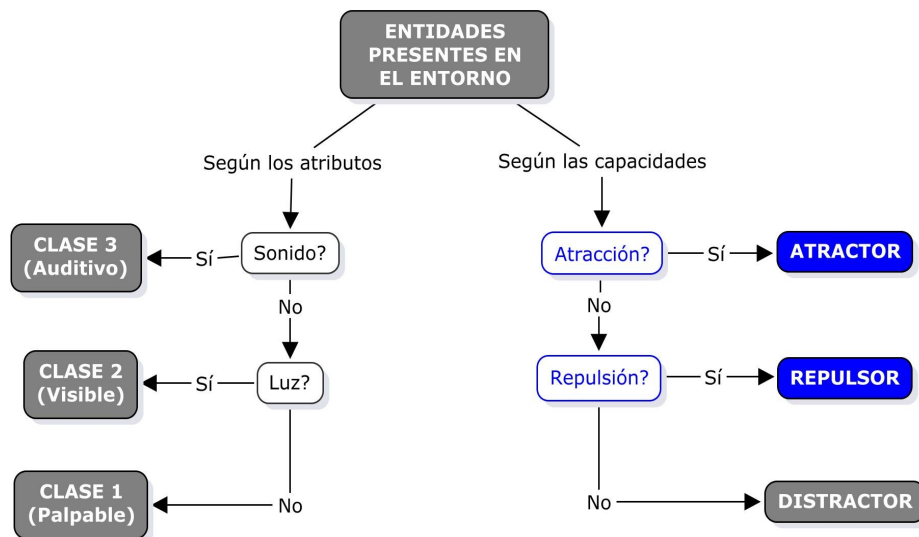


Figura 35. Tipos de entidades presentes en el entorno.

Cada uno de los comportamientos necesarios para llegar a la meta, está influenciado por diferentes estímulos provenientes del mundo exterior. Estas percepciones que poseen los agentes acerca de dichos estímulos, que se dividen en positivos y negativos, generan dos tipos de acciones: de atracción y repulsión.

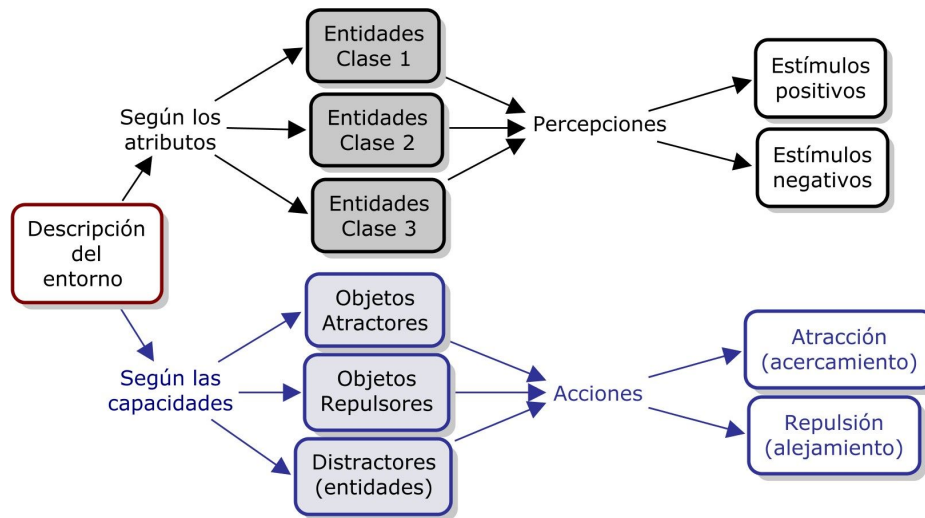


Figura 36. Percepciones y acciones esperadas en el mapeo de entornos.

Como resultado de las entidades presentes, se caracteriza el entorno específico para el modelado de las estrategias de enjambre y multiagente, junto con los conceptos con los cuales interactúa, como se observa en la Figura 37.

Concepto: Entorno
Descripción: Medio en el que se desenvuelve el sistema. Corresponde a un conjunto de entidades con diferentes características.
Atributos: <ul style="list-style-type: none"> ○ <i>Tamaño:</i> Matriz de 58X128 de reales. Cada entrada es el grado de probabilidad de que en esa posición se encuentre un objeto. ○ <i>Objetos Clase 1:</i> Sus atributos proveen únicamente un estímulo palpable, solo pueden ser percibidos por contacto directo del robot. ○ <i>Objetos Clase 2:</i> Sus atributos proveen un estímulo palpable y un estímulo visible, pueden ser percibidos a distancia solo mediante línea visual directa.
Capacidades: <ul style="list-style-type: none"> ○ <i>Atractores:</i> Tienen la capacidad de atraer al robot provocando su acercamiento, estas entidades son objetos del entorno. ○ <i>Repulsores:</i> Tienen la capacidad de rechazar al robot provocando su alejamiento, estas entidades son objetos del entorno. ○ <i>Distractores:</i> No poseen capacidades de atraer o rechazar al robot, estas entidades no son objetos del entorno.
Estímulos positivos: Son provistos por los atractores (objetivo, feromona u otro robot en estado “veo objetivo”), se clasifican de acuerdo a la percepción que obtiene el robot de acuerdo a la distancia de estos. <ul style="list-style-type: none"> ○ <i>Estímulo “Auditivo”:</i> es el estímulo percibido a mayor distancia, indica la presencia del atractor en el universo, su radio de alcance puede ser limitado (no necesariamente es perceptible en todo el universo) y no es opacado por los obstáculos. ○ <i>Estímulo “Visible”:</i> es un estímulo percibido a distancia pero es opacado por los obstáculos (requiere línea visual, de ahí su nombre). Es producido por el objetivo, simultáneamente con el estímulo “auditivo”. ○ <i>Señal de Robot:</i> es otro estímulo “visible”, es una señal que un robot emite mientras percibe un estímulo visual (es decir que está en modo Seguir “Luz”), de esta manera prolonga el alcance de percepción del objetivo. ○ <i>Estímulo “Olfativo”:</i> es el estímulo provisto por la feromona, no es perceptible a distancia. ○ <i>Estímulo “Palpable”:</i> es cuando se tiene contacto físico con el objetivo mismo.

<p><i>Estímulos negativos:</i> Son provistos por los repulsores, se clasifican de acuerdo a la percepción que obtiene el robot de acuerdo a la distancia de estos.</p> <ul style="list-style-type: none"> ○ Obstáculo percibido a distancia, es la versión negativa del estímulo visual. ○ Obstáculo percibido al contacto, es la versión negativa del estímulo palpable.
<p>Concepto: Posición</p>
<p><i>Descripción:</i> Ubicación de cualquiera de los robots en el entorno</p>
<p><i>Atributo:</i> Pareja coordenadas (x, y)</p>
<p>Concepto: Feromona</p>
<p><i>Descripción:</i> Rastro que deja una hormiga a su paso, para indicar un camino.</p>
<p><i>Atributo:</i> Color blanco con bordes amarillos.</p>
<p>Concepto: Tasa de evaporación de la feromona</p>
<p><i>Descripción:</i> Parámetro necesario para la actualización de mínimos locales, esto es, para poder identificar la ruta óptima entre todas las posibles rutas que hay entre la base y el objetivo. El algoritmo la actualiza cada que hace una iteración (tick).</p>
<p><i>Atributos:</i> Porcentaje</p>
<p>Concepto: Tasa de difusión de la feromona</p>
<p><i>Descripción:</i> Parámetro que influye en la evaporación del rastro de feromona, pues indica la concentración de feromona en la trayectoria y que tanto se difunde hacia la vecindad de esta. El algoritmo la actualiza cada que hace una iteración (tick).</p>
<p><i>Atributos:</i> Porcentaje</p>
<p>Concepto: Obstáculos (Repulsores)</p>
<p><i>Descripción:</i> Objetos que se encuentran al azar en el espacio y que no permite el avance de los robots.</p>
<p><i>Atributos:</i> Forma: Cilíndrica. Color café. Cantidad: 10. Distribución: al azar con cada configuración. Tamaño: mas grande en comparación con los agentes robóticos.</p>
<p>Concepto: Origen</p>
<p><i>Descripción:</i> Lugar en el que reposan los agentes desde el cual salen a explorar el entorno.</p>
<p><i>Atributos:</i> Forma: Circular. Color Rojo. Distribución: fija. Tamaño: grande respecto a los agentes robóticos y de igual tamaño que los obstáculos.</p>
<p>Concepto: Alcanzar objetivo</p>
<p><i>Descripción:</i> En la estrategia de enjambre, el comportamiento que un robot reactivo explorador y/o reactivo seguidor tiene al alcanzar su objetivo. Apenas lo percibe su luz, se acerca a el y se torna de color rojo, para luego regresar al origen.</p>
<p><i>Atributos:</i> Color rojo</p>

Concepto: Regreso al nido (origen)
<i>Descripcion:</i> En la estrategia de enjambre, es el comportamiento que un robot reactivo explorador y/o reactivo seguidor tiene al devolverse a su nido (origen), luego de haber estado en contacto con el objetivo; apenas regresa al origen, se torna de color verde.
<i>Atributos:</i> Color verde.
Concepto: Rol de Agentes Reactivos Exploradores
<i>Descripcion:</i> Agente robótico que explora el entorno en búsqueda del objetivo en la estrategia de enjambre; es el primero en llegar a la meta y en depositar la feromona.
<i>Atributos:</i> Color azul. Forma: circular. Tamaño: pequeño en comparación con obstáculos. Comportamientos: Llegada al objetivo y regreso al nido.
Concepto: Rol de Agentes Reactivos Seguidores
<i>Descripcion:</i> Agente robótico que explora el entorno en búsqueda del objetivo y sigue el rastro de feromona depositado por el Agente Reactivo Explorador, en la estrategia de enjambre.
<i>Atributos:</i> Color amarillo. Forma: circular. Tamaño: pequeño en comparación con obstáculos. Comportamientos: Llegada al objetivo y regreso al nido.
Concepto: Llamada a Agentes Autónomos
<i>Descripcion:</i> En la estrategia multiagente, comportamiento que exhibe un agente líder autónomo, cuando alcanza su objetivo y llama a los demás de su clase para que lo sigan.
<i>Atributos:</i> Medida: valor 1 en una variable de la programación.
Concepto: Llamada a Agentes Servidores
<i>Descripcion:</i> En la estrategia multiagente, comportamiento que exhibe un agente líder servidor, cuando alcanza conoce donde esta el objetivo y llama a los agentes servidores para que lo sigan y desarrollen su estrategia.
<i>Atributos:</i> Medida: valor 1 en una variable de la programación.
Concepto: Percepción de sonido
<i>Descripcion:</i> En la estrategia multiagente, comportamiento que exhibe un agente autónomo cuando percibe el “sonido” del objetivo.
<i>Atributos:</i> Color: Circulos Blancos
Concepto: Rol de Líder Autónomos
<i>Descripcion:</i> Rol que consiste en coordinar a los agentes autónomos para que lo sigan hasta el objetivo.
<i>Atributos:</i> Color Verde
Concepto: Rol de Líder Servidor
<i>Descripcion:</i> Rol que consiste en seguir al rol de Líder autónomo hasta el objetivo y coordinar a los servidores para que lo sigan y alcancen el objetivo.
<i>Atributos:</i> Color Azul

Concepto: Rol de Agentes Servidores
<i>Descripcion:</i> Rol que consiste en seguir al rol de Líder servidor hasta el objetivo y cumplir sus órdenes en la búsqueda del objetivo.
<i>Atributos:</i> Color azul

Figura 37. Especificación de una ontología para un sistema de navegación colaborativa empleando estrategias de enjambre y multiagente. Las señaladas con *color azul* corresponden a la estrategia de enjambre y las de *color rojo* a la estrategia multiagente. Las que no se encuentran coloreadas son comunes a ambas técnicas.

9.1.2.2.2 Modelo de Comunicación

Modelo de conocidos:

La interacción entre los diferentes tipos de agentes se evidencia en la Figura 38.

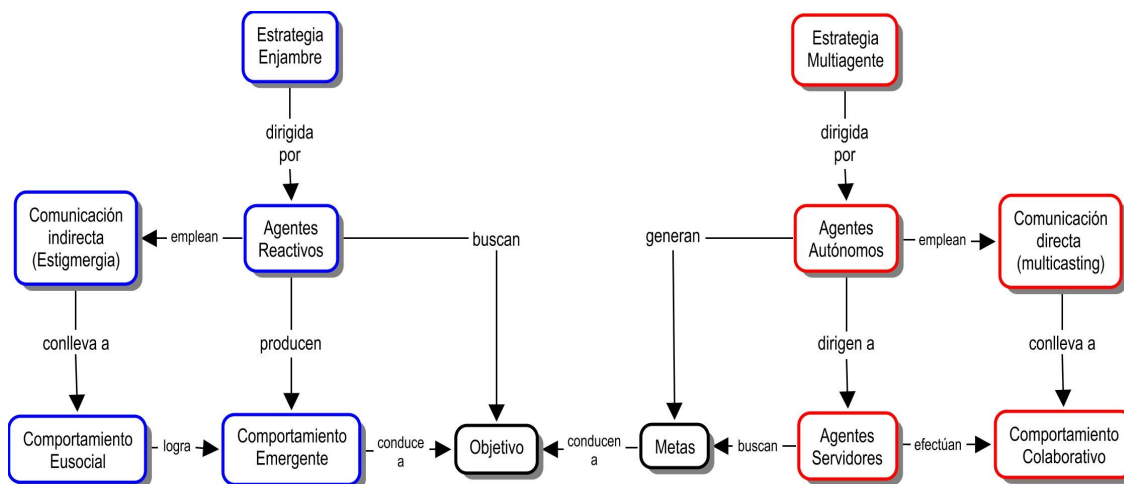


Figura 38. Comunicación entre agentes según la estrategia implementada.

En este modelo se especifican gráficamente cuales agentes interactúan. Los sentidos de las flechas indican los iniciadores de las interacciones.

Modelo de conversaciones

Consiste en describir las secuencias más relevantes del modelo de conocidos, en este caso, se describe la secuencia de Comunicación Indirecta o Estigmergia, correspondiente a la estrategia de Enjambre (ver Figura 39).

Conversación: Estigmergia
Descripción: Secuencia de mensajes entre los agentes del sistema, para poder llegar al objetivo. Los mensajes se envían a través de una feromona.
Conceptos: <ul style="list-style-type: none"> • <i>Evaporación de la feromona:</i> Parámetro necesario para la actualización de mínimos locales, esto es, para poder identificar la ruta óptima entre todas las posibles rutas que hay entre la base y el objetivo. • <i>Tasa de difusión de la feromona:</i> Parámetro que influye en la evaporación del rastro de feromona, pues indica la concentración de feromona en la trayectoria y que tanto se difunde hacia la vecindad de esta.
Secuencia: <ul style="list-style-type: none"> • Un primer agente con rol Explorador Reactivo percibe el sonido de su objetivo. • A medida que se acerca al objetivo, percibe su luz. • Al llegar al objetivo, el agente se devuelve a su punto de origen, depositando una feromona en el camino elegido. • Si la tasa de evaporación de feromona es muy alta, significa que un segundo agente con rol Explorador Reactivo no podrá detectarla y por lo tanto no puede pasar al rol de Explorador Seguidor. • Si la tasa de difusión es muy baja, significa que un segundo agente con rol Explorador Reactivo no podrá detectarla y por lo tanto no puede pasar al rol de Explorador Seguidor. • Si el agente con rol Explorador Reactivo detecta la feromona, pasa a rol Explorador Seguidor, siguiendo el rastro dejado por el primer agente y depositando su propia feromona, reforzando el camino recorrido.

Figura 39. Modelo de Conversaciones: Secuencias más relevantes del modelo de conocidos.

9.1.2.2.3 Proceso de elección del software

La simulación social puede considerarse una metodología de investigación, similar en cuanto a su lógica a otras modalidades de método científico en general, ya que consiste en:

1. *Inferencia abductiva:* Formalizar teorías complejas sobre determinados procesos sociales.
2. *Inferencia deductiva:* Llevar a cabo experimentos a partir de la formalización construida.
3. *Inferencia inductiva:* Observar generación de resultados emergentes de la formalización.

Para la simulación de los algoritmos, se realizó una búsqueda exhaustiva de todo tipo de software que se utiliza para modelar algoritmos de enjambre. En total, se encontraron 41 plataformas de simulación. En la Figura 40 se muestran los diferentes tipos de software y algunas de sus características asociadas. Al momento de la elección se tuvieron en cuenta las siguientes características deseadas e identificadas:

- Tipo de licencia
- Plataformas y/o sistemas operativos en los que funciona.
- Lenguaje de programación.
- Entorno del software.
- Exportación de datos en forma de tablas.
- Creación de interfaz gráfica.
- Creación de un archivo ejecutable con la simulación.
- Artículos en revistas indexadas que lo utilizaran.

Finalmente, se hizo la elección del Software NETLOGO.

En el ANEXO 1 Se describen las características generales del software elegido, como realizar su ejecución, cual es su funcionamiento y como se usa

SOFTWARE	TIPO	APLICACIONES	S.O.	LICENCIA
Algodoo				Freeware
ARS				
AUV Workbench				Freeware
Breve				Open Source
BugWorks				
Carmen				
Darwin2k				
Enki				Open Source
EyeSim				Comercial
EZPhysics				Open Source
Gazebo		multi-robot	Linux	Open Source
ISfMR				
LPZrobots				Open Source
Maple				
Marilou				Comercial
Microsoft Robotics Developer Studio				Comercial
RobotSim				Comercial
Mobs				
Moby				Open Source
MORSE	Simulador			Open Source
MRPT				
NetLogo	Entorno de modelado	multi-agent		
OpenHRP3				Open Source
OpenRave				
OpenSim				Open Source
Peekabot				
Player Project				Freeware
Robocode				
RobotFlow				
RobotSim				Comercial
RoboWorks				Freeware
Rossum's playhouse				
Simbad				
SimplyCube				Comercial
SimRobot				Open Source
Simulink				
Stage				Open Source
TeamBots				
Vensim				
V-rep				Open Source
Webots				Comercial

Figura 40. Lista de Software para simulación de sistemas robóticos..

9.1.2.3 Formación De Grupos

9.1.2.3.1 Modelo Relacional

El modelo jerárquico de objetivos permite determinar donde convergen los agentes para la solución de objetivos que van más lejos de su alcance individual y solo se cumplirán mediante la interacción de los agentes implicados (Acosta, 2010).

Relaciones de agentes con estímulos y el entorno:

Para la simulación de las estrategias de enjambre y multiagente se plantea una situación, en donde el objetivo principal consiste en que el robot llegue a la meta, la cual se encuentra en unas coordenadas preestablecidas, pero desconocidas para el robot. Cada uno de los comportamientos necesarios para llegar a la meta, está influenciado por diferentes estímulos provenientes del mundo exterior.

Estrategia: Enjambre, Multiagente
Agentes que intervienen: Reactivo, Autónomo, Servidor
Secuencia de la conducta ante los estímulos positivos: Está determinada por el tipo de percepciones que se obtengan de los atractores (atractor = objetivo buscado): <ul style="list-style-type: none">○ Inicialmente los robots están ubicados en el punto inicial en Modo de Reposo. Esto es posible si la presencia de los objetivos en el universo no es constante (aparecen en diferentes puntos con el transcurso del tiempo, o si son móviles); este caso es más apropiado para robots físicos en un universo real. Para ambientes simulados es más adecuado que el estado inicial de los robots (es decir, en ausencia de estímulos, tanto positivos como negativos) sea de exploración aleatoria.○ Si un robot percibe un estímulo de tipo “auditivo”, que indica la presencia del objetivo en un punto aún no visible, pasa al modo “Seguir sonido” para acercarse al objetivo.○ Cuando un robot alcanza percepción “visual” del objetivo, pasa al modo “Seguir luz”, en este modo emite una señal visible para indicar a los demás robots que ha “visto” al objetivo. Si un robot percibe la señal visible de otro, pasa al modo “Seguir robot”, con el objetivo de acercarse a este, pues esto aumenta la probabilidad de llegar al objetivo.○ Al encontrar el objetivo (contacto directo con este) el robot pasa al modo “Regresar”, en el cual se dirige al punto inicial de partida, mientras deposita feromona con la que indica que el objetivo fue hallado, y con la cual marca el camino para llegar al mismo.○ Si un robot percibe el estímulo “olfativo” (feromona), pasa al modo “seguir feromona”, ya que esta indica el efectivo hallazgo del objetivo y el camino hacia este. Cuando el robot llega al objetivo regresa al punto inicial, reforzando el rastro de feromona.

Figura 41a

Estrategia: Enjambre, Multiagente
Agentes que intervienen: Reactivo, Autónomo, Servidor
<p>Secuencia de la conducta ante los estímulos negativos:</p> <p>Es determinada por las percepciones que se obtengan de repulsores (repulsor = obstáculo):</p> <ul style="list-style-type: none"> ○ Si el obstáculo es percibido a cierta distancia (en principio todos los obstáculos deberían percibirse de esta manera, para evitar colisiones), el robot disminuye la velocidad de desplazamiento de la rueda opuesta al lado de percepción del estímulo mientras la percepción exista (y eventualmente puede detener por completo dicha rueda). ○ Si se alcanza contacto directo con el obstáculo (colisión), el robot invierte el giro de la rueda opuesta al lado de percepción del estímulo mientras la percepción exista.

Figura 41b.

Figura 41. Relaciones de los Agentes ante los estímulos. a) Estímulos positivos. b) Estímulos negativos.

Relaciones entre agentes:

Estrategia: Enjambre
Agentes que intervienen: Reactivo
<p>Secuencia de la conducta de la estrategia de enjambre</p> <ul style="list-style-type: none"> • Los robots salen de la base explorando el entorno con movimientos aleatorios. • Si perciben el “sonido” del objetivo los robots se dirigen hacia este como fuente de sonido. • Cuando los robots están más cerca del objetivo perciben su luz, entonces se dirigen hacia este como fuente de luz, aumentando la prioridad de encaminamiento hacia el objetivo. • Una vez que los robots hacen contacto con el objetivo regresan a la base dejando un rastro de feromona. • Si un robot percibe feromona ignora los demás estímulos y sigue el camino trazado por la feromona. • Al chocar con un obstáculo los robots lo evaden.

Figura 42a.

Estrategia: Multiagente
Agentes que intervienen: Autónomo, Servidor
<p>Secuencia de la conducta de la estrategia multiagente</p> <ul style="list-style-type: none"> • Los robots autónomos salen de la base explorando el entorno con movimientos aleatorios. • Los robots servidores se encuentran en modo de reposo, pero escuchando a un posible estímulo “sonido” proveniente de un robot autónomo. • Cuando un robot autónomo percibe un estímulo auditivo, se dirige hacia la fuente de este, y se convierte en líder de un grupo de robots servidores. • Cuando los robots autónomos están más cerca del objetivo perciben su luz, entonces se dirigen hacia este como fuente de luz, aumentando la prioridad de encaminamiento hacia el objetivo. • Cuando uno de ellos lo detecta primero, se convierte en el líder de los robots autónomos. • El agente autónomo líder llama a los demás agentes autónomos para que colaboren con la consecución del objetivo. • Los agentes autónomos líder de servidores llama a los agentes servidores y les imparten instrucciones para conseguir el objetivo. • Al chocar con un obstáculo los robots lo evaden.

Figura 42b.

Figura 42. Relaciones de los Agentes con el entorno. a. Estrategia de Enjambre. b. Estrategia Multiagente.

9.1.2.4 ESTRATEGIAS

Se definen las estrategias y secuencias de cooperación para el alcance de los objetivos. Estas son acciones que los agentes deben ejecutar para alcanzar objetivos a largo plazo. Cada estrategia se define como un modelo organizacional bien definido.

9.1.2.4.1 Estrategia de Enjambre

Un sistema de enjambre es un sistema de agentes que presenta las siguientes características:

- Compuesto por muchos(*) agentes
- Los agentes son todos iguales
- Los agentes son reactivos
- Se tiene una arquitectura preestablecida
- El logro del objetivo está definido por la arquitectura

(*)No hay un consenso sobre la cuantificación del calificativo “muchos”, usado en este contexto.

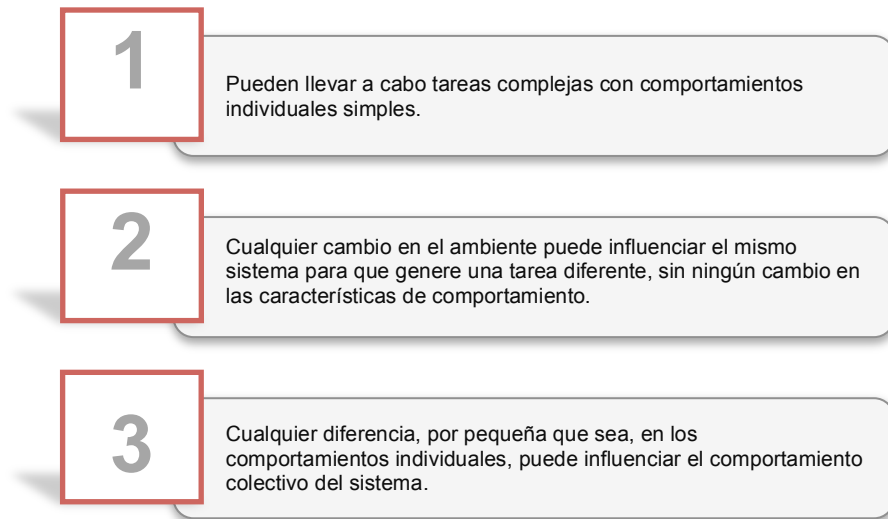


Figura 43. Características deseables en robótica de enjambre, de un sistema auto-organizado. Fuente: Bonabeau, E., & Théraulaz, G. (2000). Swarm smarts. Scientific American, 282(3), 54-61.



Figura 44. Características de un sistema de Enjambre.

Se elige en particular la técnica de enjambre ACO (Dorigo, 1992), basado en los comportamientos e interacciones presentadas por las hormigas, el cual recientemente ha sido implementado en la robótica de enjambre, y se simula mediante un algoritmo que cumple con las características deseadas, descritas anteriormente y además, ACO posee otras relacionadas directamente con su naturaleza, como la escalabilidad, flexibilidad, robustez, simplicidad individual, ejecución concurrente de tareas, control descentralizado y comportamiento colectivo emergente.

Estrategia: Enjambre
Software de implementación: NETLOGO
Entorno: MATRIZ DE 128X58
Agentes: <ul style="list-style-type: none"> • <i>Tipo:</i> Reactivo • <i>Número:</i> 100
Secuencia: <ul style="list-style-type: none"> • Todos los robots salen de la base explorando en direcciones aleatorias. • Una vez perciben el sonido del objetivo se dirigen a él. • Cuando están más cerca del sonido, perciben su luz; cada estímulo captado aumenta la prioridad de encaminamiento hacia el objetivo. • Los robots hacen contacto con el objetivo. • Los robots regresan a la base dejando un rastro de feromona que puede ser percibido por los demás.
Pseudocódigo: <p><i>Inicio del ciclo</i> <i>Evadir los límites del entorno</i> <i>el agente está en modo Buscar Atractor?</i> <i>sí-> Actualizar datos de los sensores del robot</i> <i>asignar conexiones nerviosas tipo Braitenberg a los motores:</i> <i>"agresivo" ante atractor</i> <i>"cobarde" ante repulsor</i> <i>determinar comportamiento Braitenberg según las conexiones nerviosas asignadas</i> <i>la concentración de feromona supera el umbral de atención?</i> <i>sí-> Activar modo Seguir Feromona</i> <i>avanzar según la configuración de Braitenberg</i></p>

el robot encontró el objetivo?
sí-> Activar modo Regresar a Base
el agente está en modo Regresar a Base?
sí-> Actualizar datos de los sensores del robot
Dirigirse a las coordenadas de la base
asignar conexiones nerviosas tipo Braitenberg a los motores:
"cobarde" ante repulsor
determinar comportamiento Braitenberg según las conexiones nerviosas asignadas
avanzar según la configuración de Braitenberg
depositar feromona
el agente está en modo Seguir Feromona?
sí-> Actualizar datos de los sensores del robot
asignar conexiones nerviosas tipo Braitenberg a los motores:
"amoroso" ante atractor
"cobarde" ante repulsor
determinar comportamiento Braitenberg según las conexiones nerviosas asignadas
la concentración de feromona supera el umbral de atención?
sí-> depositar feromona
no-> Activar modo Buscar Atractor
avanzar según la configuración de Braitenberg
Actualizar rastro de feromona según la tasa de difusión y la tasa de evaporación
regresa al inicio del ciclo.

Figura 45. Algoritmo de Enjambre implementado.

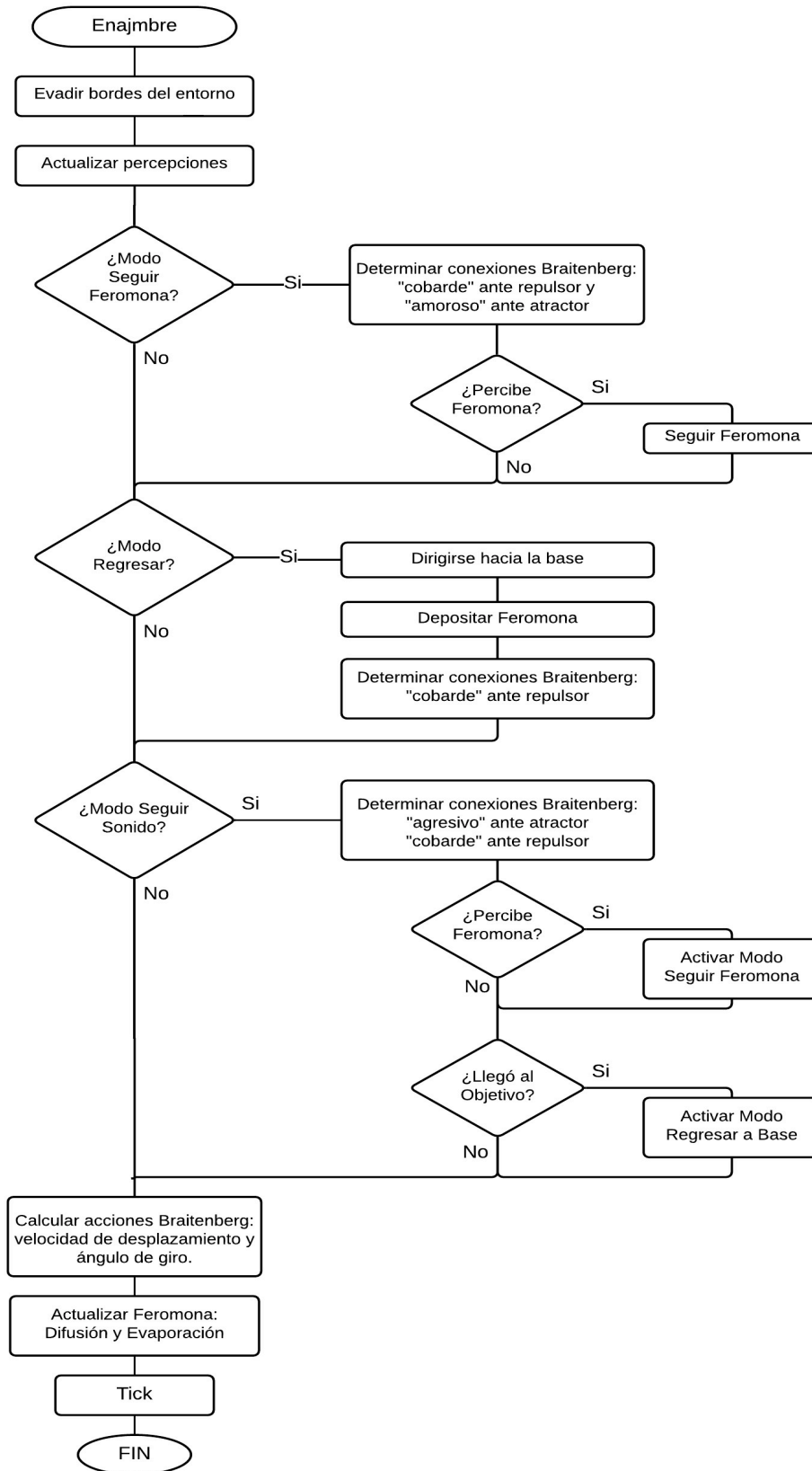


Figura 46. Diagrama de flujo del algoritmo de Enjambre, con sus respectivos modos de operación.

9.1.2.4.2 Estrategia Multiagente

Para que un sistema de agentes sea considerado un Sistema Multiagente, este debe contar con las siguientes características:

- Compuesto por varios (al menos 2) agentes.
- Los agentes pueden ser diferentes.
- Debe haber agentes autónomos (al menos 1).
- No se tiene una arquitectura preestablecida.
- El logro del objetivo está definido por las metas dadas por los agentes autónomos.



Figura 47. Características de un sistema Multiagente.

Los robots autónomos salen de la base explorando en direcciones aleatorias, una vez perciben el sonido del objetivo se dirigen a él y envían una señal a los robots servidores para que se dirijan al encuentro de sus líderes; el robot autónomo que esté más cerca del objetivo percibe su luz, en este momento envía una señal a los demás robots autónomos para que se dirijan hacia él.

Estrategia: Multiagente
Software de implementación: NETLOGO
Entorno: MATRIZ DE 128X58
Agentes: <ul style="list-style-type: none"> • <i>Tipos:</i> Autónomo, Servidor • <i>Número:</i> 30
Secuencia: <ul style="list-style-type: none"> • Los robots autónomos salen de la base explorando en direcciones aleatorias. • Una vez perciben el sonido del objetivo se dirigen a él.

<ul style="list-style-type: none"> • El robot autónomo que esté más cerca del objetivo percibe su luz, entonces envía una señal a los demás robots autónomos para que se dirijan hacia él. • Los demás robots autónomos envían una señal a los robots servidores para que se dirijan al encuentro de sus líderes. • Cuando están más cerca del sonido, perciben su luz; cada estímulo captado aumenta la prioridad de encaminamiento hacia el objetivo. • Los robots hacen contacto con el objetivo..
<p>Pseudocódigo:</p> <p><i>Inicio del ciclo</i></p> <p><i>pedir a los agentes autónomos:</i> <i>Actualizar datos de los sensores del robot</i> <i>Explorar el entorno</i> <i>el agente es líder de autónomos?</i> <i>sí-> asigna conexiones nerviosas tipo Braitenberg a los motores:</i> <i>"amoroso" ante luz de atractor</i> <i>"cobarde" ante repulsor</i> <i>no-> hay algún líder de autónomos?</i> <i>sí-> entrar a modo seguir Líder de autónomos</i> <i>el agente escucha el objetivo?</i> <i>sí-> entrar a modo Líder de Servidores</i> <i>asigna conexiones nerviosas tipo Braitenberg a los motores:</i> <i>" amoroso " ante luz de atractor</i> <i>" cobarde " ante repulsor</i> <i>no-> el agente ve el objetivo?</i> <i>sí-> asigna conexiones nerviosas tipo Braitenberg a los motores:</i> <i>" amoroso " ante luz de atractor</i> <i>" cobarde " ante repulsor</i> <i>entrar a modo Líder de autónomos</i> <i>no-> el agente escucha el objetivo?</i> <i>sí-> entrar a modo Líder de Servidores</i> <i>asigna conexiones nerviosas tipo Braitenberg a los motores:</i> <i>" agresivo " ante sonido de atractor</i> <i>" cobarde " ante repulsor</i> <i>no-> asigna conexiones nerviosas tipo Braitenberg a los motores:</i> <i>" cobarde " ante repulsor</i></p> <p><i>pedir a los agentes servidores:</i> <i>Explorar el entorno</i> <i>hay algún líder de servidores?</i> <i>sí-> el autónomo correspondiente es líder de servidores?</i> <i>sí-> entrar a modo Seguir Líder de Servidores</i></p> <p><i>regresa al inicio del ciclo</i></p>

Figura 48. Algoritmo Multiagente implementado.

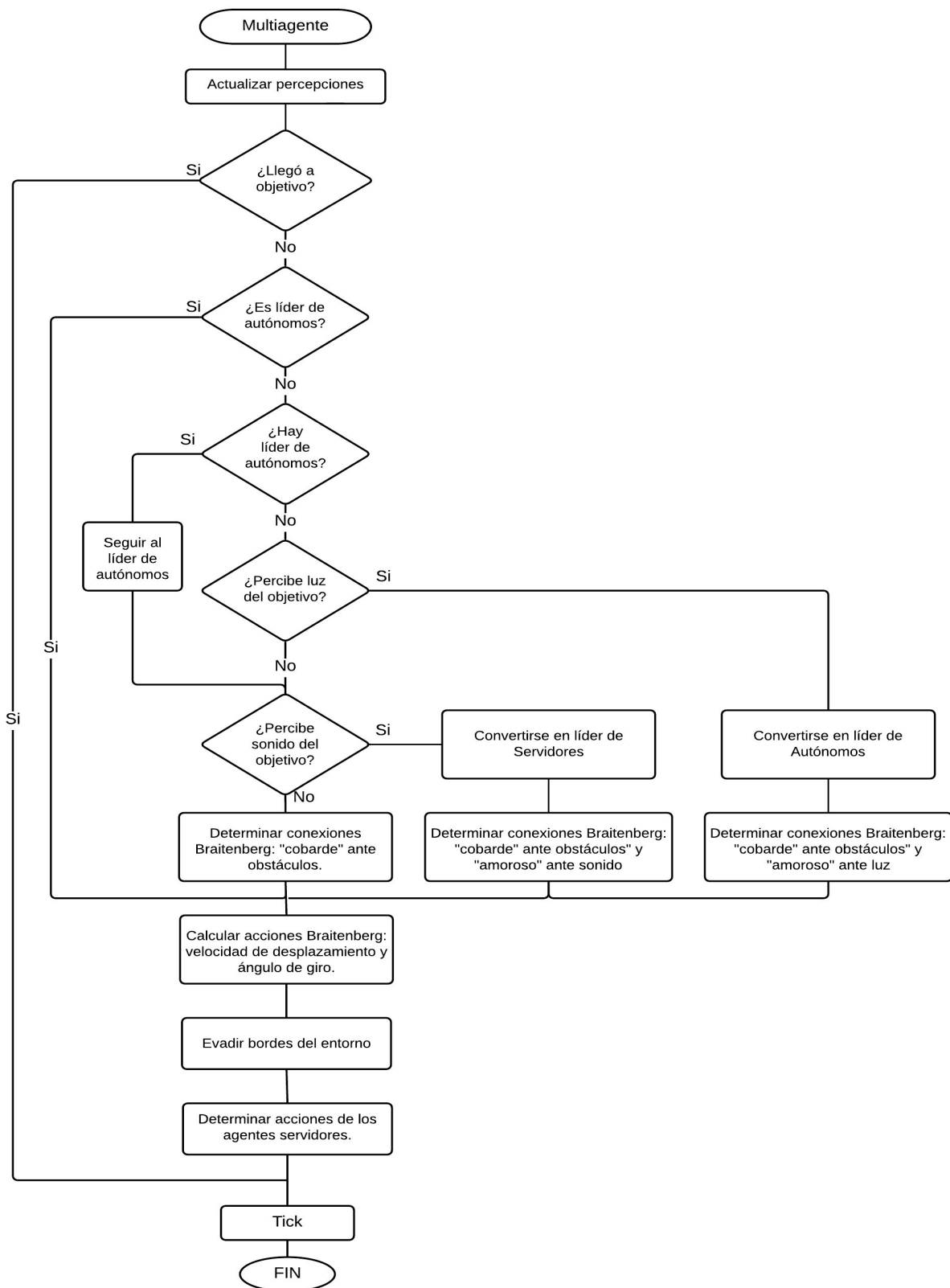


Figura 49. Diagrama de flujo del algoritmo Multiagente, con sus respectivos modos de operación.

9.1.3 Diseño

Aunque no se llevó a cabo la implementación física de la plataforma robótica, se propone un diseño a nivel de sistemas.

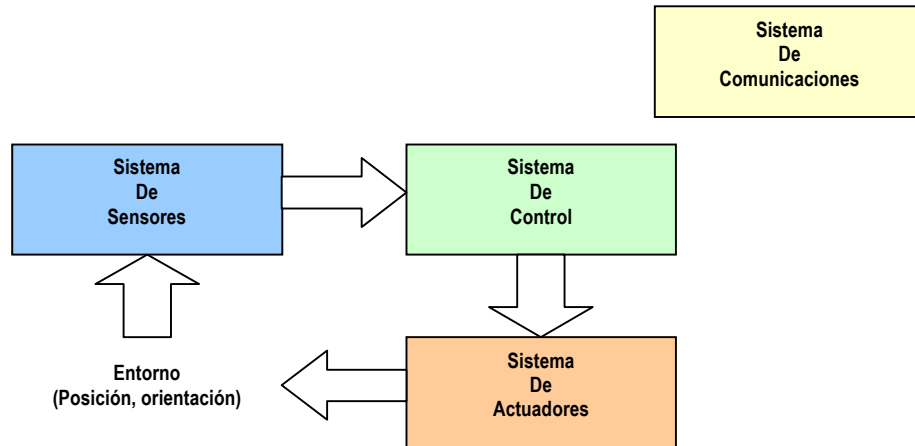


Figura 50. Sistemas de la plataforma robótica propuesta.

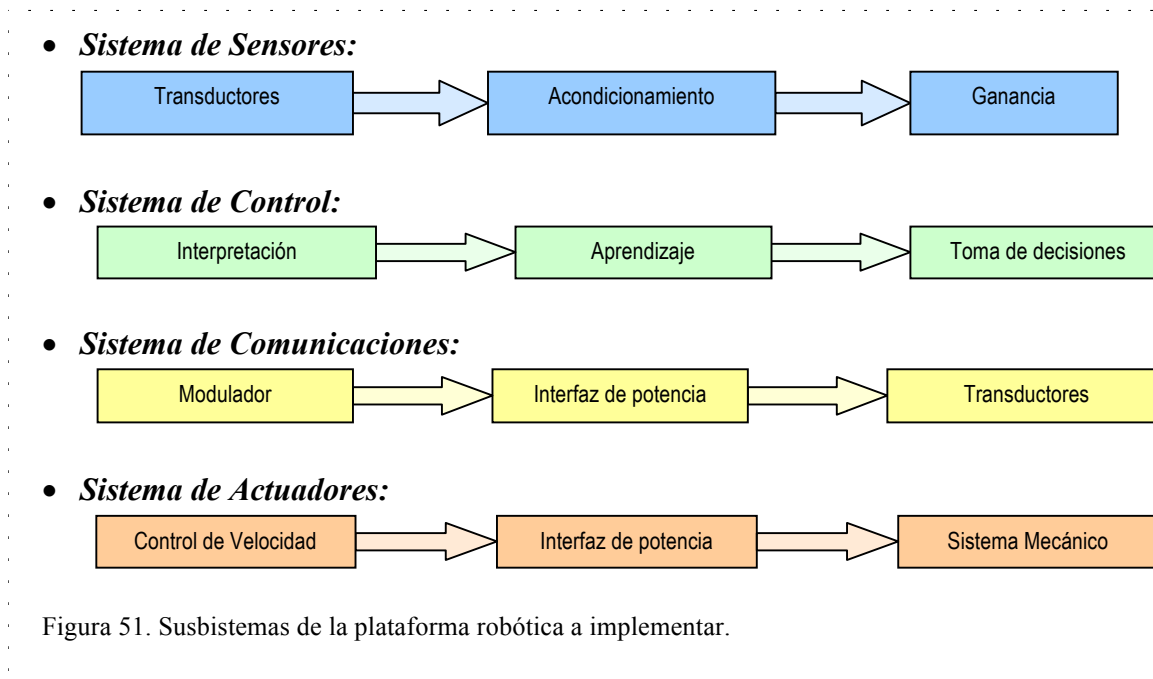


Figura 51. Subsistemas de la plataforma robótica a implementar.

La descripción detallada la arquitectura propuesta y de cada subsistema se encuentran en el ANEXO 2: Síntesis de un sistema nervioso y ANEXO 3: Identificación del sistema robótico.

9.2 METODOLOGÍA QFD

QFD (*Quality Function Deployment*) significa Despliegue de la Función de Calidad. Esto es, "transmitir" los atributos de calidad requeridos, a través de los procesos organizacionales, para que cada proceso pueda contribuir al aseguramiento de estas características (Tamayo, 2010).

QFD es un método de gestión de calidad basado en transformar las demandas del usuario en la calidad del diseño, implementar las funciones que aporten más calidad, e implementar métodos para lograr calidad del diseño en subsistemas y componentes, y en última instancia a los elementos específicos del proceso de fabricación (Akao, 1994).

9.2.1 Fases Del QFD

- **Planificación de Productos:** Se documentan los requerimientos, las oportunidades de competencia, las dimensiones del producto, las medidas de competencia de los productos y la capacidad técnica para satisfacer cada necesidad. Conseguir buenos datos en esta fase es fundamental para el éxito de todo el proceso de QFD. En muchas ocasiones sólo se pasa por esta fase.
- **Diseño de Productos:** Se crean los conceptos de productos y se documentan las especificaciones de las partes.
- **Planificación de Procesos:** Los procesos de fabricación son descritos en diagramas de flujo y son documentados los parámetros del proceso (o los valores objetivos).
- **Planificación de la Producción:** Se crean indicadores de rendimiento para supervisar el proceso de producción, se toman decisiones en cuanto a cuál proceso supone el mayor riesgo y se ponen en marcha controles para prevenir fallas.

9.2.2 Implementación Del QFD

Hay varias herramientas utilizadas para implementar la primera fase de QFD, la más utilizada es la *House Of Quality* (HOC) o Casa de la Calidad. Según la Asociación Latinoamericana de QFD, esta metodología se implementa siguiendo los siguientes pasos:

- Seleccionar un producto/servicio importante a mejorar
- Obtener la voz del cliente
- Extraer las necesidades del cliente
- Organizar las necesidades del cliente
- Priorizar las necesidades del cliente
- Establecer los parámetros de diseño
- Generar la matriz de relaciones
- Obtener la evaluación de desempeño del cliente
- Correlacionar los parámetros de diseño
- Analizar los resultados
- Iterar el proceso

9.3 METODOLOGÍA DE DISEÑO DE EXPERIMENTOS (DOE)

En un experimento, deliberadamente cambiamos uno o mas factores (variables), con la finalidad de observar de que manera los cambios afectan en la(s) variable(s) de salida o de respuesta. Es un procedimiento eficaz para la planeación de experimentos, con miras a que los datos obtenidos puedan ser analizados y así producir conclusiones válidas y objetivas (NIST & SEMATECH, 2012).

El diseño estadístico de experimentos es precisamente la forma más eficaz de hacer pruebas. El diseño de experimentos consiste en determinar cuáles pruebas se deben realizar y de qué manera, para obtener datos que, al ser analizados estadísticamente, proporcionen evidencias objetivas que permitan responder las interrogantes planteadas, y de esa manera clarificar los aspectos inciertos de un proceso, resolver un problema o lograr mejoras (Montgomery, 2005).

Una de los ejemplos en donde el diseño experimental puede ser usado de manera efectiva consiste en la selección de los factores claves que afectan a una respuesta dada. Con frecuencia existen muchos factores, algunos mas críticos que otros, los cuales pueden tener pequeña o gran influencia en una respuesta. Es deseable el reducir el número de factores a un pequeño grupo (2 a 5) para que la atención pueda dirigirse en controlar aquellos con las especificaciones apropiadas.

9.3.1 Diseño factorial completo de dos niveles:

El objetivo de un diseño factorial es estudiar el efecto de varios factores sobre una o varias respuestas, cuando se tiene el mismo interés sobre todos los factores. Los factores pueden ser de tipo cualitativo (máquinas, tipos de material, operador, la presencia o ausencia de una operación previa, etc.), o de tipo cuantitativo (temperatura, humedad, velocidad, presión, etc.). Para estudiar la manera en que influye cada factor sobre la variable de respuesta es necesario elegir al menos dos niveles de prueba para cada uno de ellos. La matriz de diseño o arreglo factorial es el conjunto de puntos experimentales o tratamientos que pueden formarse considerando todas las posibles combinaciones de los niveles de los factores. Si existen k factores, cada uno con dos niveles, un diseño factorial completo se ejecuta 2^k veces (Gutiérrez & de la Vara, 2008).

9.3.2 Diseño factorial fraccionado:

La ASQC (Sociedad americana para el control de la calidad) define el diseño factorial fraccionado de la siguiente forma “Un experimento factorial en el que sólo una fracción adecuada de entre las combinaciones necesarias para un experimento factorial completo, es seleccionada para ser ejecutada” (NIST & SEMATECH, 2012).

Aún cuando el numero de factores, k , en un diseño sea pequeño, las 2^k ejecuciones se vuelven muy tediosas. La solución a este problema es usar solo una fracción de las ejecuciones especificadas por el diseño factorial. Cuales ejecuciones realizar y cuales dejar fuera, son el punto de interés aquí. En general, se pueden elegir fracciones como $\frac{1}{2}$, $\frac{1}{4}$, etc. de las ejecuciones arrojadas por el análisis factorial completo. Las fracciones apropiadas para un

diseño factorial de experimentos, para 2 niveles tienen las propiedades de ser balanceadas y ortogonales.

9.3.3 Configuración de parámetros óptimos:

Para obtener la configuración óptima de las variables involucradas en cada estrategia, utilizando la metodología de Diseño de Experimentos DOE, se realizó un diseño factorial fraccionado con un factor de $\frac{1}{4}$, de dos niveles, con cuatro factores k (variables). El número de iteraciones es igual a multiplicar el fraccionario por el número total de iteraciones: $2^4 = 16$, lo que es igual a 4 iteraciones como mínimo. Se utilizaron 5 iteraciones para cada algoritmo.

Cada factor posee dos niveles (alto y bajo). La selección de los factores se realizó con base en la revisión de la bibliografía más relevante para ambas estrategias. Se realizaron de manera preliminar varias ejecuciones de ambos algoritmos para determinar una buena configuración de los niveles de alto y bajo, para los factores de cada estrategia utilizada.

Los factores, los niveles, y el costo promedio obtenido para cada una de las estrategias se muestran en la Tabla 1.

Tabla 1:

Configuración de Factores y Niveles Estrategias de Enjambre y Multiagente

Estrategia	Factores	Niveles		Costo Promedio [pasos]
		Bajo (-1)	Alto (+1)	
ESTRATEGIA DE ENJAMBRE	Sensibilidad al repulsor	50	75	606,06
	Sensibilidad al atractor	20	40	
	Difusión de feromona	10	20	
	Evaporación de feromona	5	10	
ESTRATEGIA MULTIAGENTE	Sensibilidad al repulsor	50	75	2258,18
	Tasa de autonomía	10	20	
	Sensibilidad a la luz	20	40	
	Sensibilidad al sonido	5	30	

Nota: Corresponde a los factores elegidos por revisión, los niveles obtenidos en simulaciones preliminares, y el costo promedio obtenido para cada una de las estrategias al aplicar el Diseño de Experimentos.

Las evaluaciones están basadas en el valor promedio de costo, obtenido al correr 5 veces cada algoritmo, de acuerdo a los diferentes parámetros. Al evaluarse la eficiencia, la característica del problema es encontrar el valor mínimo de tiempo, medido en ticks (pasos del algoritmo). En la Tabla 2 se encuentran los niveles de los factores con los que se obtuvo el valor mínimo de costo. Para observar la totalidad de los datos, referirse al ANEXO 5: Pruebas estadísticas.

Tabla 2:

Valores óptimos obtenidos para cada factor

Estrategia	Factores	Niveles óptimos	Costo
			[pasos]
ESTRATEGIA DE ENJAMBRE	Sensibilidad al repulsor	50	528,60
	Sensibilidad al atractor	20	
	Difusión de feromona	20	
	Evaporación de feromona	5	
ESTRATEGIA MULTIAGENTE	Sensibilidad al repulsor	75	2047,20
	Tasa de autonomía	10	
	Sensibilidad a la luz	40	
	Sensibilidad al sonido	30	

Nota: Corresponde al costo mínimo obtenido para cada estrategia y los niveles que dieron estos resultados.

Los cubos para cada una de las estrategias muestran las combinaciones óptimas obtenidas, para los factores:

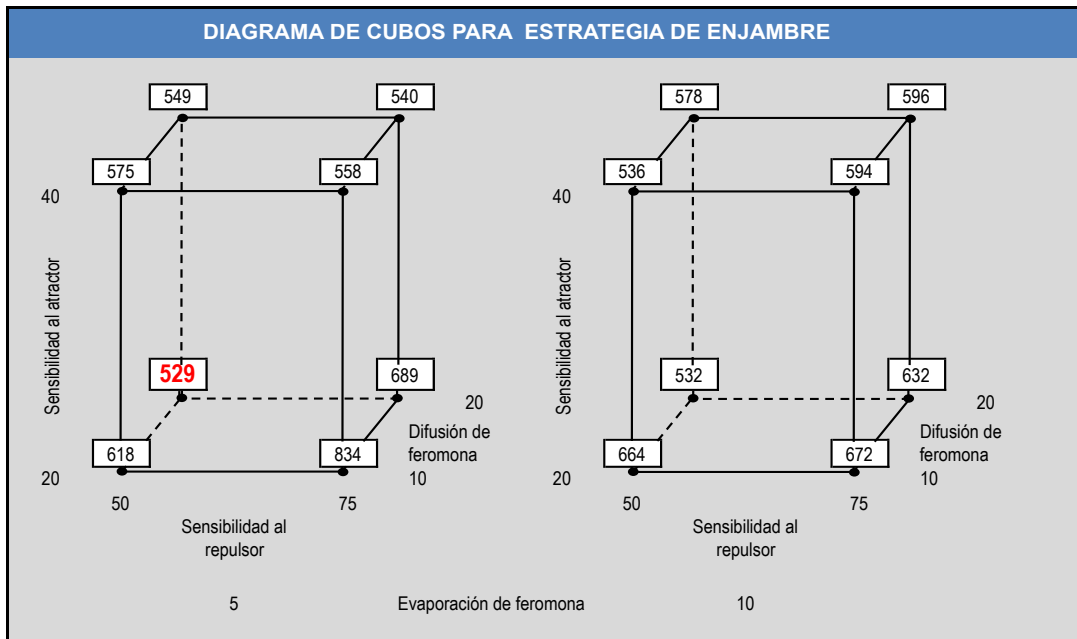


Figura 52. Diagramas de Cubos o de promedios de costos para las Estrategia de Enjambre. El menor costo se señala con color rojo.

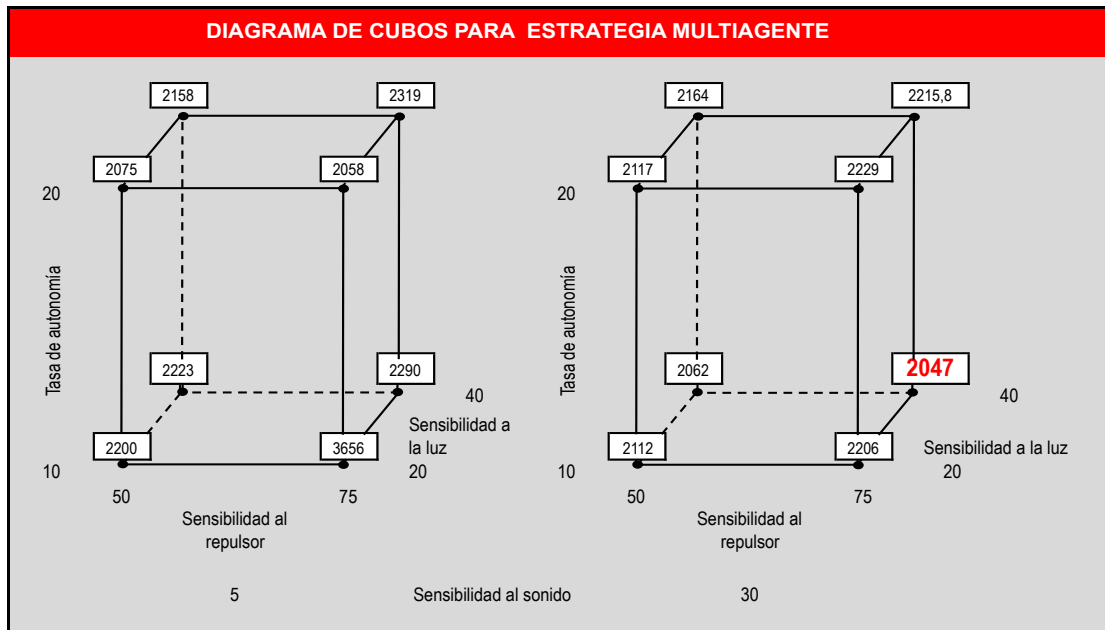


Figura 53. Diagramas de Cubos o de promedios de costos para las Estrategia Multiagente. El menor costo se señala con color rojo.

Las evaluaciones están basadas en el valor promedio de costo, obtenido al correr 5 veces cada algoritmo, de acuerdo a los diferentes parámetros. Al evaluarse la eficiencia, la característica del problema es encontrar el valor mínimo de tiempo (medido en pasos).

10 RESULTADOS

10.1 INTRODUCCIÓN

Los sistemas multirobot presentan grandes ventajas frente a los sistemas robóticos tradicionalmente usados en labores de exploración. Concretamente, el presente proyecto no puede ser implementado en un ambiente real, pues las condiciones de desarrollo hacen que se encuentre en una etapa de experimentación; sin embargo, se dan las pautas para implementar el mismo sistema de múltiples microbots reactivos, en sistemas de monitoreo y exploración.

En este capítulo se plantea, como primera medida, cuales son los resultados esperados a partir del logro de los objetivos planteados, los alcances y limitaciones, los algoritmos utilizados y como se comportan en el marco de un sistema robótico colaborativo, la elección del software utilizado, el entorno de simulación implementado y por último, se plantean una serie de casos en donde se implementaron dichos algoritmos y se analiza su comportamiento.

10.2 RESULTADOS ESPERADOS A PARTIR DEL LOGRO DE LOS OBJETIVOS PLANTEADOS

Con el logro del objetivo general de este trabajo (*Simular y contrastar el desempeño de estrategias de navegación, basadas en inteligencia colaborativa, en entornos estáticos para robots móviles*), se busca un avance significativo para continuar con el estudio de sistemas complejos y comportamientos emergentes.

10.2.1 Objetivo 1 “Desarrollar una estrategia de navegación para robótica móvil, aplicable a sistemas multirobot”: Este objetivo específicos está enfocado hacia el logro de estrategias de navegación eficientes, con miras a solucionar los problemas típicos de la robótica móvil. La estrategia de navegación para robótica móvil desarrollada se basó en algoritmos de enjambre y multiagente, mediante:

- Búsqueda y desarrollo de rutinas de comportamientos para los robots.
- Búsqueda y desarrollo de rutinas de navegación.

Con base en esto se implementaron los Algoritmos de Enjambre y Multiagente, utilizando ambos la técnica de navegación de Campos Potenciales.

10.2.2 Objetivo 2 “Contrastar el desempeño de la técnica multiagente y de la técnica de enjambre, en la solución del problema de navegación en entornos estáticos, usando el entorno virtual”: Con este objetivo se espera obtener una caracterización de las técnicas de

inteligencia artificial usadas en sistemas multirobot (multiagente y enjambre), al determinar las situaciones para las cuales cada una de ellas presenta mejor desempeño.

- Búsqueda de software para simulaciones, que permitió implementar diversos comportamientos en los robots, así como también la simulación de entornos donde interactúan los robots.
- Existen muchos criterios que han sido considerados para evaluar los algoritmos que serán implementados. Se tomaron los criterios más relevantes para los objetivos del proyecto, teniendo en cuenta la bibliografía, los antecedentes y las conclusiones de investigaciones similares y pertinentes.

Con base en lo anterior, se estableció como plataforma para la simulación el software NETLOGO (Wilensky & Rand, 2015) y se determinaron los criterios establecidos en los resultados.

10.2.3 Objetivo 3 “Establecer los resultados derivados de la observación de las acciones e interacciones, simuladas y generadas a partir del comportamiento de los robots”: Con el logro de este objetivo se espera obtener un comportamiento emergente a partir de las interacciones entre los robots en un ambiente desconocido.

- Se evaluaron las ventajas y desventajas de los algoritmos para elegir aquellos con los cuales se va a trabajar, y con ellos se elaboró una tabla en la que se asigna un peso específico genérico para cada uno de los criterios que han sido considerados.
- Desarrollo, observación y análisis de simulaciones. Se utilizaron pruebas estadísticas para comparar el desempeño de los algoritmos y se realizó la obtención de parámetros óptimos mediante la metodología de Diseño de Experimentos DOE.

10.3 ALCANCES Y LIMITACIONES

Los alcances y las limitaciones del proyecto están ligados a las características experimentales del mismo y, una vez implementado en un entorno real (lo cual está fuera del alcance de este trabajo), la mayoría de las limitaciones habrán desaparecido, al mismo tiempo que los alcances incrementarían (ambas cosas se lograrían al cambiar el sistema de locomoción empleado en el diseño, por uno más adecuado a las condiciones requeridas).

Aunque el proyecto no comprende la construcción de hardware, se plantea el diseño básico del microrobot, con las características adecuadas para ser utilizado en un entorno de laboratorio, y destinado al estudio del problema de navegación en robótica móvil. El diseño del microrobot puede ser utilizado para el estudio de sistemas complejos y comportamientos emergentes, al construir varios de estos robots.

Al hacer uso de una arquitectura reactiva, el microprocesador no se ocupa de tareas menores, estas están a cargo de circuitos especialmente diseñados para efectuar acciones reflejas como respuesta a estímulos ajenos a la actividad principal del microbot, y que podrían distraerlo de su objetivo.

Los microbots diseñados en este proyecto utilizan ruedas como medio de locomoción, sin embargo el mismo sistema reactivo puede implementarse en microbots con diferentes formas de locomoción como orugas, patas, hélices, aletas, etc.; todo depende del medio en el cual deban desenvolverse. Las características del sistema de locomoción elegido en el diseño, la tracción por ruedas, hace imposible el desplazamiento por terrenos irregulares y con pronunciados desniveles; así mismo el desplazamiento por planos inclinados con pendientes pronunciadas.

Estos microbots no ser llevados a ambientes con condiciones adversas de clima; las temperaturas extremas, la humedad, el polvo, etc. afectan su funcionamiento, y deben emplearse solo en condiciones de laboratorio; no obstante la electrónica interna y la arquitectura empleada, partes fundamentales de este proyecto, pueden implementarse en microbots más adecuados para desempeñarse efectivamente a la intemperie y en terrenos irregulares.

10.4 PRODUCTOS ESPERADOS

Una vez finalizado el presente trabajo, serán entregados los siguientes productos, dando sustento al desarrollo del mismo:

- Descripción de las estrategias de navegación y los algoritmos empleados.
- Diseño del algoritmo de enjambre mediante ACO.
- Diseño de los algoritmos mediante la metodología MAD-Smart.
- Obtención de los parámetros óptimos de cada algoritmo mediante la metodología de Diseño de Experimentos DOE.
- Bases de datos numéricas de los resultados y tablas de las pruebas estadísticas realizadas, en Excel.
- Simulaciones (pantallazos, programas y videos).
- Contraste entre las estrategias utilizadas utilizando pruebas estadísticas.

10.5 RESULTADOS DE LA IMPLEMENTACIÓN DE LOS ALGORITMOS

Se presentan los resultados para la estrategia de Inteligencia de enjambre y la Multiagente. El comportamiento de los algoritmos para diferentes valores se evalúa mediante su simulación en los algoritmos construidos en la plataforma elegida NETLOGO.

10.5.1 Casos Implementados

Para cada estrategia se analiza el caso óptimo, usando los factores con los que se obtuvo el menor costo.

Para los datos obtenidos con las simulaciones, se tuvieron en cuenta una serie de factores constantes, los cuales se observan en la *Tabla 3*.

Tabla 3:

Parámetros iniciales para las simulaciones

<i>Estrategia Enjambre</i>				
Parámetro	Caso óptimo			
Numero de agentes	100			
Número de obstáculos	10			
Tamaño del entorno	MATRIZ DE 128x58			
Porcentaje de llegada de los agentes al objetivo	80%			
Sensibilidad al repulsor	Sensibilidad al atractor	Difusión de feromona	Evaporación de feromona	Costo
50	20	20	5	396
<i>Estrategia Multiagente</i>				
Parámetro	Caso óptimo			
Numero de agentes autónomos + servidores	30			
Número de obstáculos	10			
Tamaño del entorno	MATRIZ DE 128x58			
Porcentaje de llegada de los agentes al objetivo	80%			
Sensibilidad al repulsor	Tasa de autonomía	Sensibilidad a la luz	Sensibilidad sonido	Costo
75	10	20	30	1935

Nota: Parámetros constantes para realizar la simulación y factores óptimos obtenidos como resultado.

A continuación se presentan los resultados de menor costo para cada estrategia, correspondientes a una combinación de factores específica.

Caso óptimo Enjambre:

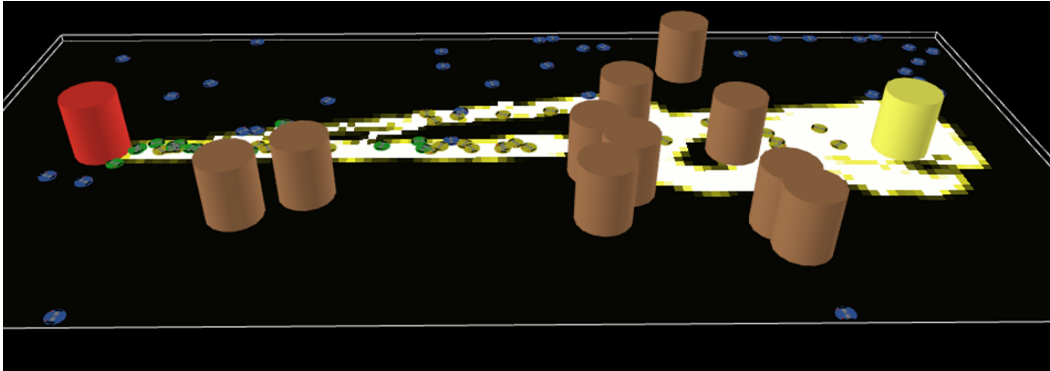


Figura 54. Simulación de la estrategia de enjambre en NETLOGO. En rojo el punto de inicio y en amarillo la meta. Se observan los caminos usados por los agentes para llegar a su objetivo final.

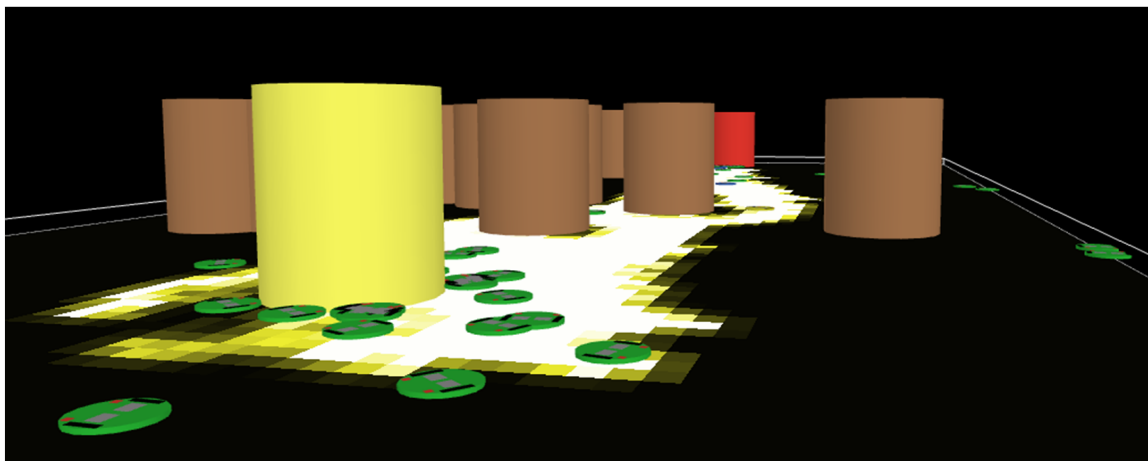


Figura 55. Llegadas al objetivo en la estrategia de enjambre en NETLOGO. Se observan a varios agentes rodeando la meta.

Se obtiene el comportamiento del sistema de enjambre bajo los valores óptimos de los factores, en este caso sobre 7 repeticiones, como se observa en la Figura 56 y posteriormente se obtiene el promedio de estas gráficas con el fin de analizar el comportamiento sistema.

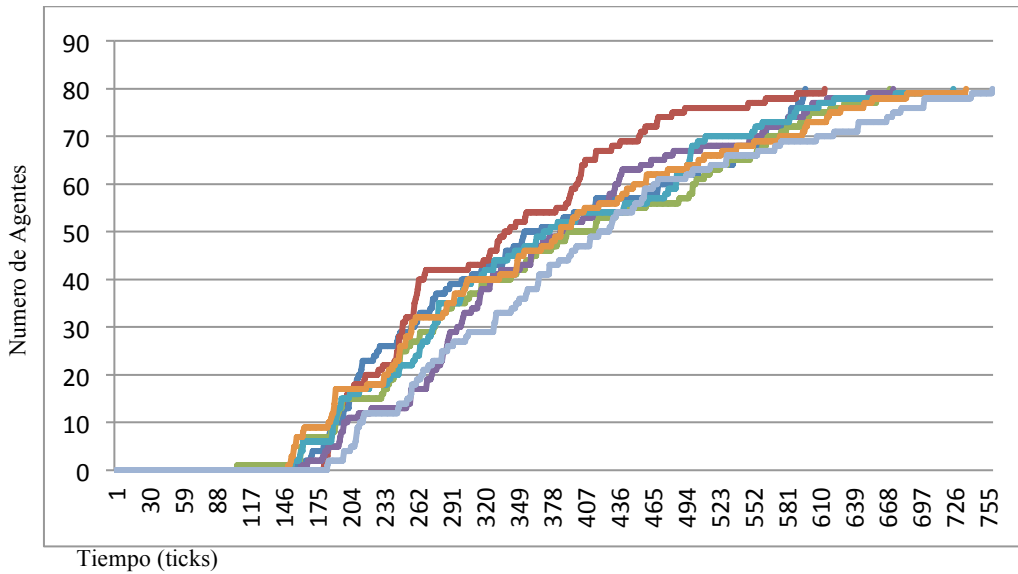


Figura 56. Comportamiento de la estrategia de enjambre en 7 simulaciones con los parámetros óptimos. Número de agentes que llegan al objetivo versus tiempo en ticks.

Como primera medida se observó si el comportamiento del sistema presenta no linealidades, para esto se acotó la gráfica promedio para eliminar las inconsistencias por los datos extremos constantes, es decir, se tomó como primer valor de abscisas el mayor de los 7 tiempos correspondientes a la primera llegada; y como último valor de abscisas el menor de los 7 tiempos correspondientes a la llegada número 80 (el 80% del total de robots). Se observa en la Figura 57 que la gráfica del comportamiento del sistema en este intervalo es casi una línea recta.

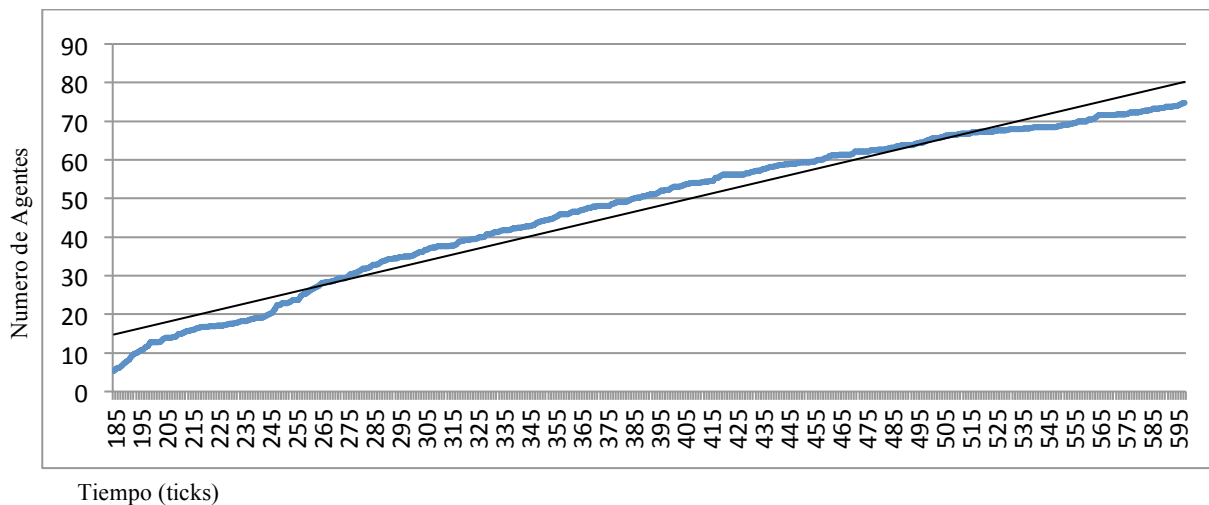


Figura 57. Comportamiento de la estrategia de enjambre, con los parámetros óptimos, en intervalo lineal.

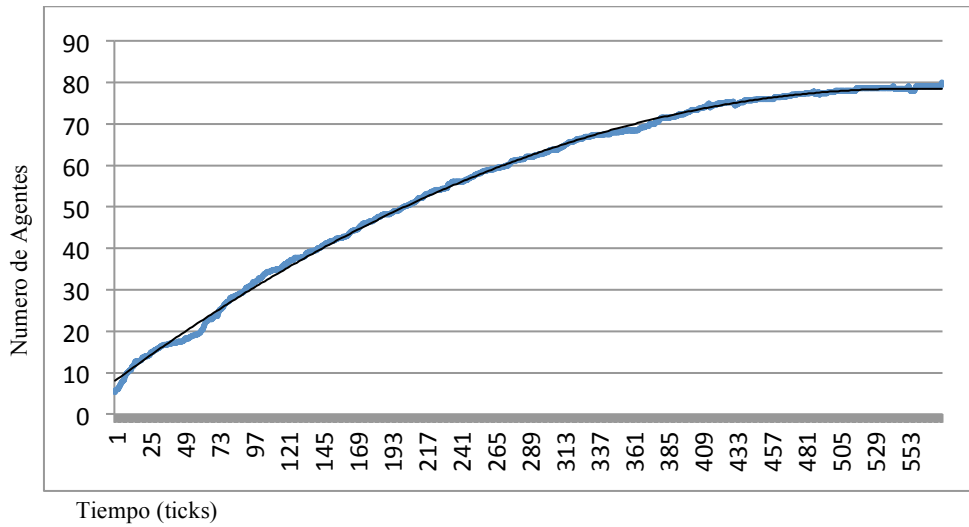


Figura 58. Comportamiento logarítmico de la estrategia de enjambre.

Caso óptimo Multiagente:

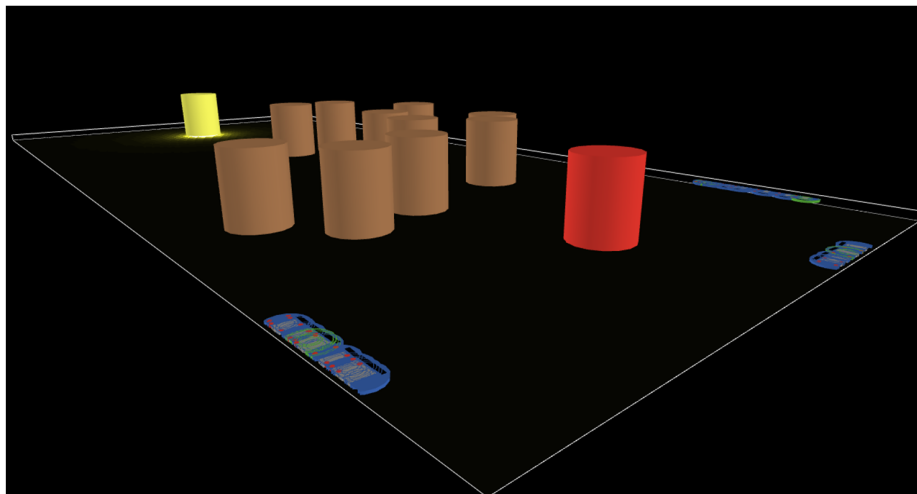


Figura 59. Simulación de la estrategia multiagente en NETLOGO. En rojo el punto de inicio y en amarillo la meta. Se observa la estrategia de los agentes para evadir los obstáculos.

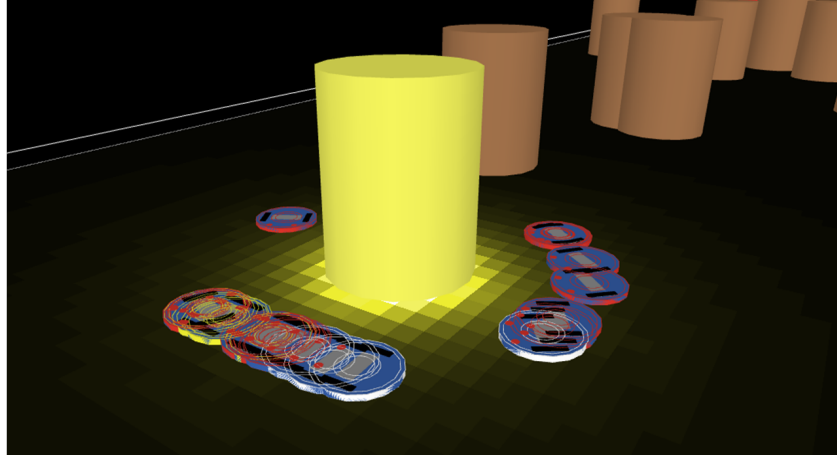


Figura 60. Llegadas al objetivo en la estrategia multiagente en NETLOGO. Se observan a varios agentes rodeando la meta (en amarillo).

Se obtiene el comportamiento del sistema multiagente bajo los valores óptimos de los factores, en este caso sobre 7 repeticiones, como se observa en la Figura 61; posteriormente se obtiene el promedio de estas gráficas con el fin de analizar el comportamiento sistema.

Como primera medida se observó si el comportamiento del sistema presenta no linealidades, para esto se acotó la gráfica promedio para eliminar las inconsistencias por los datos extremos constantes, es decir, se tomó como primer valor de abscisas el mayor de los 7 tiempos correspondientes a la primera llegada; y como último valor de abscisas el menor de los 7 tiempos correspondientes a la llegada número 24 (el 80% del total de robots). Se observa en la Figura 62 que el comportamiento del sistema en este intervalo es lineal.

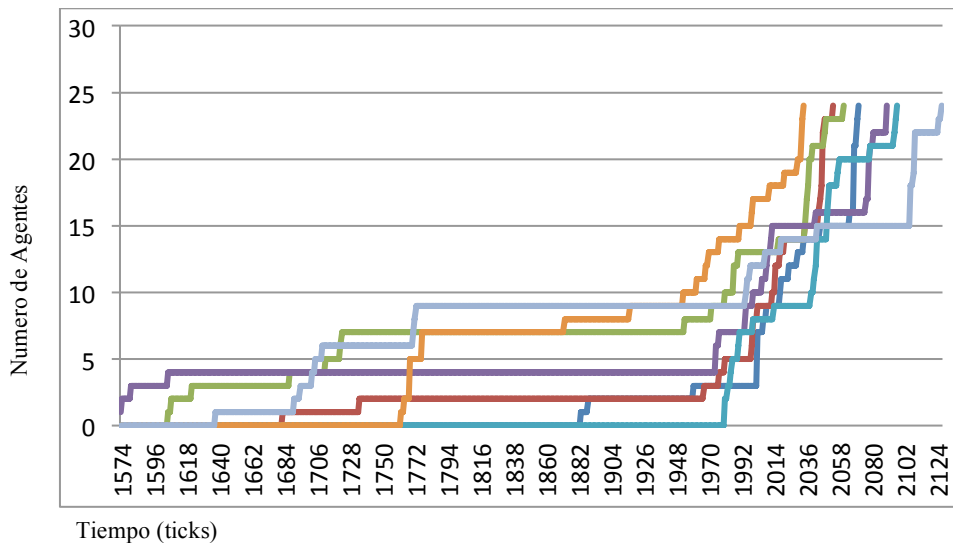


Figura 61. Comportamiento de la estrategia multiagente en 7 simulaciones con los parámetros óptimos.

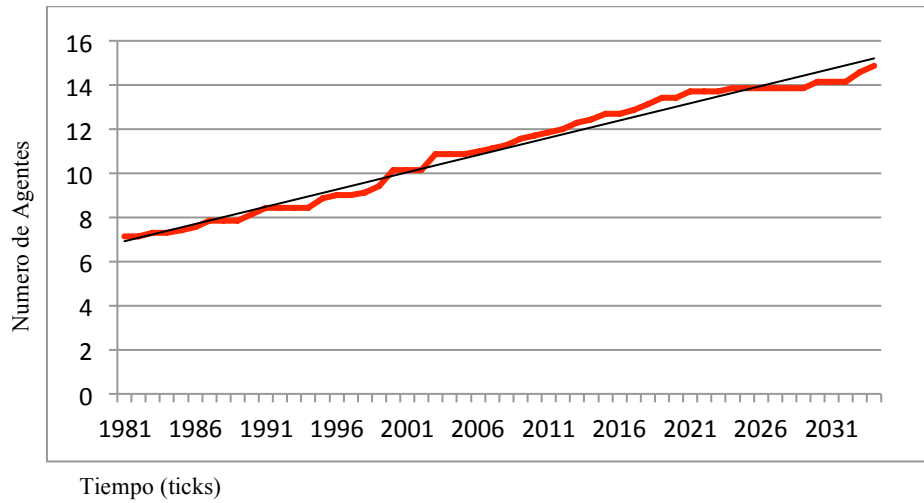


Figura 62. Comportamiento de la estrategia multiagente, con los parámetros óptimos, en intervalo lineal.

La gráfica promedio de la figura 63, sin la cota inferior, muestra que el comportamiento del sistema obedece a una curva exponencial.

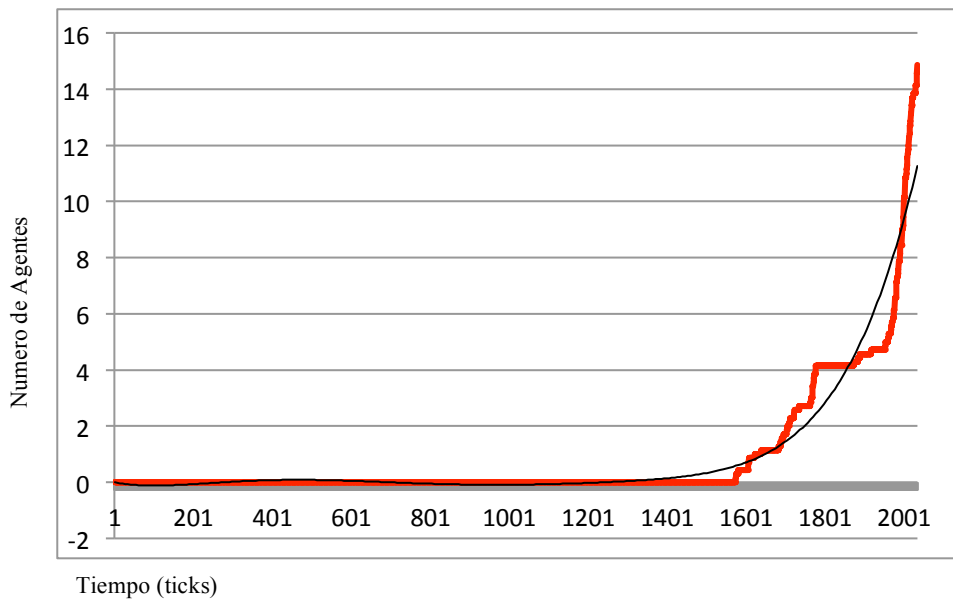


Figura 63. Comportamiento exponencial de la estrategia multiagente.

11 DISCUSION DE RESULTADOS

11.1 INTRODUCCIÓN

En éste capítulo se presentan el análisis de los resultados, para cada uno de los algoritmos y los casos considerados, además del análisis estadístico de los datos arrojados por las simulaciones, según las pruebas de normalidad y homocedasticidad (igualdad de varianza) descritas en el capítulo de operacionalización de variables, con el objeto de probar la hipótesis a realizar y proceder a realizar la comparación de los resultados obtenidos mediante la prueba de ANOVA y el diagrama de Caja y Bigotes.

11.2 ANÁLISIS DE RESULTADOS

De acuerdo a los resultado obtenidos en la metodología de Diseño de Experimentos, se realiza la comparación de las estrategias de enjambre y multiagente, con el menor costo (menor número de ticks). En total se realizaron 16 ejecuciones de cada algoritmo, con 5 iteraciones por ejecución, realizando todas las combinaciones posibles de los valores correspondientes a los 4 factores elegidos. El número de ticks se estableció como mínimo, cuando el 80% de los agentes llegaban a su objetivo. Se establece un nivel de significancia de 0.05, es decir, el riesgo de equivocarse y rechazar erróneamente la hipótesis nula es del 5%, obteniendo un 95% de confianza.

Para mayor claridad acerca de los datos obtenidos y las pruebas estadísticas realizadas, dirigirse al ANEXO 5 PRUEBAS ESTADISTICAS.

Se describen los resultados detallados de las pruebas estadísticas para el **Caso 1** en el que los valores de costo para ambos algoritmos es el mínimo (valores óptimos).

11.2.1 Análisis de las simulaciones correspondientes a la estrategia de Enjambre:

Desempeño de la estrategia:

La Figura 64 muestra el desempeño de la estrategia con cada una de las 16 posibles configuraciones de los factores, en el eje de las abscisas se representan los 16 grupos de 5 repeticiones con la misma configuración de factores; en el eje de ordenadas se representa la diferencia en cantidad de tiempo, ticks o pasos para llegar al objetivo, respecto al promedio total (que es de 606,06 pasos); los valores negativos indican mayor cantidad de ticks que el promedio, los valores positivos indican menor cantidad de ticks que el promedio.

Puede verse que con la configuración de factores usada en las simulaciones del grupo 9, se obtiene un desempeño extremadamente pobre, con alrededor de 230 ticks más que la cantidad promedio de pasos para llegar al objetivo. La configuración que da mejores resultados es la correspondiente al grupo de simulaciones número 3, la cual logra un ahorro de casi 80 ticks respecto al promedio.

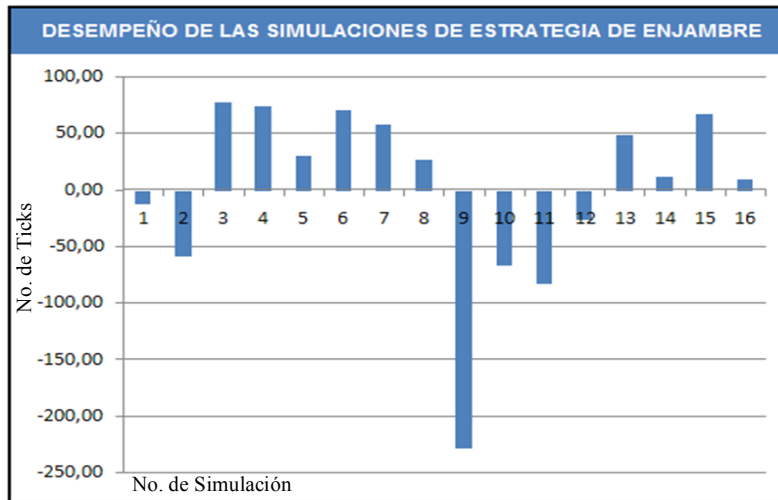


Figura 64. Desempeño de la estrategia de enjambre en cada una de las 16 iteraciones. En 0,00 se encuentra el promedio. Los valores positivos corresponden a ahorro de tiempo (ticks) y los negativos,

Efectos de cada factor:

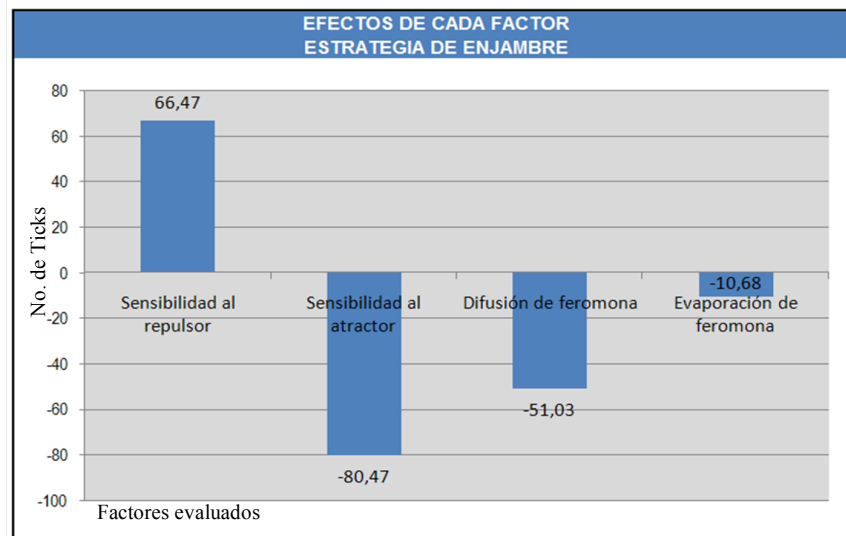


Figura 65. Efecto de los factores en el tiempo de llegada promedio en la estrategia de enjambre. En 0 el valor promedio de llegada; valores positivos disminuyen el tiempo de llegada, mientras que los negativos lo aumentan.

En la figura 65 puede verse el efecto que tiene cada factor, de manera independiente, sobre el tiempo de llegada del enjambre al punto de objetivo; los valores indicados corresponden al aumento promedio (o disminución promedio) que ejerce cada factor sobre el tiempo promedio de llegada.

Uno de los factores mas influyentes es *Sensibilidad al atractor*; un aumento en este factor, del nivel bajo al nivel alto, disminuye el tiempo de llegada (Figura 66b) en mayor medida que los factores *Difusión de Feromona* y *Evaporación de feromona*.

Un aumento en *Sensibilidad al repulsor* del nivel bajo al nivel alto, aumenta considerablemente el tiempo de llegada (Figura 66a).

El factor menos influyente es *Evaporación de feromona*, un cambio en este factor casi no produce efecto, un cambio del nivel bajo al nivel alto en *Evaporación de feromona* disminuye levemente el tiempo de llegada (Figura 66d).

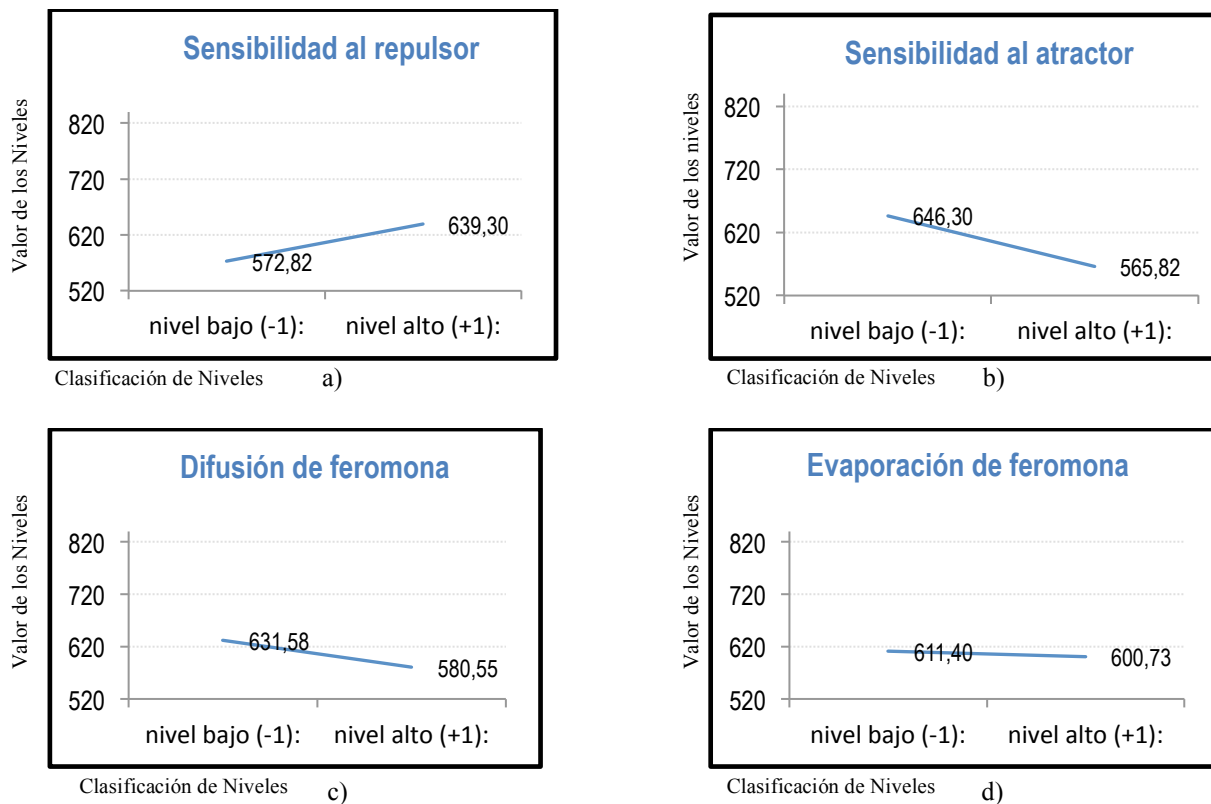


Figura 66. Efectos independientes de cada factor relacionado con la estrategia de Enjambre. a) Sensibilidad al repulsor. b) Sensibilidad al atractor. c) Difusión de feromona. d) Evaporación de feromona.

Interacciones:

Se evaluó también el efecto que tenía un factor, sobre otro, para cada estrategia, con el fin de describir sus interacciones e influencias.

1. *Variable principal*: Sensibilidad al repulsor. *Variable condicionada*: Sensibilidad al atractor

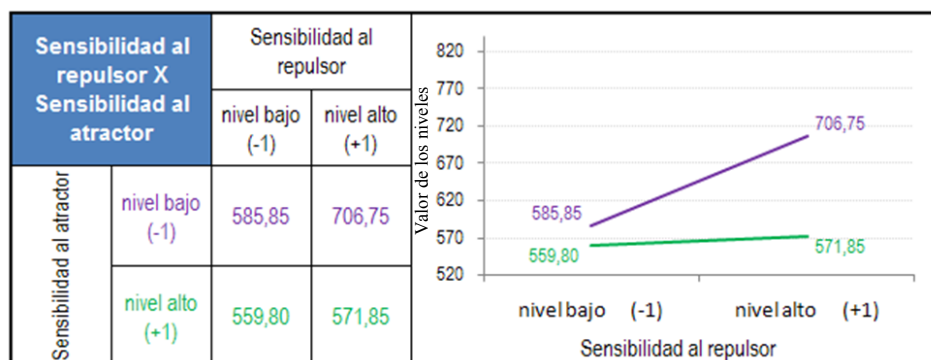


Figura 67. Influencia de la sensibilidad al repulsor sobre la sensibilidad al atractor, en la estrategia de enjambre.

En la Figura 68 se observa que con un nivel bajo en *Sensibilidad al atractor*, un cambio de un nivel bajo a un nivel alto en *Sensibilidad al repulsor* hace aumentar sus valores. Con un nivel alto en *Sensibilidad al atractor*, un cambio de un nivel bajo a un nivel alto en *Sensibilidad al repulsor* casi no produce efecto, hace aumentar levemente sus valores. El cambio en *Sensibilidad al repulsor* es contrarrestado por la interacción, resultando una acción casi nula sobre *Sensibilidad al atractor*.

2. *Variable principal*: Sensibilidad al repulsor. *Variable condicionada*: Difusión de feromona

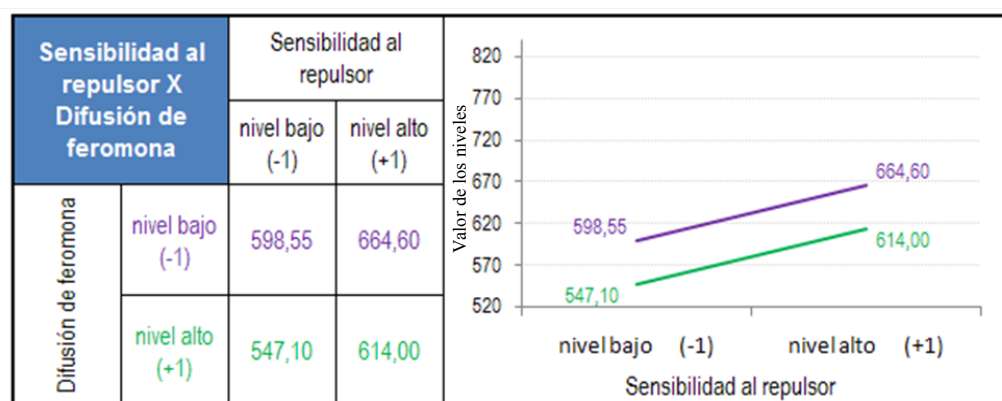


Figura 68. Influencia de la sensibilidad al repulsor sobre la difusión de feromona, en la estrategia de enjambre.

3. **Variable principal:** Sensibilidad al repulsor. **Variable condicionada:** Evaporación de feromona

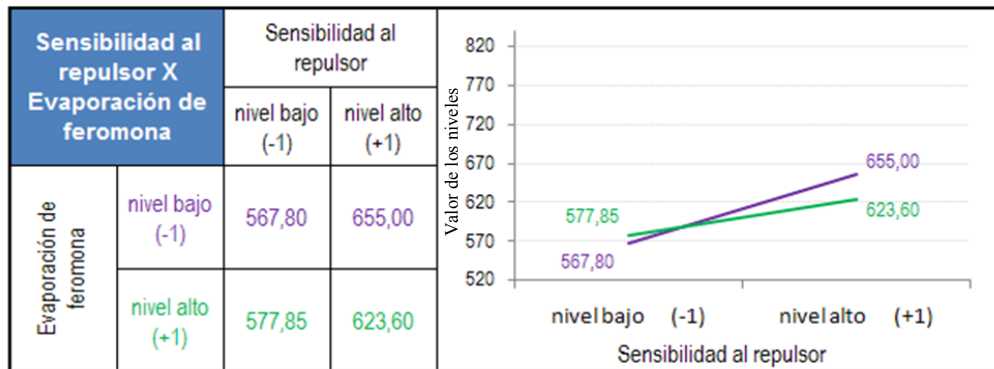


Figura 69. Influencia de la Sensibilidad al repulsor sobre la Evaporación de feromona, en la estrategia de enjambre.

Sensibilidad al repulsor produce un incremento en *Evaporación de feromona* al pasar del nivel bajo al nivel alto.

4. **Variable principal:** Sensibilidad al atractor. **Variable condicionada:** Difusión de feromona

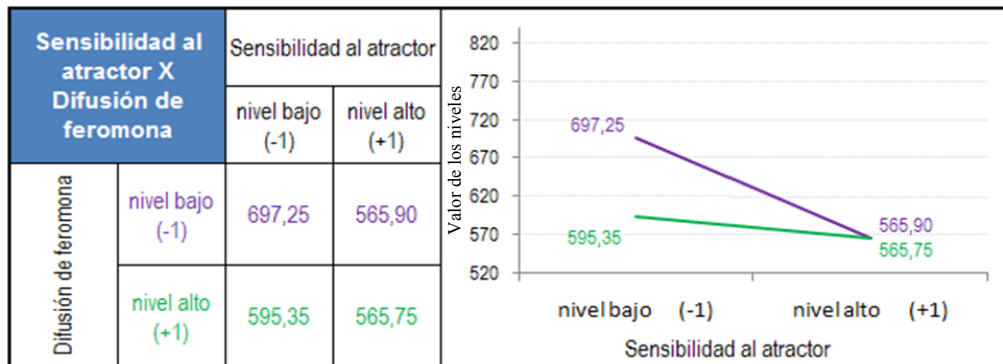


Figura 70. Influencia de Sensibilidad al atractor sobre Difusión de feromona, en la estrategia de enjambre.

Sensibilidad al atractor produce una disminución en *Difusión de feromona* al pasar del nivel bajo al nivel alto.

5. **Variable principal:** Sensibilidad al atractor. **Variable condicionada:** Evaporación de feromona

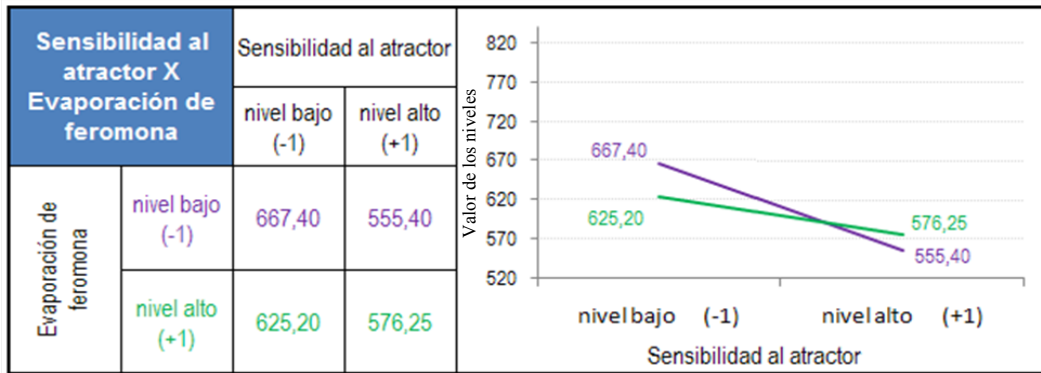


Figura 71. Influencia de Sensibilidad al atractor sobre la Evaporación de feromona, en la estrategia de enjambre.

Sensibilidad al atractor produce una disminución en Evaporación de feromona al pasar del nivel bajo al nivel alto.

6. **Variable principal:** Difusión de feromona. **Variable condicionada:** Evaporación de feromona

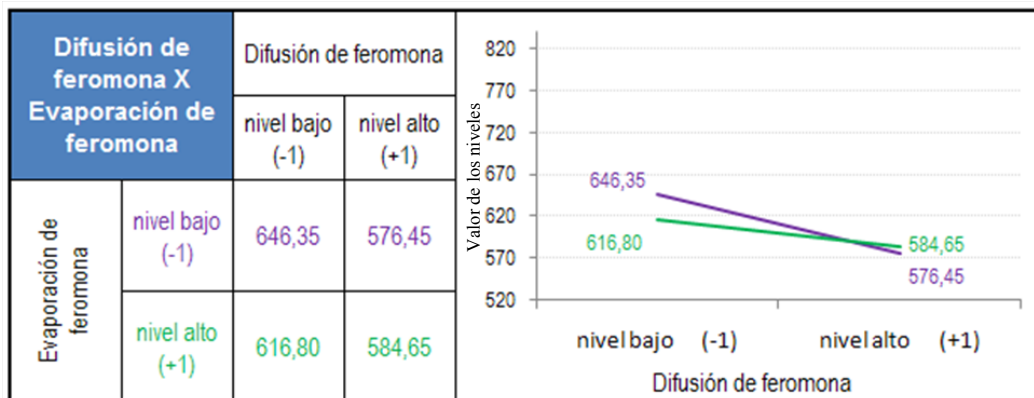


Figura 72. Influencia de la Difusión de feromona sobre la Evaporación de feromona, en la estrategia de enjambre.

Difusión de feromona produce una disminución en *Evaporación de feromona* al pasar del nivel bajo al nivel alto.

11.2.2 Análisis de las simulaciones correspondientes a la estrategia Multiagente:

Desempeño de la estrategia:

La Figura 73 muestra el desempeño de la estrategia con cada una de las 16 posibles configuraciones de los factores; en el eje de las abscisas se representan los 16 grupos de 5 repeticiones con la misma configuración de factores; en el eje de ordenadas se representa la diferencia en cantidad de pasos para llegar al objetivo, respecto al promedio total (que es de 2258,18 pasos); los valores negativos indican mayor cantidad de pasos que el promedio, los valores positivos indican menor cantidad de pasos que el promedio.

Puede verse que con la configuración de factores usada en las simulaciones del grupo 9, se obtiene un desempeño extremadamente pobre, con alrededor de 1400 pasos más que la cantidad promedio de pasos para llegar al objetivo.

La configuración que da mejores resultados es la correspondiente al grupo de simulaciones número 12, la cual logra un ahorro de casi 211 pasos respecto al promedio.

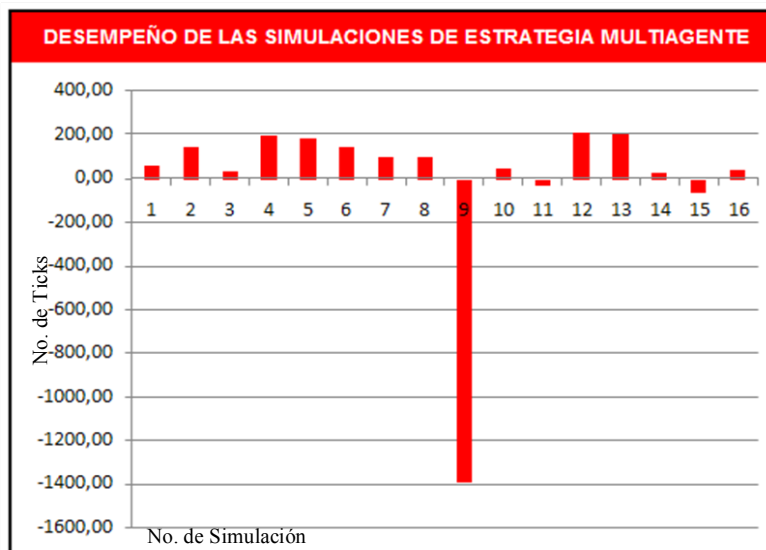


Figura 73. Desempeño de la estrategia multiagente en cada una de las 16 iteraciones. En 0,00 se encuentra el promedio. Los valores positivos corresponden a ahorro de tiempo (ticks) y los negativos.

Efectos de cada factor:

En la Figura 74 puede verse la influencia de cada factor, sobre el tiempo de llegada del sistema Multiagente al punto de objetivo; los valores indicados corresponden al aumento promedio (o disminución promedio) que ejerce cada factor sobre el tiempo promedio de llegada.

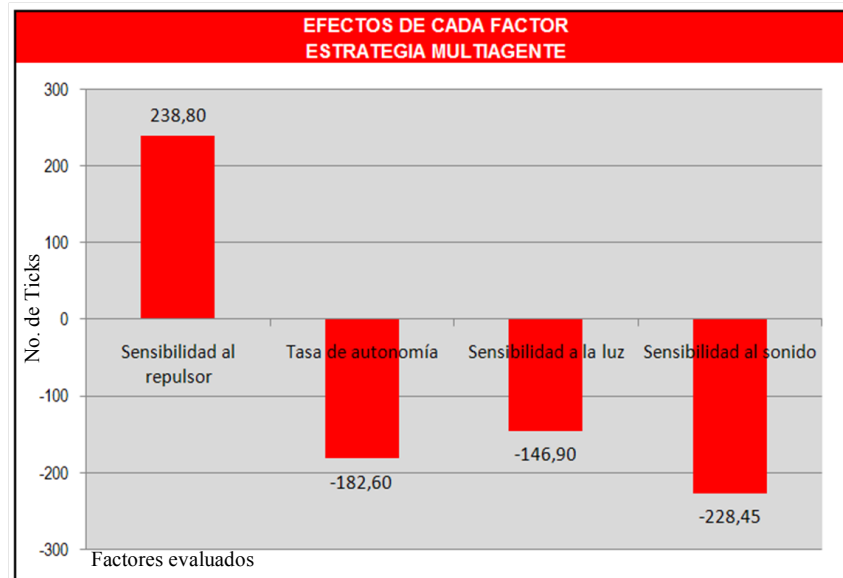


Figura 74. Efecto de los factores en el tiempo de llegada promedio en la estrategia multiagente. En 0 el valor promedio de llegada; valores positivos disminuyen el tiempo de llegada, mientras que los negativos lo aumentan.

El factor más influyente es *Sensibilidad al repulsor*, un aumento en este factor, del nivel bajo al nivel alto, aumenta considerablemente el tiempo de llegada (Figura 75a).

El factor *Sensibilidad al sonido* es el segundo factor más influyente, un aumento en este factor, del nivel bajo al nivel alto, disminuye el tiempo de llegada (Figura 75d)., en mayor medida que los factores *Tasa de autonomía* y *Sensibilidad a la luz*.

El factor menos influyente es *Sensibilidad a la luz*, sin embargo un cambio en este factor disminuye el tiempo de llegada (Figura 75c), en apenas menor medida que los factores *Tasa de autonomía* y *Sensibilidad al sonido*.

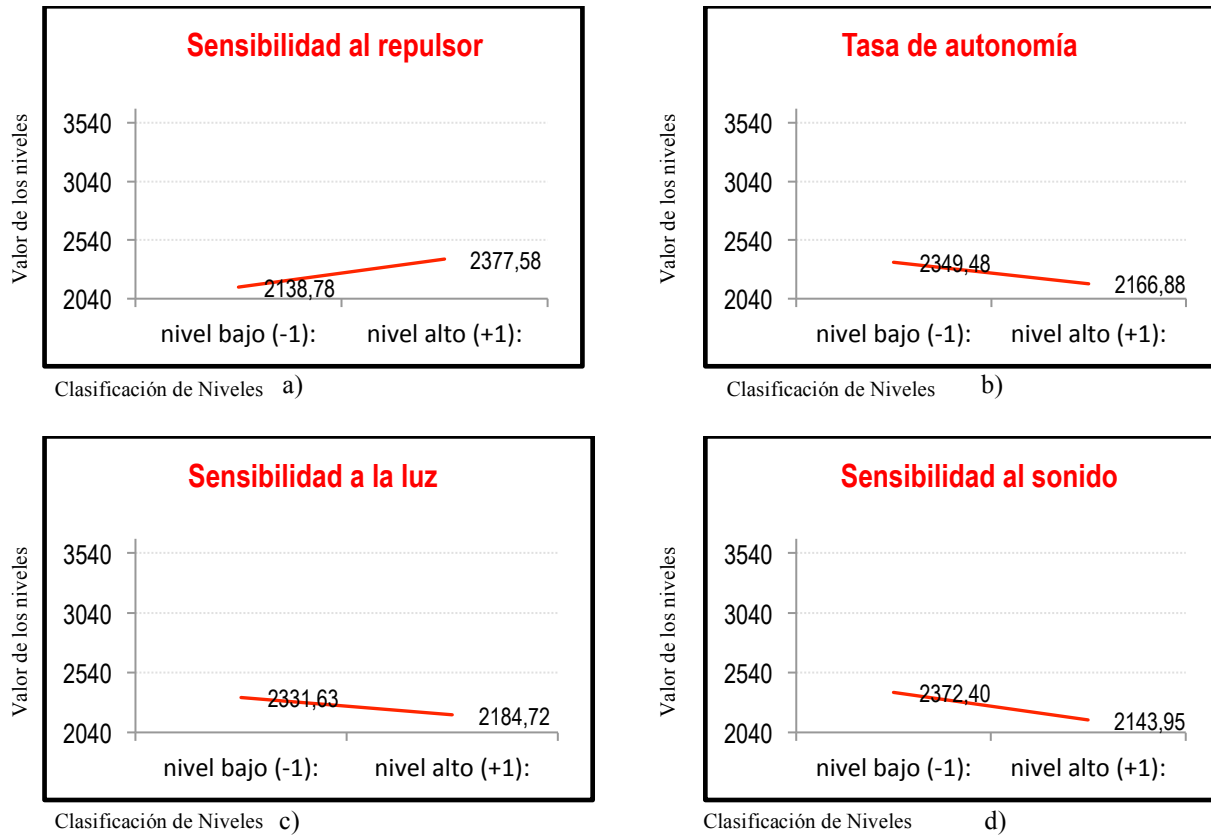


Figura 75. Efectos independientes de cada factor relacionado con la estrategia Multiagente. a) Sensibilidad al repulsor. b) Tasa de autonomía. c) Sensibilidad a la luz. d) Sensibilidad al sonido.

Interacciones:

Al igual que con la estrategia de enjambre, se evaluó el efecto que en la estrategia multiagente tenía un factor sobre otro.

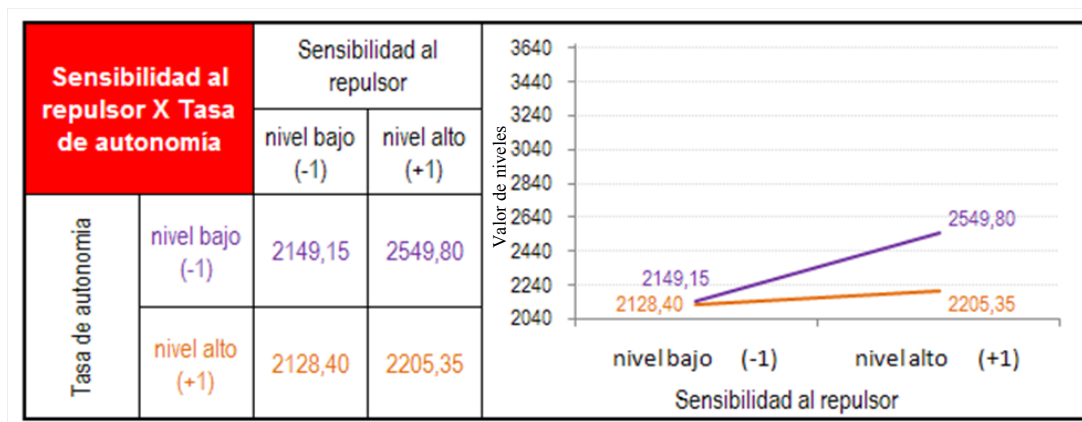


Figura 76. Influencia de Sensibilidad al repulsor en la Tasa de autonomía, en la estrategia multiagente.

1. **Variable principal:** Sensibilidad al repulsor. **Variable condicionada:** Tasa de autonomía

En la Figura 76 se observa que la *Tasa de autonomía* no cambia mucho a medida que la *sensibilidad al repulsor* aumenta, pues este cambio es contrarrestado por la interacción entre ambos factores, resultando una acción casi nula sobre *Tasa de autonomía*.

2. **Variable principal:** Sensibilidad al repulsor. **Variable condicionada:** Sensibilidad a la luz

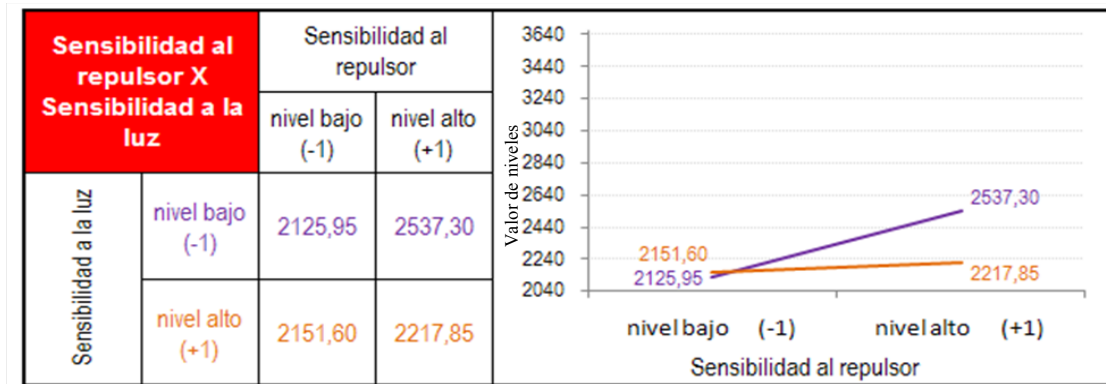


Figura 77. Influencia de Sensibilidad al repulsor en la Sensibilidad a la luz, en la estrategia multiagente.

Con un nivel bajo en *Sensibilidad al repulsor* un cambio de un nivel bajo a un nivel alto no se aumentan significativamente los valores de *Sensibilidad a la luz*, pues este cambio de niveles es contrarrestado por la interacción de los factores.

3. **Variable principal:** Sensibilidad al repulsor. **Variable condicionada:** Sensibilidad al sonido

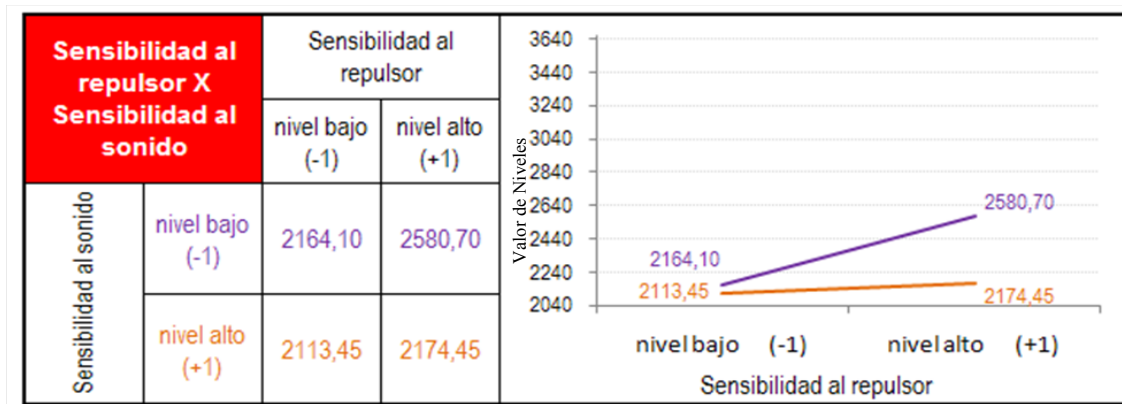


Figura 78. Influencia de Sensibilidad al repulsor sobre Sensibilidad al sonido, en la estrategia multiagente.

Un cambio de un nivel bajo a un nivel alto en *Sensibilidad al repulsor* casi no produce efecto en *Sensibilidad al sonido*, hace aumentar muy levemente sus valor, pues este cambio en la variable independiente es contrarrestado por la interacción, resultando una acción casi nula sobre *Sensibilidad al sonido*.

4. **Variable principal:** Tasa de autonomía. **Variable condicionada:** Sensibilidad a la luz

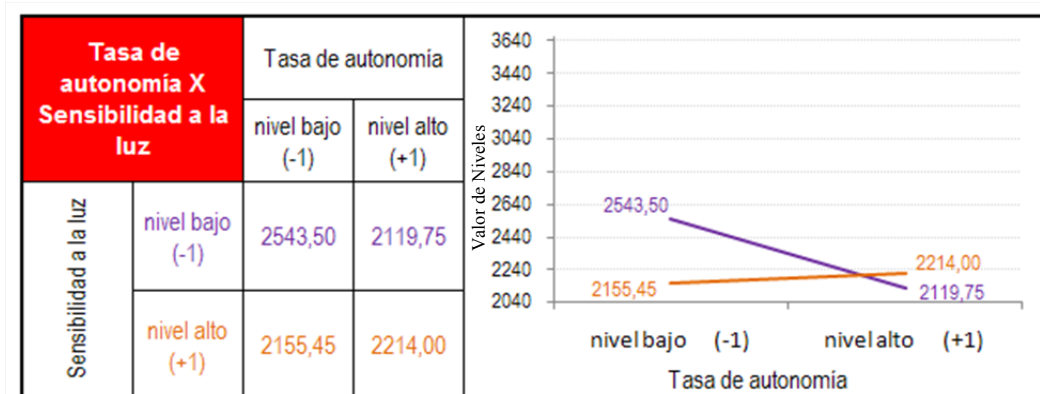


Figura 79. Influencia de *Tasa de autonomía* sobre *Sensibilidad a la luz* menor que la interacción.

Se observa que el comportamiento de *Sensibilidad a la luz* es inversamente proporcional a *Tasa de autonomía*. Mientras *Tasa de autonomía* disminuye, sensibilidad a la luz aumenta muy levemente sus valores. Este cambio en *Tasa de autonomía* es contrarrestado por la interacción, resultando una acción casi nula sobre *Sensibilidad a la luz*.

5. **Variable principal:** Tasa de autonomía. **Variable condicionada:** Sensibilidad al sonido

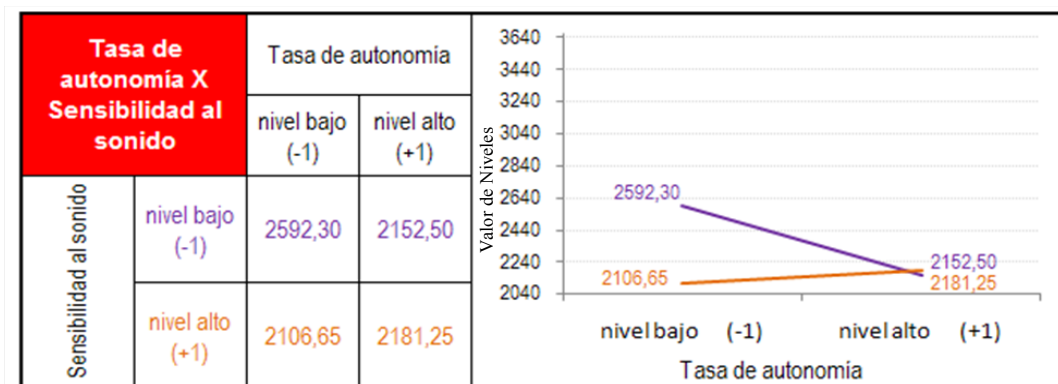


Figura 80. Influencia de *Tasa de autonomía* sobre *sensibilidad al sonido*, en la estrategia multiagente.

Se observa que el comportamiento de *Sensibilidad al sonido* es inversamente proporcional a *Tasa de autonomía*. Mientras *Tasa de autonomía* disminuye, *Sensibilidad al sonido* aumenta muy levemente sus valores. El cambio en *Tasa de autonomía* es contrarrestado por la interacción, resultando una acción casi nula sobre *Sensibilidad al sonido*.

6. *Variable principal:* Sensibilidad a la luz. *Variable condicionada:* Sensibilidad al sonido

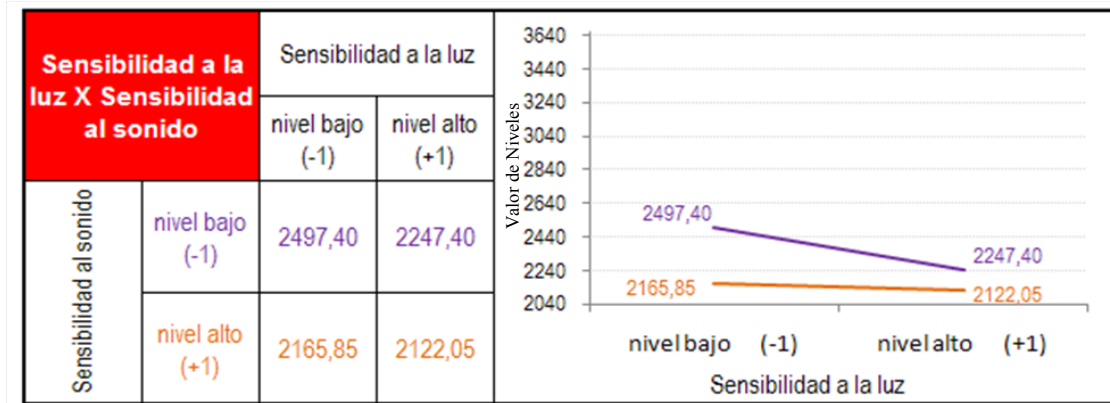


Figura 81. Influencia de Sensibilidad a la luz sobre Sensibilidad al sonido, en la estrategia multiagente.

El nivel de *Sensibilidad al sonido* cambia muy levemente de un nivel alto a bajo, cuando *Sensibilidad a la luz* cambia significativamente de nivel alto a bajo. El cambio en *Sensibilidad a la luz* es contrarrestado por la interacción, resultando una acción casi nula sobre *Sensibilidad al sonido*. Como las rectas no se tocan, la interacción entre las variables no tiene un efecto significativo en *Sensibilidad al sonido*.

11.2.3 Prueba de Normalidad

Se utiliza el test de Kolmogorov - Smirnov y se plantean las hipótesis de la siguiente manera:

- **H₀: Hipótesis nula.** Los resultados obtenidos para el algoritmo elegido presentan una distribución normal.
- **H₁: Hipótesis alternativa.** Los resultados obtenidos para el algoritmo elegido no presentan una distribución normal.

Se toma un nivel de significancia de 0.05, correspondiente a un error del 5%, en las pruebas de normalidad, homocedasticidad y comparación de los grupos de datos, para aceptar la hipótesis nula. El caso que se consideró para el análisis de las estrategias corresponde al óptimo (costo promedio menor, por iteración). Los datos se relacionan en la Tabla 4.

Con un nivel α de significancia de 0,05 y un tamaño de muestra de 5, según la tabla de Kolmogorov – Smirnov, el valor crítico es de 0,565. Al realizar la prueba con los costos de la Tabla 4, el estimador p se encuentra por debajo del valor crítico, siendo de 0,0266; por lo tanto se comprueba la normalidad y no se rechaza la Hipótesis nula.

Tabla 4:

Factores óptimos obtenidos mediante simulación para la estrategia de enjambre

Ejecución	Sensibilidad al repulsor	Sensibilidad al atractor	Difusión de feromona	Evaporación de feromona	Costo	Costo Promedio
3	50	20	20	5	396	528,60
	50	20	20	5	596	
	50	20	20	5	475	
	50	20	20	5	545	
	50	20	20	5	631	

Nota: La ejecución en donde se obtuvo el menor costo fue la número 3. Se muestran las diferentes configuraciones de factores iniciales y el costo de cada una de ellas, al igual que el costo promedio de la ejecución.

Se realiza el mismo procedimiento y se obtuvieron los datos de la Tabla 5, correspondientes a la estrategia Multiagente.

Tabla 5:

Factores óptimos obtenidos mediante simulación para la estrategia multiagente

Ejecución	Sensibilidad al repulsor	Tasa de autonomía	Sensibilidad a la luz	Sensibilidad al sonido	Costo	Promedios
12	75	10	40	30	2062	2047,20
	75	10	40	30	2103	
	75	10	40	30	2035	
	75	10	40	30	2025	
	75	10	40	30	2011	

Nota: La ejecución en donde se obtuvo el menor costo fue la número 12. Se muestran las diferentes configuraciones de factores iniciales y el costo de cada una de ellas, al igual que el costo promedio de la ejecución.

Al aplicar la prueba de Kolmogorov-Smirnov se obtiene un estimador p de 0,04459, por lo tanto se comprueba la normalidad y no se rechaza la Hipótesis nula.

A continuación se procede a realizar la prueba de homocedasticidad (igualdad de varianzas).

11.2.4 Prueba de Homocedasticidad

Se utiliza el test de Levene y se plantean las hipótesis de la siguiente manera:

- **H₀: Hipótesis nula.** Los resultados obtenidos para los dos grupos de estrategias presentan valores de varianza iguales.
- **H₁: Hipótesis alternativa.** Los resultados obtenidos para los dos grupos de estrategias no presentan valores de varianza iguales.

Se procede a aplicar la prueba de Levene a los datos obtenidos en la simulación del caso óptimo para cada algoritmo, presentados en la Tabla 6:

Tabla 6:

Costo mínimo para las estrategias de enjambre y multiagente.

Ejecución 3	Ejecución 12
Enjambre	Multiagente
396	2062,00
596	2103,00
475	2035,00
545	2025,00
631	2011,00

Nota: Se muestra el costo mínimo de cada una de las 5 iteraciones que se hicieron por ejecución.

Se obtiene un valor del estimador p de 0,130, el cual determina que no se rechaza la hipótesis nula, pues las varianzas son iguales. Esto indica que se cumple con el criterio de Homocedasticidad. El valor obtenido del estadístico F es de 6,77 lo que significa que las varianzas no difieren significativamente (a mayor valor de F , mayor diferencia entre las varianzas).

A continuación se analizan los datos mediante la prueba ANOVA.

11.2.5 Prueba de Medias

Se estudió el efecto de un único factor (el tipo de algoritmo) con dos variantes (algoritmo de enjambre y algoritmo multiagente), sobre la media del número de ticks empleado, que debe ser el menor posible.

Se utiliza el test de ANOVA de un solo factor y se plantean las hipótesis de la siguiente manera:

- **H₀: Hipótesis nula.** Los resultados obtenidos para los dos grupos de algoritmos presentan valores de media iguales.
- **H₁: Hipótesis alternativa.** Los resultados obtenidos para los dos grupos de algoritmos no presentan valores media iguales.

Esta prueba se aplica los datos obtenidos en la Tabla 6, obteniendo un valor del estimador p de 0,0 y un estadístico $F = -5,145$.

Tabla 7:

Estimador p del test de ANOVA para ambas estrategias.

ANOVA de un factor	Estadístico F	p
<i>Caso óptimo: Costo vs. Estrategia</i>	1123,90	0

Nota: Se utilizó el test ANOVA de un factor, para la evaluación de la eficiencia de los algoritmos utilizados.

Se observa en la Tabla 7 que el valor de F es grande, (factor de cuadrados medios MS entre grupos es mucho mayor que el factor de cuadrados medios dentro del grupo) así que existe un efecto real y muy marcado sobre la influencia del tipo de algoritmo en la variable costo. (Para un análisis más detallado ver Anexo 5. Pruebas estadísticas).

Al ser el valor de p menor que el valor de significancia de 0,005, el test nos indica que se rechaza la hipótesis nula, pues las medias son diferentes.

11.2.6 Desempeño de los algoritmos

Se realiza un análisis del desempeño de los algoritmos, en términos de efectividad, eficiencia y consistencia.

11.2.6.1 Efectividad

Se estableció como criterio en las simulaciones que la medida de costo mínimo de los ticks se establecía en el momento en el que el 80% de los agente cumpliera su objetivo. Por lo tanto, como se analizan dos estrategias colaborativas, se asume que la efectividad de ambos algoritmos es del 80%.

11.2.6.2 Eficiencia

Para la comparación de eficiencia, en términos de mínimo número de ticks para cada algoritmo,

se realizó el análisis del caso de valores óptimos.

Caso 1. Valores óptimos: Enjambre vs. Multiagente

En los datos obtenidos del test de ANOVA correspondiente a los datos de la Tabla 6, se obtienen los valores para $F= 1123,907$ y $F_{crit}= 5,3176$ (valor crítico de F); como $F \gg F_{crit}$, los cambios entre los factores en las simulaciones tuvieron un efecto estadísticamente significativo sobre los resultados de éstas, es decir, la diferencia en el criterio (tipo de estrategia: enjambre y multiagente) influye de manera significativa en el resultado (costo mínimo: ticks).

Al comparar el costo mínimo, es decir, el menor número de ticks para cada algoritmo, estamos calculando su rapidez de convergencia, por lo tanto, determinamos directamente el desempeño del algoritmo en términos de eficiencia. Se realizó un diagrama de Caja y Bigotes, que se muestra en la Figura 82 para comparar la eficiencia del algoritmo de enjambre y multiagente. Se comprueba que el algoritmo de enjambre es el más eficiente, pues presenta el menor número de ticks para su convergencia, para una efectividad igual a la correspondiente al algoritmo multiagente.

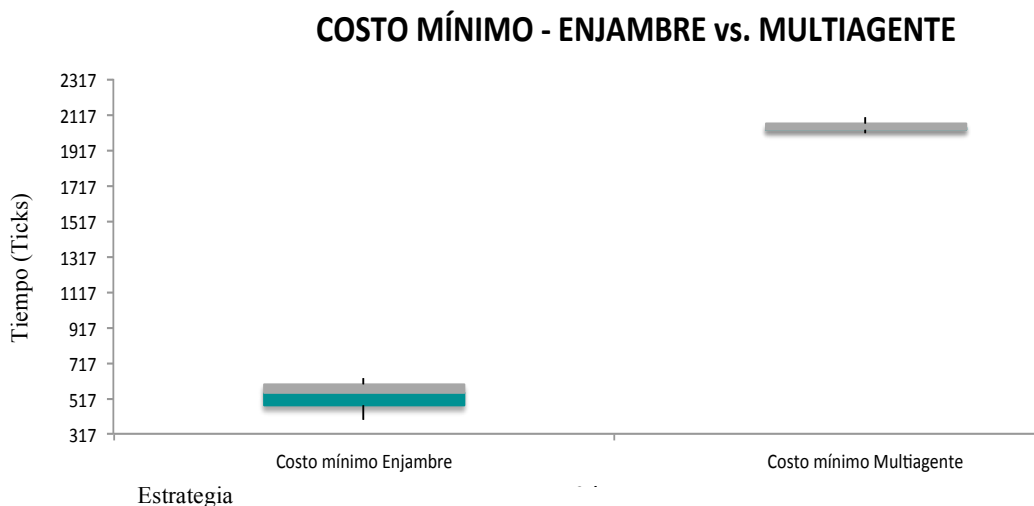


Figura 82. Diagrama de Caja y Bigotes para los algoritmos de Enjambre y Multiagente simulados. El eje x corresponde a la estrategia utilizada. El eje y corresponde al número de ticks.

Se observa que la medias para cada algoritmo (línea negra dentro de las cajas) se encuentran bastante alejadas entre sí, siendo mucho menor el valor de ticks correspondiente al algoritmo de enjambre y mucho mayor el correspondiente al algoritmo multiagente, estableciendo *la eficiencia del algoritmo de Enjambre sobre el algoritmo Multiagente*.

La configuración óptima del algoritmo Multiagente obtenida mediante la simulación se observa en la Tabla 8.

Tabla 8:

Configuración de factores para el caso 1: estrategia Multiagente

Sensibilidad al repulsor	Tasa de autonomía	Sensibilidad a la luz	Sensibilidad al sonido	Costo
75	10	40	30	2011

Nota: Corresponde al costo mínimo para la estrategia multiagente.

De los datos observados en Tabla 8, se puede concluir, que una configuración eficiente para el algoritmo supone una sensibilidad al repulsor de 75% (nivel alto), es decir, es mas probable que el agente lo ignore. La tasa de autonomía corresponde a un nivel bajo de 10%, lo que significa que la solución funciona eficientemente cuando la cantidad de robots autónomos (líderes) es menor. En cuanto a la sensibilidad al sonido, el valor óptimo es de 30 %, lo que corresponde a un nivel alto; este valor es el primer acercamiento que el agente tiene del objetivo, por lo tanto es deseable que ese valor de percepción sea alto. Por último se tiene la Sensibilidad a la luz, la cual corresponde a un nivel alto de 40% y de igual forma que el sonido, entre mejor percepción tenga el agente de su objetivo final, el costo de la solución se optimiza.

Tabla 9:

Configuración de factores para el caso 1: estrategia de Enjambre

Sensibilidad al repulsor	Sensibilidad al atractor	Difusión de feromona	Evaporación de feromona	Costo
50	20	20	5	396

Nota: Corresponde al costo mínimo para la estrategia de enjambre.

De la Tabla 9, se puede concluir, que una configuración eficiente para el algoritmo supone una sensibilidad al repulsor de 50%, es decir, es igual de probable que el lo ignore o que le atraiga (el agente se tropiece), este corresponde al nivel mínimo. La sensibilidad al atractor tiene como valor óptimo un 20%, que corresponde al nivel mínimo y la tasa de difusión de la feromona tiene un valor óptimo de 20%, correspondiente al nivel bajo. Una tasa de difusión muy alta implica que la feromona tendrá mayor alcance en el entorno, lo cual puede formar falsos atractores y por lo tanto, a que los agentes presenten un retraso en alcanzar su objetivo.

La evaporación de la feromona se encuentra en un nivel mínimo de 5%, lo que significa que a medida que el agente la deposita, esta se queda una cantidad de tiempo muy corto en el camino elegido, lo que permite que si el camino elegido es largo, otra hormiga pueda “reprogramarlo”, elegir uno nuevo y optimizar la solución; este valor es particularmente sensible al número de agentes, pues si existen muy pocos, la probabilidad de que la feromona se evapore y que ningún otro agente la perciba es alta, así que si existe una gran cantidad de agentes, la solución funciona de manera óptima, como en este caso.

11.2.6.3 Consistencia

Para la comparación de eficiencia, en términos de mínimo número de ticks para cada algoritmo, se realizó un test de ANOVA consistente en el análisis de los siguientes casos:

Caso 2. Enjambre - Peor caso: Se obtienen los valores de configuración de los factores para el peor de los casos (mayor costo) para la estrategia de enjambre, los cuales se muestran en la Tabla 10.

Tabla 10:

Configuración de Factores para el caso 2: estrategia de Enjambre

Iteración	Sensibilidad al repulsor	Sensibilidad al atractor	Difusión de feromona	Evaporación de feromona	Costo	Costo Promedio
9	75	20	10	5	855	834,20
	75	20	10	5	689	
	75	20	10	5	792	
	75	20	10	5	986	
	75	20	10	5	849	

Nota: Corresponde al costo máximo. Los valores se obtienen como resultado de las simulaciones. La ejecución con mayor costo fue la número 9.

Caso 2. Multiagente - Peor caso: Se obtienen los valores de configuración de los factores para el peor de los casos (mayor costo) para la estrategia de enjambre, los cuales se muestran en la Tabla 11.

Tabla 11:

Configuración de Factores para el caso 2: estrategia Multiagente

Iteración	Sensibilidad al repulsor	Tasa de autonomía	Sensibilidad a la luz	Sensibilidad al sonido	Costo	Costo Promedio
9	75	10	20	5	2040	3656,40
	75	10	20	5	6637	
	75	10	20	5	2199	
	75	10	20	5	2015	
	75	10	20	5	5391	

Nota: Corresponde al costo máximo. Los valores se obtienen como resultado de las simulaciones. La ejecución con mayor costo fue la número 9.

Antes de realizar el Test de ANOVA, se comprobó la normalidad de los datos con el test de Kolmogorov – Smirnov y el la igualdad de varianzas (homcedasticidad) con el Test de Levene, a las configuraciones descritas en la Tabla 12, obteniendo resultados con un valor de significancia $\alpha = 0,05$.

Tabla 12:

Datos obtenidos del Test de ANOVA para las diferentes configuraciones, para evaluar Consistencia.

No.	Configuraciones realizadas	F	Fcrit	p
1.	ENJAMBRE Caso 1 (óptimo) vs. Caso 2 (peor caso)	22,692	5,3176	0,001
2.	MULTIAGENTE Caso 1 (óptimo) vs. Caso 2 (peor caso)	2,6794	5,3176	0,140

Nota: Datos obtenidos con el Test de ANOVA de un solo factor y con $\alpha = 0,05$.

Configuración 1: ENJAMBRE Caso 1 (óptimo) vs. Caso 2 (peor caso)

Se obtiene un valor de $p < 0,05$ (significancia), por lo tanto la hipótesis nula se rechaza (las medias son diferentes). También se observa que $F \gg F_{crit}$, así que el factor de elección (tipo de caso: óptimo y peor) tiene un peso demasiado significativo en el desempeño del algoritmo, es decir, la configuración de los factores de sensibilidad al repulsor, al atractor, la tasa de difusión de feromona y la evaporación de feromona se deben elegir adecuadamente, pues cualquier variación en los factores afecta significativamente el valor de costo mínimo (número de ticks), *por lo tanto el algoritmo no es el mas consistente en encontrar el costo mínimo, pero si el más eficiente.*

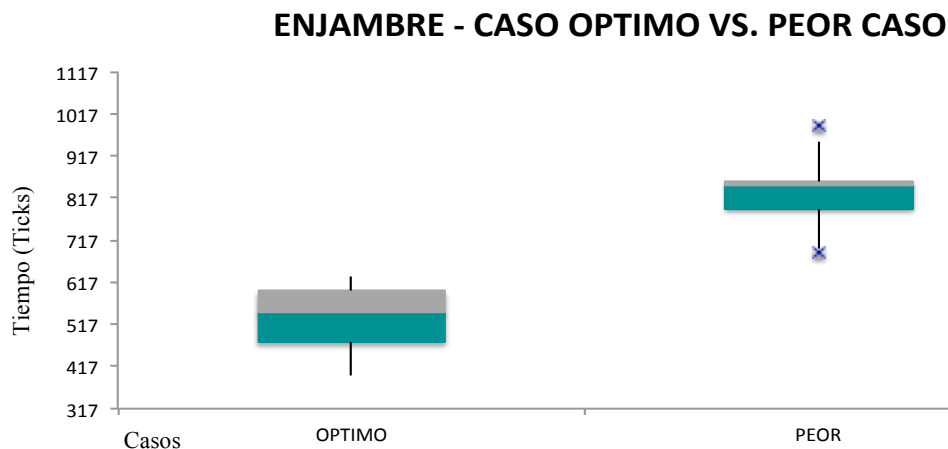


Figura 83. Diagrama de Caja y Bigotes para el algoritmo de Enjambre, configuración 1. El eje x corresponde al tipo de casos. El eje y corresponde al número de ticks.

Se observa en la Figura 83 que las medias de cada caso se encuentran muy alejadas entre sí, estableciendo que la configuración de los factores que dan como resultado el caso óptimo es la adecuada para obtener un costo mínimo.

Configuración 2: MULTIAGENTE Caso 1 (óptimo) vs. Caso 2 (peor caso)

Como se puede observar de la Tabla 12, el valor de $p > 0,05$ (significancia), por lo tanto la hipótesis nula no se rechaza (las medias son iguales). También se observa que $F < F_{crit}$, así que el factor de elección (tipo de caso: óptimo y peor) no tiene un peso demasiado significativo en el desempeño del algoritmo, es decir, al variar las diferentes configuraciones de los factores de sensibilidad al repulsor, al sonido, a la luz y la tasa de autonomía, no afectan significativamente el valor de costo mínimo (número de ticks).

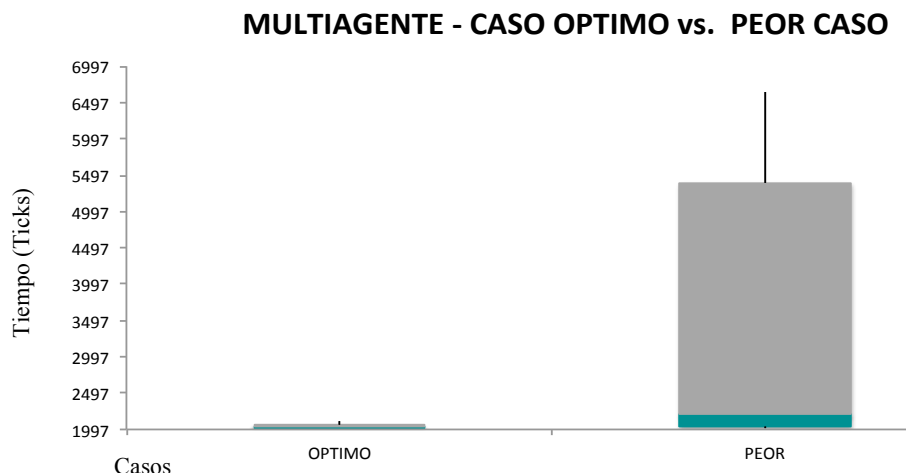


Figura 84. Diagrama de Caja y Bigotes para el algoritmo Multiagente, configuración 2. El eje x corresponde a los casos. El eje y corresponde al número de ticks.

Se observa en la *figura 64*, correspondiente al diagrama de Caja y bigotes, que las medias de cada caso se encuentran casi en el mismo valor, estableciendo que la configuración de los factores que dan como resultado el mejor de los casos no afecta de manera significativa el desempeño del algoritmo, *por lo tanto el algoritmo de enjambre es el mas consistente en encontrar el costo, pero no el costo mínimo, por lo tanto, no es el mas eficiente.*

11.2.6.4 Resumen del desempeño

Se presenta en la Tabla 13 el resumen desempeño de los algoritmos de Enjambre y Multiagente según las pruebas y la metodología aplicada.

Tabla 13:

Comparación del desempeño de las estrategias de Enjambre y Multiagente

Algoritmo	Desempeño		
	Efectivo	Eficiente	Consistente
Enjambre	X	X	
Multiagente	X		X

12 CONCLUSIONES Y RECOMENDACIONES

Más allá de lo que las pruebas estadísticas aplicadas demuestran, según lo observado en las simulaciones, se puede deducir que la estrategia Enjambre presenta mejor desempeño, no solo por la velocidad con la que el sistema encuentra el objetivo, sino también por el menor uso de recursos comparada con la estrategia de Multiagente.

La selección de las estrategias utilizadas se debe a la amplia utilización de estas (o técnicas similares) en el ámbito de la investigación con sistemas de robótica móvil colectiva. El uso de metaheurísticas de enjambre como ACO está ampliamente difundido no solo en investigaciones con robots sino en temas diversos que van desde las redes de comunicaciones hasta el estudio de los sistemas sociales y de comportamiento grupal. Aunque esto hace relativamente fácil encontrar documentación al respecto, presenta la dificultad de tener que elegir entre tanta documentación, la información pertinente al objeto de estudio.

Por otra parte, las estrategias de sistemas Multiagente son difíciles de hallar, pues en el marco de la robótica móvil, al tocar el tema de agentes casi toda la documentación disponible redirecciona a los sistemas de enjambre.

Aunque se logra determinar una trayectoria desde el punto de partida hacia el objetivo, en ocasiones algunos robots quedan atascados entre los obstáculos, impidiendo que puedan desplazarse hasta el objetivo todos los componentes del sistema, ya sea enjambre o multiagente, presentándose esta situación con mayor frecuencia al aplicar la estrategia de enjambre.

La distribución de feromonas modifica la forma en que las hormigas perciben el entorno, creando una serie de campos potenciales atractivos para las hormigas, creando “falsos atractores”, puntos o trayectorias cíclicas en los que los robots permanecen atascados; cuando esto sucede el algoritmo de enjambre no presenta la eficiencia deseada.

El algoritmo Multiagente resultó ser el más consistente, pues las variaciones de los factores correspondientes no afectaron considerablemente el tiempo (ticks) de convergencia de dicho algoritmo.

Las “pequeñas” diferencias en las variables iniciales para el algoritmo de enjambre mostraron que daban como resultado tiempos de convergencia muy diferentes entre sí; mediante las pruebas se estableció la poca consistencia de este tipo de algoritmos, sensibles a las mínimas variaciones en sus parámetros de entrada.

Es muy importante establecer y hacer pruebas para hallar los valores óptimos del algoritmo de enjambre, que dará como resultado un tiempo de convergencia promedio regular y óptimo para la tarea específica que se desea llevar a cabo.

Es interesante señalar que el factor que mas influencia tuvo en el tiempo de llegada en la estrategia de enjambre fue la sensibilidad al atractor, seguida por sensibilidad al repulsor, difusión de la feromona y por último la evaporación de la feromona. Esto puede significar que es es mas importante para el algoritmo detectar los atractores y los objetivos que evadir los obstáculos.

Analizando las interacciones entre variables en la estrategia de enjambre, se encontró una relación entre la difusión de la feromona y su evaporación; al aumentar la difusión, disminuye la evaporación de la feromona; esto tiene sentido, pues si el radio de difusión es mas grande, el tiempo de evaporación disminuye.

El factor que mas influencia tuvo en el tiempo de llegada en la estrategia multiagente fue la sensibilidad al repulsor, seguida por la tasa sensibilidad al sonido, tasa de autonomía y por último, la sensibilidad a la luz; esto podría significar que es muy importante establecer sensorialmente en un sistema robótico donde se encuentran los obstáculos.

Analizando las interacciones entre variables en la estrategia multiagente, se encontró una relación entre tasa de autonomía y sensibilidad a la luz, lo que puede concluir que si la tasa de autonomía es baja, el agente no puede dirigirse efectivamente hacia el atractor, que en este caso es la luz.

Se espera que el presente trabajo sirva como referencia para trabajos futuros en el campo de los sistemas de múltiples robots móviles, y trascienda de la experimentación en software para dar paso a los robots en hardware; con este propósito, se presentan los Anexos 1 y 2, en los que se propone una arquitectura inspirada en un sistema nerviosos biológico (Anexo 1), y una sencilla implementación de la misma en hardware (Anexo 2).

13 REFERENCIAS

- Acosta, G. A. (2010). Ambiente multi-agente robótico para la navegación colaborativa en escenarios estructurados/Multi-agent robotic environment for collaborative navigation in structured scenarios (Doctoral dissertation, Universidad Nacional de Colombia).
- Akao, Y. (1994). Development history of quality function deployment. *The Customer Driven Approach to Quality Planning and Deployment*, 339.
- Aldana, E. J. (2009). Robots Móviles Colaborativos: Un Caso de Estudio Basado en Cooperación Eusocial. *Universidad Nacional Autónoma De México*, 2009.
- Angulo, C., Ponsa, P., & Raya, C. (2006). Construcción modular de robots móviles. Proyecto basado en portafolio para estudiantes de grado. *IEEE-RITA*,1(1), 19-26.
- Arkin, Ronald. (1990). Integrating Behavioural, Perceptual and World Knowledge in Reactive Navigation. *Robots and Autonomous Systems*, 6, 105-122.
- Arkin, Ronald. (2000). Behavior Based Robotics. *Lecture Notes in Computer Science*, 1790, 103–116.
- Asada, H., &Slotine, J. J. (1986). Robot analysis and control. John Wiley & Sons.
- Ayers, J. &Witting, J. (2007) Biomimetic approaches to the control of underwater walking machines. *Philosophical Transactions of the Royal Society A-Mathematical Physical and Engineering Sciences*, 365, 273-295.
- Ayers, J., Blidberg, D. R. & Massa, D. (1994). Biologically Based Underwater Robots. *Unmanned Systems*, 12, 30-30.
- Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM) part 2: State of the art. *Robotics and Automation Magazine*.
- Balch, T. & Arkin, R. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 27-52.
- Batra, S. W. T. (1966). Nests and social behavior of halictine bees of India (Hymenoptera: Halictidae). *Indian J. Entomol*, 28(3), 375.

- Beckers, R., Holland, O. E. & Deneubourg, J. L. (1994). From local actions to global tasks: Stigmergy and collective robotics. *In Artificial life IV*, 181, 189.
- Beer, R. D., Quinn, R. D., Chiel, H. J., & Ritzmann, R. E. (1997). Biologically inspired approaches to robotics: What can we learn from insects? *Communications of the ACM*, 40(3), 30-38.
- Belpaeme, T., & Birk, A. (2001). Hungry robots. *Crossroads*, 8(2), 15-19.
- Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?* (pp. 703-712). Springer Berlin Heidelberg.
- Bonabeau, E., & Théraulaz, G. (2000). Swarm smarts. *Scientific American*, 282(3), 54-61.
- Bonabeau, Eric. (2002). Predicting the unpredictable. *Harvard Business Review*. 80(3), 109-116.
- Boorer, M. (1980). Los Mamíferos. *Bruguera*.
- Borenstein, J. & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7 (3), 278–288.
- Braitenberg, V. (1986). *Vehicles: Experiments in synthetic psychology*. MIT Press.
- Brooks, R (1991). New approaches to robotics. *Science* 253(5025), 1227–1232.
- Brooks, R. A., Breazeal, C., Irie, R., Kemp, C. C., Marjanovic, M., Scassellati, B., & Williamson, M. M. (1998). Alternative essences of intelligence. *AAAI/IAAI*, 961-968.
- Brooks, R. A. (1986A). Achieving artificial intelligence through building robots. MIT Artificial Intelligence Laboratory. AI Memo 899.
- Brooks, R. A. (1986B). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 14-23.
- Brooks, R. A. (1987). Planning is just a way of avoiding figuring out what to do next. *Cambrian Intelligence the early history of the new AI*, 103–110.
- Brooks, R. A. (1990). Elephants don't play chess. *Robotics and autonomous systems*, 6(1), 3-15.
- Brooks, R. A. (1991A). Intelligence Without Representation, *Artificial Intelligence* 47(1), 139-159.

- Brooks, R. A. (1991B). Challenges for complete creature architectures. In Meyer, JA and SW Wilson (1991) From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior. MIT Press/Bradford Books. Cambridge Ma (pp. 434-443).
- Brooks, R. A. (1991C). How to build complete creatures rather than isolated cognitive simulators. *Architectures for intelligence*, 225-239.
- Cañas, J.M. (2003). Jerarquía dinámica de esquemas para la generación de comportamiento autónomo (tesis doctoral). Universidad Politécnica De Madrid, España.
- Cao, Y. U., Fukunaga, A. S., & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1), 7-27.
- Carthy, J. D. (1970). La conducta de los animales. *Biblioteca básica Salvat*.
- Cen, Y., Song, C., Xie, N., & Wang, L. (2008, June). Path planning method for mobile robot based on ant colony optimization algorithm. 3rd *IEEE Conference on Industrial Electronics and Applications, 2008. ICIEA 2008*, 298-301.
- Chaimowicz, L., Kumar, V. & Campos, M. (2004). A paradigm for dynamic coordination of multiple robots. *Autonomous Robots*, vol. 17, no 1, 2004, p. 7-21.
- Cham, J., Karpick, J., & Cutkosky, M. (2004). Stride period adaptation of a biomimetic running hexapod. *The International Journal of Robotics Research*, 23(2), 141-153.
- D'Angelo, A., Menegatti, E., & Pagello, E. (2007). How a cooperative behavior can emerge from a robot team. *In Distributed Autonomous Robotic Systems 6*(pp. 75-84). Springer Japón.
- D'Inverno, M., & Luck, M. (2007). Understanding Agent Systems. Springer-Verlag Berlin.
- Delcomyn, F. (2007). *Biologically inspired robots*. Advanced Robotic Systems International and I-Tech.
- Deneubourg, J. L., Aron, S., Goss, S., & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior*, 3(2), 159-168.
- Dias, M. B. (2004). *Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments* (Tesis doctoral). Carnegie Mellon University).
- Dillmann, R., Albiez, J., Gassmann, B., Kerscher, T. & Zoellner, M. (2007). Biologically inspired walking machines: design, control and perception. *Philosophical Transactions of the Royal Society A-Mathematical Physical and Engineering Sciences* 365, 133-151.

- Dorigo, M., Tuci, E., Gross, R., Trianni, V., Labella, T. H., Nouyan, S. & Gambardella, L. M. (2005). The swarm-bots project. In *Swarm Robotics* (pp. 31-44). Springer Berlin Heidelberg
- Dorigo, M. & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1), 53-66.
- Dorigo, M. & Sahin, E. (2004). *Swarm robotics*. Autonomous Robots.
- Dorigo, M. & Stützle, T. (2004) *Ant Colony Optimization*. USA. MIT Press.
- Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms* (tesis doctoral). Politecnico di Milano, Italia.
- Ducatelle, F., Di Caro, G. A., & Gambardella, L. M. (2010). Cooperative stigmergic navigation in a heterogeneous robotic swarm. *From Animals to Animats*, 11.
- El-Abd, M. (2010). A cooperative approach to the artificial bee colony algorithm. 2010 *IEEE Congress on in Evolutionary Computation (CEC)*, (pp. 1-5). IEEE.
- Espitia, H.E. & Sofrony, J. I. (2012). Algoritmo para Planeación de Trayectorias de Robots Móviles Empleando Campos Potenciales y Enjambres de Partículas Brownianas. *Ciencia e Ingeniería Neogranadina*, 22(2), 75 – 96.
- Floreano, D. & Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence*. USA. MIT Press.
- Fukuda, T. (1988). Self organizing robots based on cell structures-CEBOT. In Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS'88) (pp. 145-150).
- Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1), 3-31.
- Glover, F (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research* 13 (5): 533–549.
- Granosik, G., & Borenstein, J. (2005). Integrated joint actuator for serpentine robots. *Mechatronics, IEEE/ASME Transactions on*, 10(5), 473-481.
- Grassé P. P., (1959). La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes Natalensis* et *Cubitermes* sp. La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6(1), 41-80.
- Gutiérrez, H., & de la Vara, R. (2008). *Análisis y diseño de experimentos*. México: McGraw-Hill.

- Holland, J. M. (2004). *Designing autonomous mobile robots: inside the mind of an intelligent machine*. Newnes.
- Holland, O., & Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial life*, 5(2), 173-202.
- Ijspeert, A. J., Martinoli, A., Billard, A., & Gambardella, L. M. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2), 149-171.
- Iocchi, L., Nardi, D. & Cesta, A. (2003). Analyzing Features of Multi-Robot Systems An Initial Step for the RoboCare Project. *First RoboCare Workshop*, 71.
- Jaramillo Morales, M. F. (2010). *Planificador de trayectorias para un robot móvil orientado a la reconstrucción en 3D de entornos desconocidos* (tesis doctoral). Universidad Nacional de Colombia - Sede Manizales.
- Jiménez, J., Vallejo, M., & Ochoa, J. (2007). Metodología para el Análisis y Diseño de Sistemas Multi-Agente Robóticos: MAD-Smart. Segundo Congreso Colombiano de Computación.
- Kaminka, G. A., & Frenkel, I. (2005). *Flexible teamwork in behavior-based robots*. In *Proceedings Of The National Conference On Artificial Intelligence* (Vol. 20, No. 1, p. 108). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Kelly, I. D., Keating, D. A., & Warwick, K. (1997, June). Mutual learning by autonomous mobile robots. In *Proceedings of the 1st Workshop on Telepresence and Robotics Applications in Science and Art*, Linz, Austria (pp. 103-116).
- Kennedy, J. & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks IV*, 1942-1948.
- Kephart, J. O. (1994). A biologically inspired immune system for computers. *Proceedings of Artificial Life IV: The Fourth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press. 1994. pp. 130-139.
- Koenig, S., Likhachev, M. (2002). Improved fast replanning for robot navigation in unknown terrain. *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), 968-975.
- Koenig, S., Szymanski, B., & Liu, Y. (2001). Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, 31(1-4), 41-76.

- Koenig, S., & Liu, Y. (2001, May). Terrain coverage with ant robots: a simulation study. In *Proceedings of the fifth international conference on Autonomous agents* (pp. 600-607). ACM.
- Koren, Y., & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. *Proceedings IEEE International Conference on Robotics and Automation*. (pp. 1398-1404). IEEE.
- Kube, C. R., & Bonabeau, E. (2000). Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1), 85-101.
- Kube, C. R., & Zhang, H. (1992). Collective robotic intelligence. In *Second International Conference on Simulation of Adaptive Behavior* (pp. 460-468).
- Kumar V., Sahin F. and Cole E., “Ant Colony Optimization Based Swarms: Implementation for the Mine Detection Application”, in the *Proceedings of 2004 IEEE International Conference on Systems, Man & Cybernetics*, vol. 1, pp. 716 – 721, October 10 – 13, 2004.
- Labella, T., Dorigo, M., & Deneubourg, J. (2007). Self-organized task allocation in a group of robots. In *Distributed Autonomous Robotic Systems 6*(pp. 389-398). Springer.
- Liu, J. & Wu, J. (2001). *Multi-Agent Robotic Systems*. Boca Ratón, USA. CRC Press.
- López Pérez, L. H., Mata Herrera, E. & García Pérez, E. (2001). Estado del arte en robótica móvil autónoma distribuida. *Conciencia Tecnológica*, (017).
- Lozano-Pérez, T. (1990). *Foreword: Mobile Robots and Robotics*. *Autonomous Robot Vehicles*. Editores I.J. Cox y G.T. Wilfong. Springer-Verlag. pp vii-xi.
- Marsh, L., & Onof, C. (2008). Stigmergic epistemology, stigmergic cognition. *Cognitive Systems Research*, 9(1), 136-149.
- Martínez, F. J. (2003). Azar y caos. En: *Diccionario Crítico de Ciencias Sociales*, Publicación Electrónica, Universidad Complutense, Madrid. <http://ucm.es/info/eurotheo/diccionario>
- Martinoli, A., & Mondada, F. (1997). Collective and cooperative group behaviours: Biologically inspired experiments in robotics (pp. 1-10). Springer Berlin Heidelberg.
- Martinoli, A. (2001). Collective complexity out of individual simplicity. *Artificial Life*, 7(3), 315-319.
- Mataric, M. J. (2001). Learning in behavior-based multi-robot systems: policies, models, and other agents. *Journal of Cognitive Systems Research*, 2(1), 81-93

- Mataric, M. J. (2002) Sensory-Motor Primitives as a Basis for Imitation: Linking Perception to Action and Biology to Robotics. *Imitation in animals and artifacts*, 1–24.
- Matellan, V. (2012) *ABC2: Un Modelo Para el Control de Robots Autónomos* (tesis doctoral). Facultad de Informática de la Universidad Politécnica de Madrid.
- McFarland, D. & Bossert, T. (1993). *Intelligent behavior in animals and robots*. Mit Press.
- McFarland, D. (1994). Towards robot cooperation. *From animals to animats*, 3, 440-444.
- Melhuish, C., Sendova-Franks, A. B., Scholes, S., Horsfield, I., & Welsby, F. (2006). Ant-inspired sorting by robots: the importance of initial clustering. *Journal of the Royal Society Interface*, 3(7), 235-242.
- Miceli, J.E. et al. (2005). Teorías de la Complejidad y el Caos en Ciencias Sociales. Modelos Basados en Agentes y Sociedades Artificiales. *Actas del I Congreso latinoamericano de Antropología*. Universidad Nacional de Rosario, Argentina.
- Millán, J. D. R., & Torras, C. (1992). A reinforcement connectionist approach to robot path finding in non-maze-like environments. *Machine Learning*, 8(3-4), 363-395.
- Mondada, F., Pettinaro, G., Guignard, A., Kwee, I.W., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L.M., Dorigo, M. (2003). Swarm-bot: From concept to implementation. *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2003, Proceedings*.p. 1626-1631.
- Montgomery, D. C. (2005). *Diseño y análisis de experimentos*, 2ª edición, México Limusa Wiley.
- Muir, P. F. & Neuman, C. P. (1987). Kinematic modeling of wheeled mobile robots. *Journal of robotic systems*, 4(2), 281-340.
- Munirajan, V. K., Sahin, F., & Cole, E. (2004). Ant colony optimization based swarms: implementation for the mine detection application. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Vol. 1, pp. 716-721). IEEE
- Murphy, R. (2000). *An introduction to AI robotics*. Estados Unidos. The MIT Press.
- Newman, L. H. (1970). *El mundo de los insectos*. Salvat Editores S.A., 1970
- Nilsson. N.J., (1980). *Principles of Artificial Intelligence*. Paloalto, USA. Springer.
- Nilsson, N. J. (1984). *Shakey the robot*. Sri International Menlo Park, Estados Unidos.
- NIST National Institute of Standards and Technology & SEMATECH (2012). e-Handbook of Statistical Methods. Recuperado de: <http://www.itl.nist.gov/div898/handbook/index.htm>

- Ollero, A. (2006) *Robótica: manipuladores y robots móviles*. Barcelona, España. Marcombo.
- Ortega, J. G. (1994). *Navegación en robots móviles basada en técnicas de control predictivo neuronal* (tesis doctoral). Universidad de Sevilla, Departamento de ingeniería de sistemas y automática. España.
- Pagello, E., D'Angelo, A. & Ferrari, C. (2003). Emergent behaviors of a robot team performing cooperative tasks. *Advanced Robotics*, 17(1), 3-19.
- Polya, G. (2008). *How to solve it: A new aspect of mathematical method*. USA. Princeton University Press.
- Posadas, J. (2003). *Arquitectura para el Control de Robots Móviles mediante Delegación de Código y Agentes* (tesis doctoral). Universidad Politécnica de Valencia. España.
- Quijano, N. & Passino, K.M. (2007). Honey Bee Social Foraging Algorithms for Resource Allocation, Part I: Algorithm and Theory, . *Memorias del IEEE American Control Conference, ACC '07*, 3383-3388.
- Quijano, N. & Passino, K.M. (2007). Honey Bee Social Foraging Algorithms for Resource Allocation, Part II: Application. *Memorias de IEEE American Control Conference, ACC '07*, 3389-3394.
- Reynoso, C. (1998) *Teorías y Métodos de la Complejidad y el Caos: Una exploración antropológica*. Buenos Aires, Argentina. Editorial Sb.
- Ritzmann, R. E., Quinn, R. D., Watson, J. T. & Zill, S. N. (2000). Insect walking and biorobotics: a relationship with mutual benefits. *Bioscience*, 50, 23-33.
- Russell, S. Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Paloalto, California. Norvig, Publisher.
- Russell, A., Thiel, D., & Mackay-Sim, A. (1994). Sensing odor trails for mobile robot navigation. *In Robotics and Automation, Proceedings, IEEE International Conference on*, 2672-2677.
- Rutishauser S., Correll N., Martinoli A., (2009). Collaborative coverage using a swarm of networked miniature robots. *Robotics and Autonomous Systems*, 57(5), 517–525.
- Rybski, P. E., Larson, A., Veeraraghavan, H., LaPoint, M.A. &Gini, M. (2008). Performance evaluation of a multi-robot search & retrieval system: Experiences with MinDART. *Journal of Intelligent and Robotic Systems*, 53(3-4), 363-387.
- Saffari M. H. & Mahjoob, M. J. (2009). Bee colony algorithm for real-time optimal path planning of mobile robots. *Fifth International Conference on Soft Computing. Computing with Words and Perceptions in System Analysis Decision and Control*, 1–4.

- Saffiotti, A., Broxvall, M. (2005). *PEIS Ecologies: Ambient intelligence meets autonomous robotics, Proceedings of the International Conference on Smart Objects and Ambient Intelligence*, 275-280.
- Saffiotti, A., Broxvall, M. & Gritti, M. (2008). *The PEIS-ecology project: vision and result. IEEE International Conference on Intelligent Robots and Systems IROS 2008*, 2329-2335.
- Sato, T. (2007). Bee system finding solution by a concentrated search, *Conference on Computational Cybernetics and Simulation, IEEE*, 3954-3959.
- Shannon, C. E. (1951). Presentation of a maze-solving machine. *8th Conf. of the Josiah Macy Jr. Found*, 173-180.
- Silva Ortigoza, R., García Sánchez, R., Barrientos Sotelo, R., Molina Vilchis, M. A., Hernández Guzmán, V. M., & Silva Ortigoza, G. (2007). Una panorámica de los robots móviles. *Télématique*, 6(3), 1-14.
- Sparks, J. (1997). *Vida y costumbres de las aves*. Barcelona, España. Editorial Bruguera.
- Steels, L. (1990). Cooperation between distributed agents through self-organization. *IROS'90 IEEE International Workshop on Intelligent Robots and Systems' 90, Towards a New Frontier of Applications, Proceedings*, 8-14.
- Stone, P. & Veloso, M. (2000). Multi-agent System, a Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3). 345-383
- Stormont, D. P. (2005). Autonomous Rescue Robot Swarms for First Responders. *Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*. 151-157.
- Svennebring, J. & Koenig, S. (2004). Building terrain-covering ant robots: A feasibility study. *Autonomous Robots*, 16(3), 313-332
- Tamayo, F. (2010). Qué es QFD, Descifrando el Despliegue de la Función de Calidad. Asociación Latinoamericana de QFD. <http://www.qfdlat.com> 2010
- Tilden, M. W. (1994). Patente EUA No. 5,325,031. Washington, DC: Oficina de Patentes y Marcas de los Estados Unidos.
- Toal, D., Flanagan, C., Jones, C., & Strunz, B. (1996). Subsumption architecture for the control of robots. In Proc. of the IMC, 13.
- Todd, D. J. (1985). *Walking machines: an introduction to legged robots*. Chapman & Hall.

- Trianni, V., Nolfi, S. (2009). Self-organizing sync in a robotic swarm: A dynamical system view. *IEEE Transactions on Evolutionary Computation*, 13(4), 722-741.
- Trianni, v., Nolfi, S., & Dorigo, M. (2008). Evolution, self-organization and swarm robotics. *Swarm Intelligence*, 163–191.
- Tuci, E., Ampatzis, C., Vicentini, F., & Dorigo, M. (2006). Evolved homogeneous neuro-controllers for robots with different sensory capabilities: coordinated motion and cooperation. In *From Animals to Animats*, 9, 679-690.
- Vallejo, M., Ochoa, J. F., & Jiménez, J. A. (2009). Sistemas multi-agente robóticos: Revisión de metodologías. *Revista Avances en Sistemas e Informática*, 6(3), 189-196.
- Verret, S. (2005). Current state of the art in multirobot systems. Defence Research and Development Canada-Suffield, Ralston ALTA, Technical Memorandum DRDC-SUFFIELD-TM-2005-241.
- Walter, W. G. (1950). An Imitation of Life. *Scientific American*, mayo, 42-45.
- Walter, G. (1953). *The Living Brain*. Londres, Inglaterra. Gerald Duckworth & Co.
- Webb, B. (1996). A cricket robot. *Scientific American* 275(6), 94-99.
- Webb, B. (2006). Validating Biorobotic Models. *Journal of Neural Engineering*, 3, 25-35.
- Webb, B. (2009). Animals versus animats: ¿Or why not model the real iguana? *Adaptive Behavior*, 17(4), 269-286.
- Wiener, N. (1948). *Cybernetics: Control and communication in the animal and the machine*. New York, Estados Unidos. Wiley.
- Wilensky, U., & Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. MIT Press.
- Wilson, E. O. (1971). *The insect societies*. The insect societies.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02), 115-152.
- Yoneda, K. & Ota, Y. (2003). Non-bio-mimetic walkers. *International Journal of Robotics Research*, 22, 241-249

ANEXO 1. SOFTWARE DE SIMULACIÓN

CARACTERÍSTICAS GENERALES DEL SOFTWARE ELEGIDO (NETLOGO):

Es un entorno de programación que permite la simulación de fenómenos naturales y sociales. Fue creado por Uri Wilensky en 1999 y está en continuo desarrollo por el Center for Connected Learning and Computer-Based Modeling. Originariamente se creó como una herramienta educativa para niveles básicos, como ampliación de StarLogo (que a su vez ampliaba el lenguaje LOGO original). Es un entorno integrado de programación gratuito, dedicado específicamente a la modelización y simulación de fenómenos naturales y sociales como sistemas complejos que evolucionan con el tiempo y constituidos por centenares o miles de “agentes” independientes que trabajan de forma concurrente.

Dispone de una extensa documentación, tutoriales para su aprendizaje y una biblioteca de modelos referentes a diversas disciplinas que pueden ser reutilizados y modificados, además de una amplia y creciente comunidad de usuarios.

Las simulaciones generadas por Netlogo están implementadas internamente en lenguaje Java, pero no es preciso conocer Java para programar modelos Netlogo ya que se utiliza un lenguaje de programación propio.

Como características distintivas, además de un entorno “inteligente” de programación (que controla en tiempo real diversos aspectos relativos a la correcta escritura del código Netlogo), Netlogo incorpora una herramienta de automatización para la repetición de simulaciones, incluso con cambios de parámetros iniciales, y para la exportación de resultados en formato CSV (recuperables desde MS-Excel, SPSS, etc.).

Puede ejecutarse en todo tipo de sistemas operativos, o incluso a través de la web, mediante un navegador. A partir de la versión 5 (beta01, marzo 2011) incluye una interfaz en castellano.

Es particularmente útil para modelar sistemas complejos que evolucionan en el tiempo. Los implementadores de modelos pueden dar instrucciones a cientos o miles de agentes para que todos ellos operen de manera independiente, entre sí y con el entorno. Esto hace posible explorar la relación entre el comportamiento a bajo nivel de los individuos y los patrones macroscópicos que surgen a partir de la interacción de muchos individuos entre sí.

Netlogo permite a los usuarios abrir simulaciones y “jugar” con ellas, así como explorar su comportamiento bajo una serie de condiciones. Asimismo, permite al usuario la creación de sus propios modelos. Netlogo es lo suficientemente sencillo como para que los estudiantes y los profesores puedan ejecutar las simulaciones o incluso construir las suyas propias. Además, su grado de desarrollo actual es suficiente como para servir como una herramienta potente para investigadores en muchos ámbitos.

Como ejecutar el Software:

Existen dos maneras de ejecutar Netlogo:

1. Descargando e instalando el programa (permite simular y editar modelos, así como la creación de modelos propios).
2. Ejecutar un applet desde una página web (permite la ejecución de los modelos, pero no editarlos ni crear modelos nuevos).

El programa puede descargarse de manera gratuita desde aquí. Para su funcionamiento, requiere tener instalada en el ordenador una máquina virtual de Java (JVM - Java Virtual Machine) versión 1.4.2 o superior.

Funcionamiento del software:

Netlogo es un lenguaje de programación que sigue la filosofía del modelado basado en agentes. Concretamente, en Netlogo existen 3 tipos de agentes:

- Turtles (tortugas).
- Patches (celdas).
- Links (relaciones entre tortugas).
- Observer (observador).

Las tortugas son los agentes que se mueven por el mundo. Interaccionan entre sí y con el medio. Cada tortuga viene identificada por un identificador que es único para cada tortuga. Netlogo denomina “mundo” (world) al terreno en el que se mueven las tortugas. Cada porción cuadrada de mundo se denomina patch. Cada patch está identificado por las coordenadas de su punto central.

Las tortugas se mueven por el mundo (y, por tanto, por encima de los patches). Las tortugas interaccionan entre sí según unas reglas de comportamiento y con el medio (es decir, con los patches). Se pueden modelar la relación entre distintas tortugas mediante links, que es el tercer tipo de agente presente en Netlogo. Los links se designan mediante un par (tortuga1, tortuga2), que indica las dos tortugas relacionadas mediante dicho link.

Finalmente, la última figura presente en los modelos de Netlogo es el observador. Éste no está representado en el mundo, pero puede interactuar con él (crea y destruye agentes, asigna propiedades a los agentes, etc).

ENTORNO DE SIMULACIÓN DE LOS ALGORITMOS

Esta es una simulación de un enjambre de robots cuya misión es encontrar la ruta óptima entre un punto de salida y un punto de llegada. El algoritmo se basa en la metaheurística ACO (Ant Colony Optimization) e incorpora una aproximación a un sistema nervioso biológico al incluir comportamientos reactivos, instintivos e inteligentes. La ruta óptima se encuentra mediante el comportamiento emergente logrado por el enjambre, a partir de acciones simples ejecutadas por los agentes individuales.

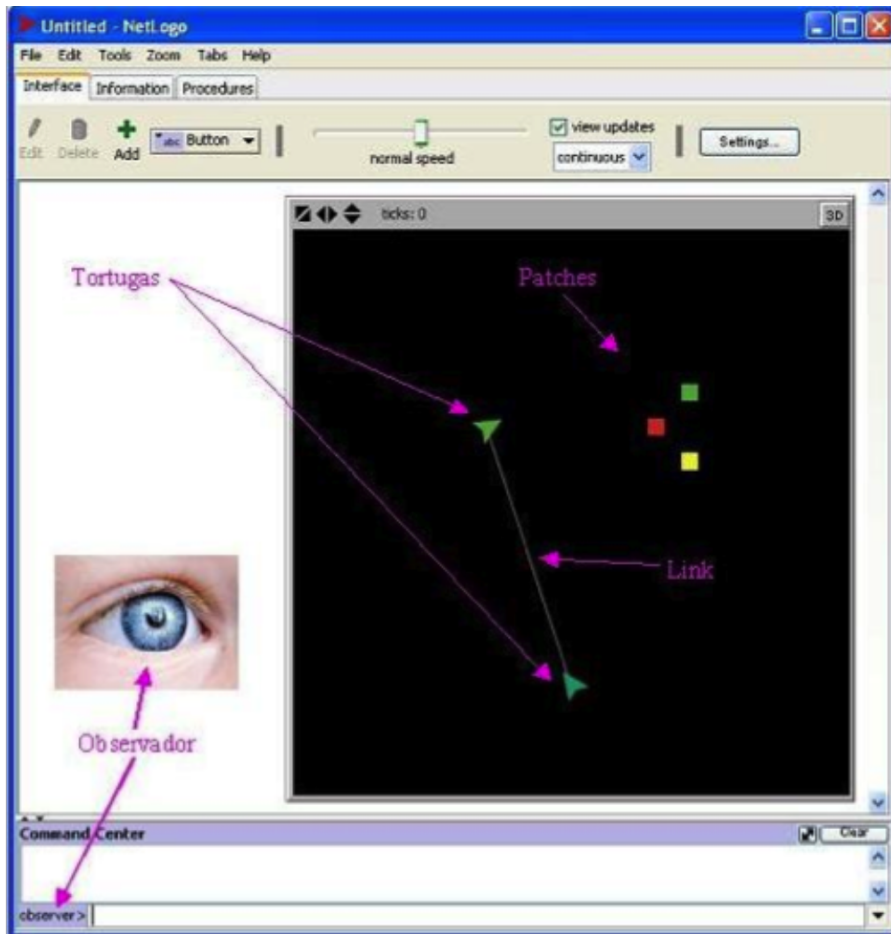


Figura 85. Entorno de simulación NETLOGO

Funcionamiento:

- Los robots salen de la base explorando el entorno con movimientos aleatorios.
- Si perciben el “sonido” del objetivo los robots se dirigen hacia este como fuente de sonido.
- Cuando los robots están más cerca del objetivo perciben su luz, entonces se dirigen hacia este como fuente de sonido, aumentando la prioridad de encaminamiento hacia el objetivo.
- Una vez que los robots hacen contacto con el objetivo regresan a la base dejando un rastro de feromona.
- Si un robot percibe feromona ignora los demás estímulos y sigue el camino trazado por la feromona.
- Al chocar con un obstáculo los robots lo evaden.

Como se usa:

- Determinar las coordenadas del objetivo mediante los controles deslizantes:

- Posición_X: coordenada ‘x’ del objetivo (control ubicado en la parte inferior del plano).
- Posición_Y: coordenada ‘y’ del objetivo (control ubicado a la izquierda del plano).
- Seleccionar la estrategia que va a usarse en la simulación, Enjambre o Multiagente.
- Elegir la cantidad de obstáculos con el control deslizante correspondiente, mínimo 1 máximo 10; los obstáculos se ubicarán en posiciones aleatorias.
- Establecer los porcentajes de sensibilidad a los estímulos, sonido, luz y obstáculos, mediante los controles deslizantes correspondientes.
- Indicar las tasas de evaporación y de difusión de feromona.
 - Tasa de evaporación: un valor menor indica mayor permanencia de la feromona.
 - Tasa de difusión: mayor difusión de feromona indica mayor radio de acción.
- Presionar el botón “configurar” para aceptar los parámetros indicados, el programa establece el entorno de la siguiente manera:
 - Base: punto de origen de las trayectorias de los robots, es ubicado en la parte izquierda del plano y se representa mediante una circunferencia de color azul.
 - Objetivo: ubicado en la parte derecha del plano, en las coordenadas elegidas por el usuario, se representa mediante una circunferencia de color amarillo.
 - Obstáculos: ubicados aleatoriamente en el área central del plano, se representan mediante círculos de color café.
- Iniciar la ejecución de la simulación:
 - Ejecución por pasos: presionar el botón “por pasos” una vez por cada paso.
 - Ejecución continua: al presionar el botón “continua” las iteraciones del algoritmo se ejecutan en un ciclo indefinido; para pausar la simulación se debe hacer click en el botón “continua” (este vuelve a su posición inicial de no presionado); para reanudar la simulación se debe presionar el mismo botón una vez más.
- A la derecha del plano de entorno de simulación, se encuentra un área de trazado donde se indica gráficamente la cantidad de percepciones de luz, sonido y objetivo:
 - Percepción de sonido: solo en el caso de usar estrategia multiagente
 - Percepción de luz: indica la cantidad de percepciones instantáneas, es decir la cantidad de robots que estén percibiendo este estímulo en cada iteración del algoritmo.
 - Percepción de objetivo: indicador incremental que muestra el número de robots que han llegado al objetivo, aunque no necesariamente permanezcan junto a este.
- Debajo del área de trazado de gráficos se muestran los indicadores numéricos de los valores instantáneos de las tres gráficas mencionadas en el punto anterior.
- El tiempo es medido en “ticks” o pasos de ejecución de la simulación, el número de ticks se muestra en un indicador numérico ubicado debajo del área de gráficos.

ANEXO 2. TÉCNICAS DE ROBÓTICA

Robótica basada en comportamientos:

Es un enfoque que se centra en exhibir comportamientos de apariencia compleja, principalmente al ir corrigiendo gradualmente las acciones a través de enlaces sensor-motor. La mayoría de estos sistemas son también *reactivos*³, lo que significa que no necesitan programación de representaciones internas; en cambio toda la información se obtiene de lo percibido por los sensores del robot, el cual la utiliza para corregir gradualmente sus acciones de acuerdo a los cambios en su entorno inmediato. En lugar de construir sistemas inteligentes por resoluciones abstractas, y altamente formalizadas de problemas, la robótica basada en comportamientos se enfoca en sistemas que podrían ser considerados “no inteligentes”, inspirados por la naturaleza (Belpaeme & Birk 2001). Los robots basados en comportamientos presentan acciones de apariencia más biológica que las de sus homólogos de computación intensiva, a menudo cometen errores, repiten acciones, y parecen confundidos, esto ha llevado a que se les compare con los insectos. Estos robots a veces se consideran ejemplos de *Inteligencia Artificial Débil*⁴, aunque algunos investigadores han afirmado que son modelos de la inteligencia completa (Brooks, 1991C).

Una de las aplicaciones de la robótica basada en comportamientos está en los equipos multi-robot (Kaminka & Frenkel 2005) con el objetivo de desarrollar mecanismos genéricos simples que resulten en un comportamiento de grupo coordinado, ya sea implícita o explícitamente.



Figura 86. Robótica basada en comportamientos. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press

Según Brooks (Brooks 1991C) el problema del comportamiento se puede dividir en varios campos: el nivel *micro*, el nivel *macro* y el nivel *multitud*:

- **Nivel micro:** se encarga de comportamientos unitarios como seguir una pelota, atravesar una puerta, mover un brazo, etc.
- **Nivel macro:** se encarga de la integración de todo un repertorio de comportamientos micro en un solo individuo.
- **Nivel multitud:** pone énfasis en comportamientos sociales, de conjuntos de individuos.

Robótica Evolutiva:

³Ver sección 6.4.3 Agentes Reactivos

⁴Ver sección 6.4.1

Es una nueva técnica para la creación automática de robots autónomos, se inspira en el principio de la reproducción selectiva de los más aptos; basándose en la biología y la etología, la robótica evolutiva hace uso de herramientas como redes neuronales, algoritmos genéticos, sistemas dinámicos e ingeniería biomórfica. Los algoritmos operan sobre poblaciones de controladores candidatos, repetidamente modificadas de acuerdo con una función de aptitud. En el caso de los algoritmos genéticos, la población de controladores candidatos se cultiva de acuerdo a cruces, mutaciones y otros operadores y luego son sacrificados de acuerdo con una función de aptitud.



Figura 87. Aproximaciones de la Robótica evolutiva. Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press

Los controladores candidatos utilizados en aplicaciones de Robótica Evolutiva pueden provenir de algún subconjunto del conjunto de redes neuronales artificiales, aunque algunas aplicaciones usan colecciones de reglas "If-Then-Else" como partes constitutivas de un controlador individual. En teoría, es posible utilizar cualquier conjunto de formulaciones simbólicas de leyes de control como el espacio de posibles controladores.

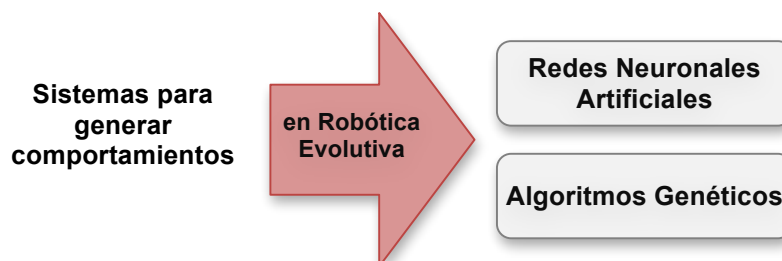


Figura 88. Sistemas para generar comportamientos en Robótica Evolutiva Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press

Robótica Cognitiva:

Se ocupa de proporcionar a los robots funciones cognitivas superiores de nivel, al darles una arquitectura de procesamiento que les permitan razonar, aprender y actuar frente a objetivos complejos en entornos complejos.

Un robot cognitivo debe presentar las siguientes características:

- Conocimiento.
- Creencias.
- Preferencias.
- Objetivos.
- Actitudes informativas.
- Actitudes motivacionales (observación, comunicación, revisión de creencias, planificación).
- Capacidad *para moverse* en el mundo físico, y para interactuar de forma segura con sus objetos.

La robótica cognitiva implica el comportamiento de los robots en el mundo físico (o en un mundo virtual, en el caso de la robótica cognitiva simulada), en última instancia el robot debe ser capaz de actuar en el mundo real.

Dentro de la robótica cognitiva, pueden adoptarse diferentes metodologías que incluyen no sólo el enfoque de la IA simbólica clásica (haciendo énfasis en el razonamiento y la representación simbólicos), sino también los enfoques inspirados más biológicamente (que usan representaciones ruidosas y distribuidas).

La robótica cognitiva ve en la cognición animal el punto de partida para el desarrollo de los algoritmos computacionales, *a diferencia de las técnicas tradicionales de inteligencia artificial*, es por esto que las capacidades cognitivas de la robótica incluyen: el proceso de la percepción, la asignación de la atención, la anticipación, la planificación, la coordinación motora compleja, el razonamiento sobre otros agentes, y quizás el razonamiento sobre sus propios estados mentales.

Robótica Epigenética:

También llamada robótica de desarrollo (*developmental robotics*), estudia los mecanismos de desarrollo, arquitecturas y restricciones que permiten, a los robots, el aprendizaje permanente y abierto de nuevas habilidades y conocimientos. Se espera que este aprendizaje sea acumulativo y de complejidad creciente, y que se dé como resultado de la exploración autónoma del entorno, en combinación con la interacción social.

Debido a que el concepto de máquina inteligente adaptativa es fundamental para la robótica epigenética, se tienen relaciones con campos como la inteligencia artificial, el aprendizaje automático, la robótica cognitiva y la neurociencia computacional. Sin embargo, si bien puede reutilizar algunas de las técnicas elaboradas en estos campos, se diferencia de ellos así:

- Se diferencia de la Inteligencia Artificial clásica, pues no asume capacidad de razonamiento simbólico avanzado y se centra en habilidades sociales y sensorimotoras y no en los problemas simbólicos abstractos.
- Se diferencia del aprendizaje de máquina tradicional, ya que apunta a un aprendizaje auto-determinado e independiente de la tarea y no en inferencias específicas hacia la tarea.
- Se diferencia de la Robótica Evolutiva, ya que esta utiliza poblaciones de robots que evolucionan con el tiempo, mientras que la robótica epigenética se ocupa en cómo la

organización del sistema de control, de un solo robot, se desarrolla a través de la experiencia.

- Se diferencia de la Robótica Cognitiva, pues se centra en los procesos que permiten la formación de las capacidades cognitivas en lugar de en estas mismas capacidades.
- Se diferencia de la Neurociencia Computacional, ya que se enfoca en el modelado funcional de arquitecturas integradas de desarrollo y aprendizaje.

En términos más generales, la robótica Epigenética presenta las siguientes tres características:

- Se enfoca hacia las arquitecturas no dependientes de tareas y los mecanismos de aprendizaje, la máquina tiene que ser capaz de aprender nuevas tareas desconocidas por el investigador.
- Hace énfasis en el desarrollo abierto y el aprendizaje permanente, es decir, en la capacidad de un individuo para adquirir nuevas habilidades. Esto no debe entenderse como una capacidad de aprender "todo" sino como que el conjunto de habilidades que se adquieren se puede extender infinitamente al menos en algunas direcciones.
- La complejidad de conocimientos y habilidades adquiridas se incrementa progresivamente.

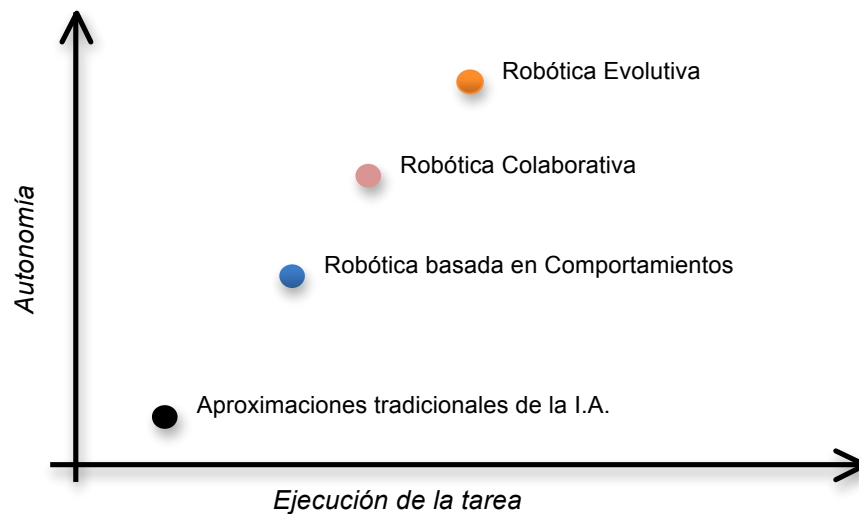


Figura 89. Desempeño de las diferentes técnicas de Robótica. En el eje vertical, se mide la autonomía del tipo de robótica (origen significa poca autonomía) y en el eje horizontal, el desempeño o ejecución de la tarea (cerca al origen es desempeño pobre). Fuente: Adaptado de Liu, J. & Wu, J. (2001). Multi-Agent Robotic Systems. Boca Ratón, USA. CRC Press

ANEXO 3. SÍNTESIS DE UN SISTEMA NERVIOSO

La arquitectura propuesta en este trabajo es una arquitectura reactiva, basada en el paradigma de subsunción puesto que permite acciones inhibitorias; de esta manera es posible que los robots dispongan de diferentes modos de operación, de acuerdo a las situaciones en que se encuentren dentro del entorno. Esta arquitectura hace una analogía con el sistema nervioso de un insecto, el cual se ha subdividido en tres subsistemas: *Subsistema Reflejo*, *Subsistema Instintivo* y *Subsistema Inteligente*. Así pues, la *Arquitectura de Sistema Nervioso* permite a los robots exhibir modos de desplazamiento comparables a los comportamientos reflejo, instintivo e inteligente de los seres vivos, dependiendo de la situación ante la cual se encuentren.

Modelo estímulo-respuesta: Dado que la conducta puede definirse simplemente como la respuesta de un organismo al medio ambiente; toda forma de conducta, sea simple o compleja, es el resultado de las respuestas a los estímulos. Un estímulo es generalmente un cambio en un aspecto determinado del medio ambiente de un organismo (Carthy, 1970). La respuesta es una reacción positiva o negativa que presenta un organismo frente a determinados factores externos o internos. El modelo estímulo-respuesta es una función matemática que describe la relación f entre un estímulo dado x y el valor esperado E (u otra medida de ubicación) de la respuesta y :

$$E(y) = f(x)$$

La forma más frecuente que asumen estas funciones es lineal, y se espera ver una relación como:

$$f(x) = A + Bx$$

Cabe anotar que, para el estudio del modelo estímulo/respuesta, se requiere el uso de regresión lineal puesto que no siempre se sabe la causa de un determinado comportamiento, ya que el estado interno de un organismo no solo produce tendencia hacia ciertas clases de conducta, sino que también cambia las respuestas usuales a los estímulos (Carthy, 1970).

Descripción de comportamientos: El funcionamiento de los robots está basado en respuestas a estímulos que provee el entorno, con miras a imitar el comportamiento que presentan las hormigas en su búsqueda de alimento (evasión de obstáculos y selección de rutas más cortas o más prometedoras); de esta manera, los robots evitan obstáculos y siguen contornos mientras buscan continuamente un lugar que presente las condiciones requeridas.

De acuerdo a las funciones del robot, sus acciones se clasifican como comportamientos que, según la arquitectura propuesta, deben ser reflejos, instintivos o inteligentes:

- **Comportamiento reflejo:** evasión de obstáculos
- **Comportamiento instintivo:** acercamiento a fuentes de luz
- **Comportamiento inteligente:** seguimiento de feromona artificial

Según lo anterior, se tienen 3 tipos de sensores, de obstáculo, de luz y de feromona; las señales de los sensores se dirigen a los respectivos sistemas y de allí parten las señales a los motores para efectuar los movimientos del robot, según la Figura 90.

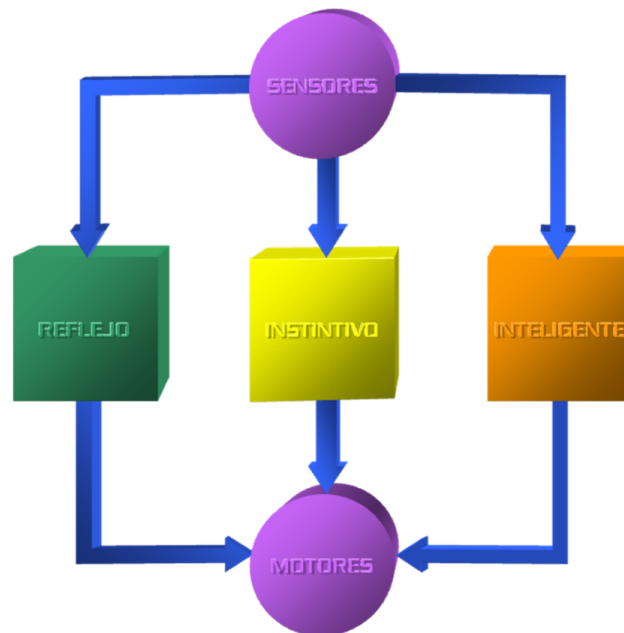


Figura 90. Comunicación entre sensores y motores con los subsistemas de la arquitectura de sistema nervioso.

Funciones de inhibición: Los subsistemas tienen una jerarquía de acuerdo a la prioridad de funcionamiento; de acuerdo a la arquitectura algunos subsistemas pueden inhibir el funcionamiento de otros; en esta arquitectura se dan 3 casos para las funciones de inhibición, de acuerdo a las funciones de los componentes del sistema nervioso:

- **Caso 1:** Se presentan simultáneamente estímulos para los sistemas reflejo e instintivo. La prioridad de acciones la tiene el sistema reflejo, este envía una señal de inhibición al sistema instintivo, el cual queda en espera, mientras se perciba el estímulo que disparó el reflejo (Figura 91a). Se observa que primero se debe sortear el obstáculo para poder acercarse a la fuente de luz.
- **Caso 2:** Se presentan simultáneamente estímulos para los sistemas instintivo e inteligente. La prioridad de acciones la tiene el sistema inteligente, este envía una señal de inhibición al sistema instintivo, el cual queda en espera, mientras se percibir el estímulo que disparó el comportamiento inteligente (Figura 91b). Se tiene que es más probable llegar a una fuente de luz siguiendo un camino reforzado de feromona.
- **Caso 3:** Se presentan simultáneamente estímulos para los sistemas reflejo e inteligente. Se presentan dos posibilidades, en la primera el sistema reflejo inhibe al inteligente, que queda en espera mientras se perciba el estímulo que disparó el reflejo (Figura 92); en este punto, se tiene que el obstáculo impide seguir la trayectoria de feromona, por lo tanto debe ser evadido (este comportamiento corresponde con las acciones del sistema nervioso simpático en los sistemas biológicos).

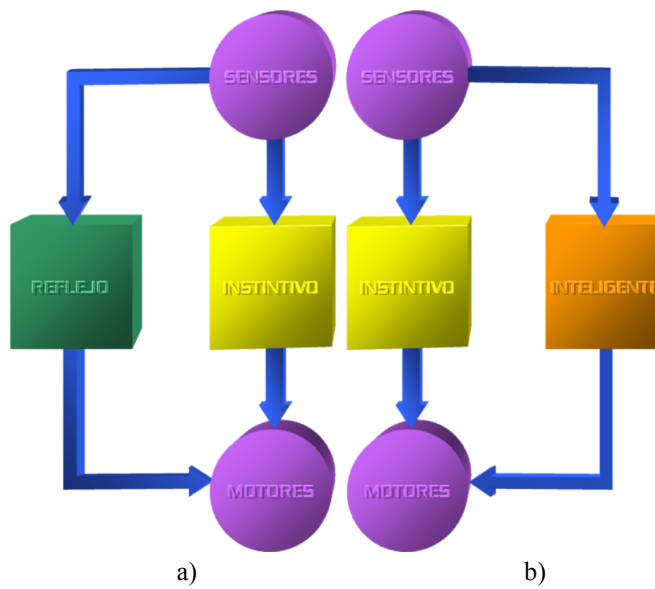


Figura 91. a) Inhibición del sistema instintivo por parte del sistema reflejo
 b) Inhibición del sistema instintivo por parte del sistema inteligente.

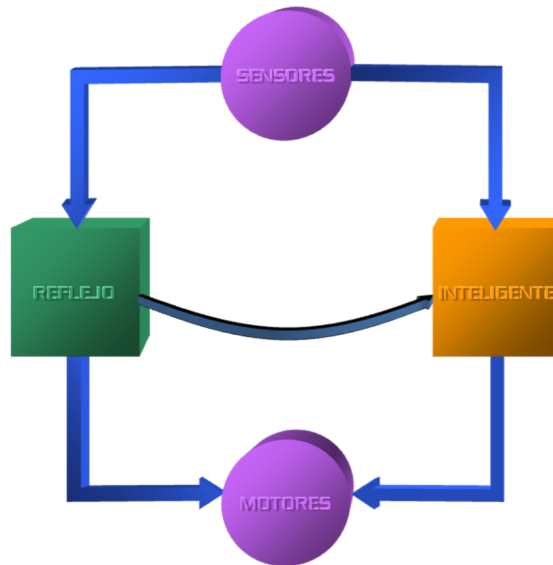


Figura 92. Inhibición del sistema inteligente por parte del sistema reflejo

En la segunda posibilidad, el sistema inteligente inhibe al reflejo, mientras se perciba el estímulo que disparó el comportamiento inteligente (Figura 94); en el ejemplo, se tiene que el obstáculo es realmente un objeto luminoso el cual no debe ser evadido (en los sistemas biológicos, este comportamiento corresponde al sistema nervioso parasimpático).

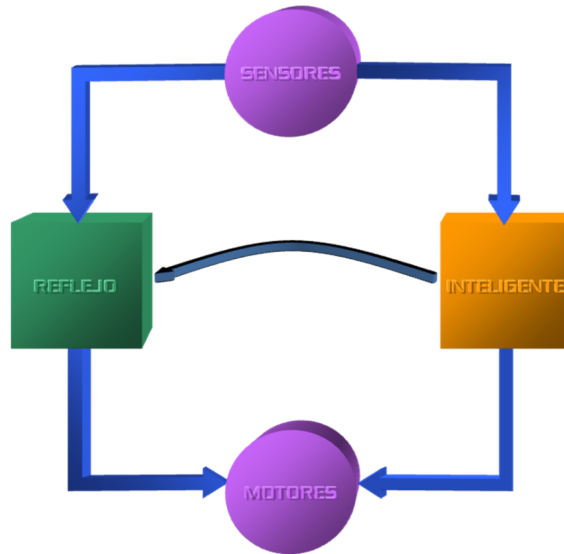


Figura 93. Inhibición del sistema reflejo por parte del sistema inteligente

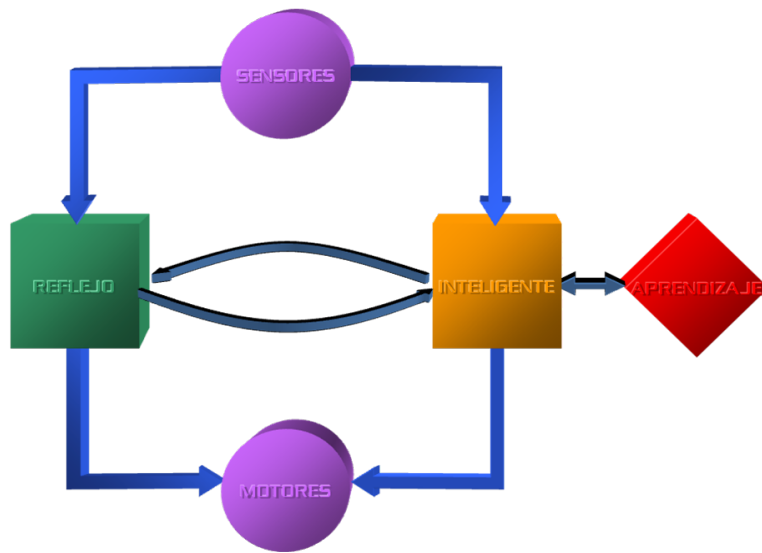


Figura 94. Señales de inhibición, entre el sistema reflejo y el sistema inteligente, dependiendo de la experiencia adquirida

Para diferenciar estas dos posibilidades, se usa la capacidad de aprendizaje (Figura 95), esto da lugar al fenómeno del *reflejo condicionado*. La arquitectura de sistema nervioso favorece la técnica de aprendizaje por refuerzo, si al seguir un rastro de feromona, se encuentra una situación desfavorable, el robot aprende a prestar menos atención a la feromona (modificando el umbral de intensidad mínima que desencadena la acción de seguir feromona), en otras palabras el robot aprende que para seguir la feromona, en un futuro, esta deberá tener mayor concentración.

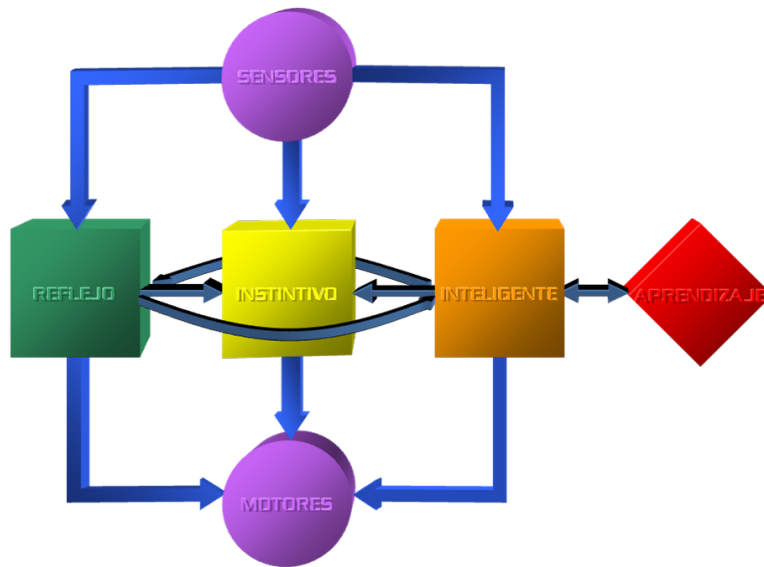


Figura 95. Señales de inhibición presentes en la arquitectura propuesta

Los objetos del entorno: Se toma como universo un entorno idealmente estático, discreto, accesible y determinista.

Se toman como bases la arquitectura de Subsuncion (Brooks, 1987) y el comportamiento de vehículos de Braitenberg (1986).

ANEXO 4. IDENTIFICACIÓN DEL SISTEMA ROBÓTICO

“El diseño y el trabajo con robots reales consume tanto tiempo y es tan poco flexible comparado con la simulación, que tal esfuerzo debería, al menos, beneficiar a ambos campos: la biología y la robótica”. Alcherio Martinoli (2001)

En lo referente a las partes de las que se compone un sistema robótico, se tiene un arreglo cinemático y un sistema de actuadores. Ambos sistemas están íntimamente ligados y son dignos de estudiarse en conjunto, no obstante se ha logrado un mayor avance, en el estado del arte, al estudiarlos por separado.

ELEMENTOS DEL CONCEPTO DE DISEÑO

A Nivel de Sistema:

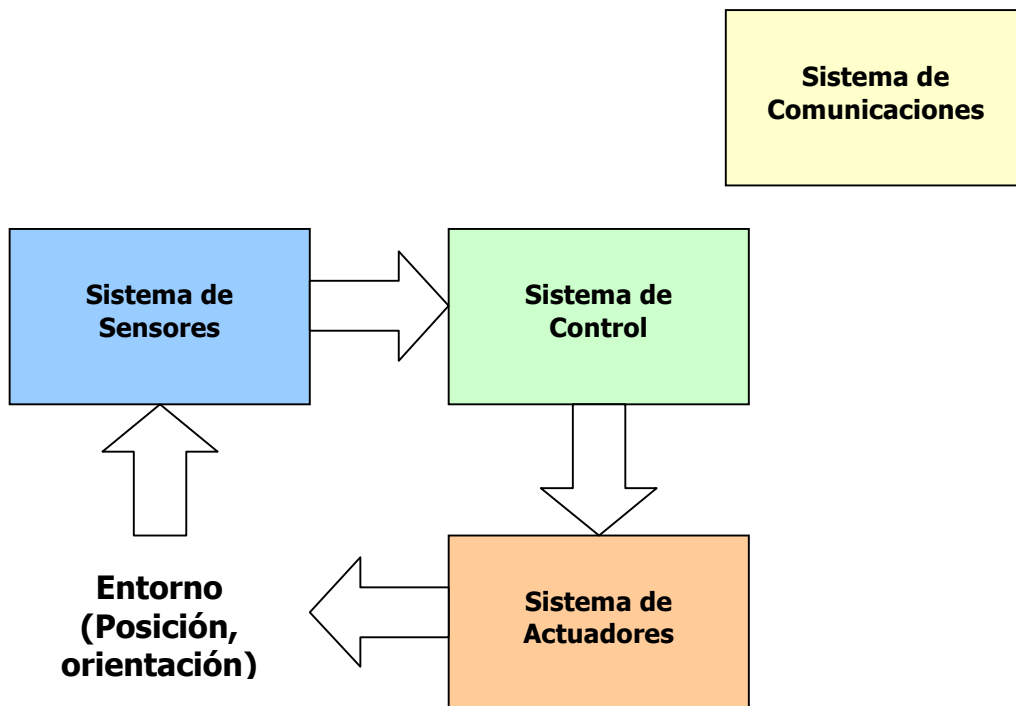


Figura 50. Organización de un Sistema Robótico a nivel de Sistemas

A Nivel de Subsistemas:

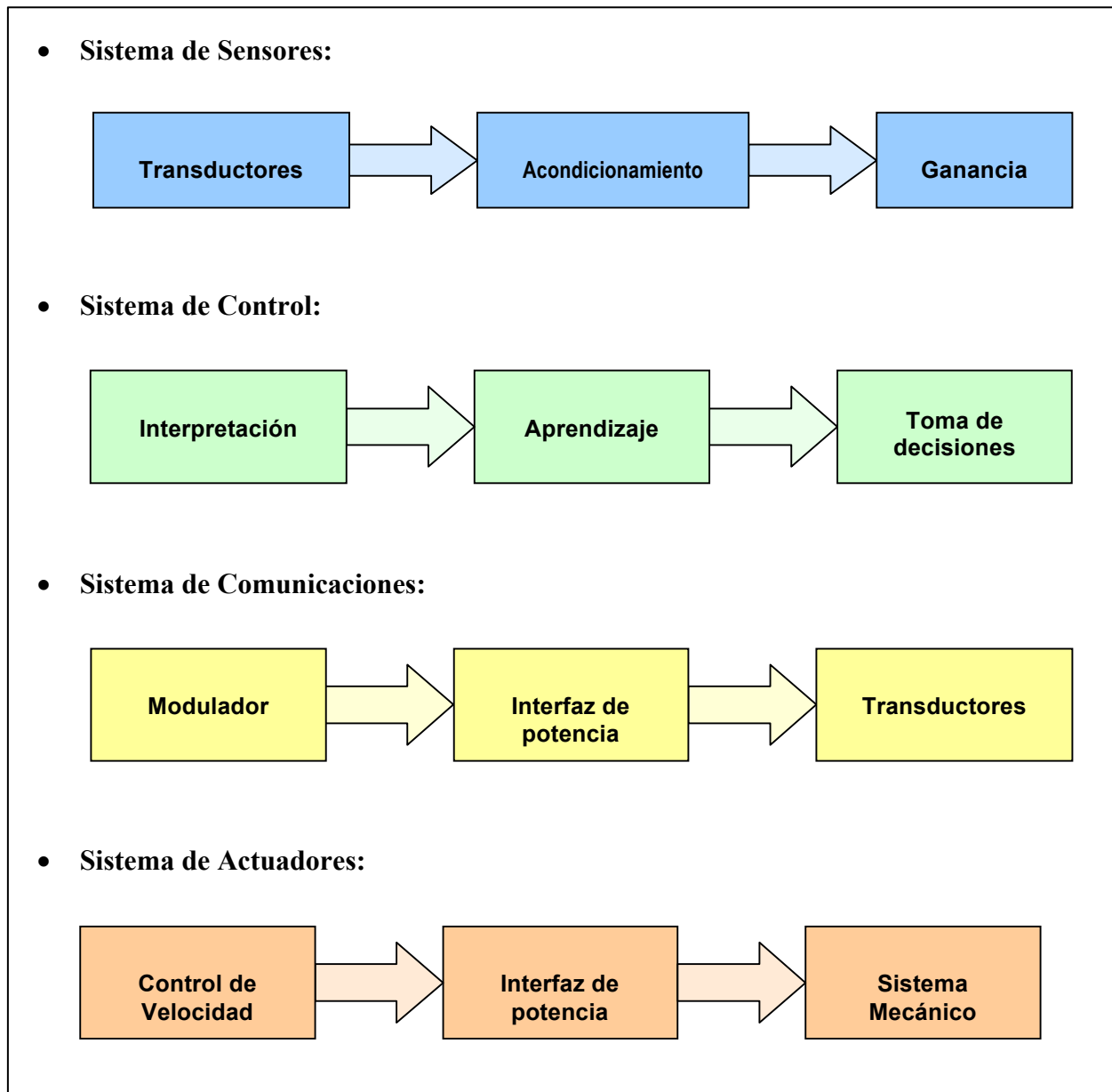


Figura 51. Organización de un Sistema Robótico a nivel de Subsistemas

A Nivel De Módulos:

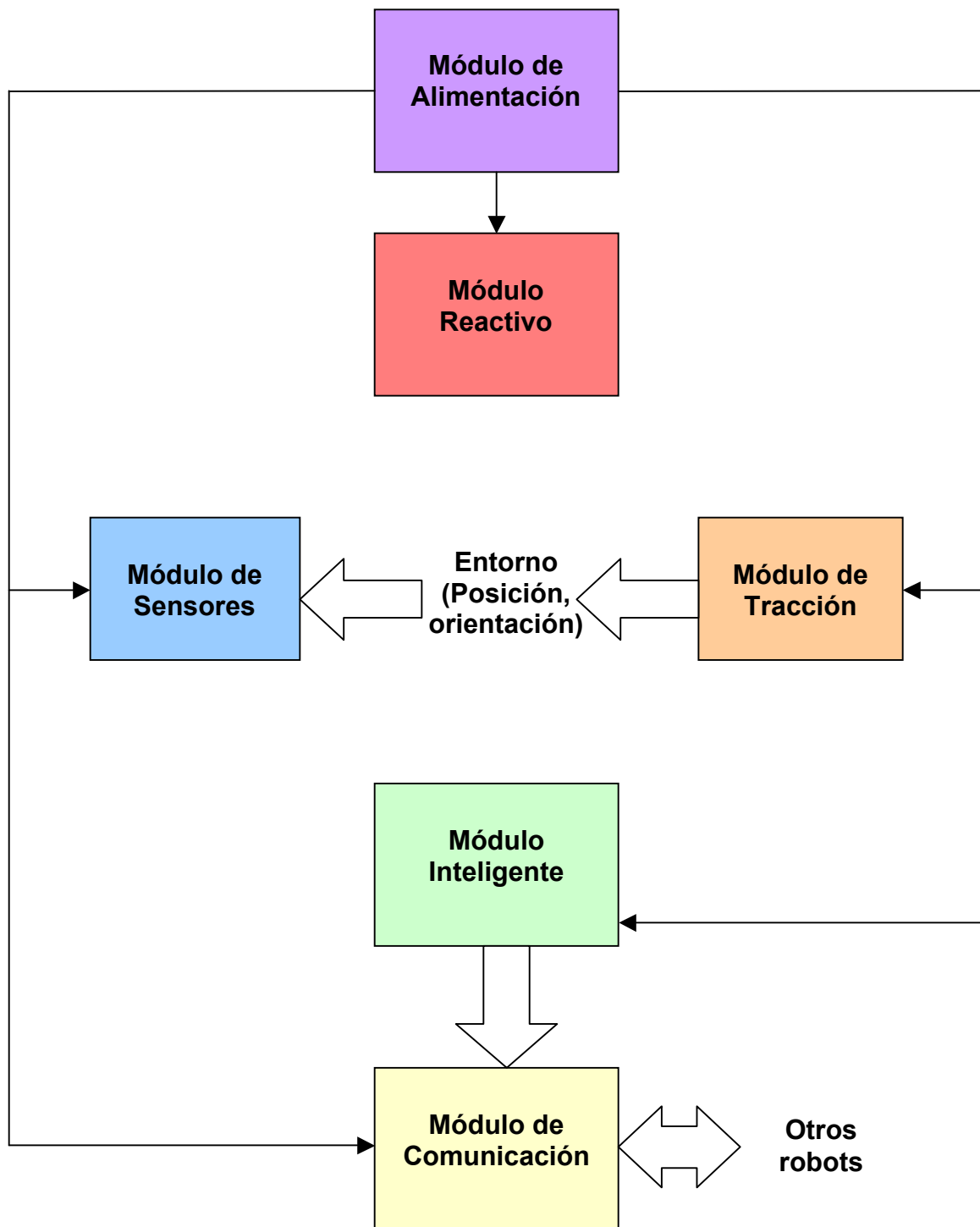


Figura 96. Organización de un Sistema Robótico a nivel de Módulos

COMPONENTES DEL SISTEMA MECATRÓNICO

Módulo De Alimentación:

- **Batería para alimentación de actuadores (motores):** Proporciona la energía requerida por los motores que se encargan del desplazamiento del robot.
- **Batería para alimentación de circuitos:** Provee una tensión estable a los circuitos electrónicos del robot.
- **Referencias activas de tensión:** Circuitos encargados de garantizar estabilidad en las tensiones de referencia que sean requeridas por los diferentes módulos, proporcionando así robustez y confiabilidad.

Módulo De Tracción:

- **Estructura mecánica (chasis):** Armazón que sirve de base para el robot, da soporte directo al sistema de tracción; determina la forma física del robot.
- **Reductores de velocidad:** Sistema de engranajes que proporciona el torque necesario para el desplazamiento del robot.
- **Motores:** Motores DC que permiten los movimientos del robot
- **Interfaz de potencia:** Recibe las señales de control provenientes de los módulos inteligente y reflejo, y entrega a los motores la energía suficiente para su funcionamiento.
- **Control de dirección y velocidad:** Recibe señales del módulo inteligente, las cuales indican la dirección hacia la cual debe desplazarse el robot, así como también la velocidad de este desplazamiento; este elemento se encarga de manejar los giros del robot y en general del proceso de navegación.

Módulo De Sensores:

- **Sensores de obstáculos:** Se encargan de detectar obstáculos y generar señales que indican la presencia y tipo de los mismos, estas señales son utilizadas por los módulos reactivo e inteligente y, a partir de sus características y de estados anteriores del sistema, el robot reacciona o decide la acción a realizar.
- **Sensores de objetivo:** Detectan la presencia de condiciones del entorno que indican un lugar específico (ya sea un punto inicial o un punto final de un recorrido); estas condiciones pueden estar dadas por el entorno mismo o mediante uso de balizas.
- **Monitor de carga de baterías:** Indica el estado de carga de las baterías del robot y envía señales a los módulos reactivo e inteligente para reaccionar o decidir la acción a realizar según el caso.

Módulo Reactivo:

- **Red Nerviosa:** Circuito que proporciona las interconexiones entre el Módulo de Sensores y el Módulo de Tracción, de esta manera permite reacciones rápidas del robot ante ciertas situaciones. Recibe además una señal proveniente del Módulo Inteligente, la cual habilita o deshabilita su funcionamiento, de acuerdo a condiciones externas e internas y a estados previos del sistema.
- **Etapa de ganancias:** Permite modificación de ganancias de las conexiones provenientes de los diversos sensores (Módulo de Sensores) y de las conexiones que van hacia los actuadores (Módulo de Tracción); las señales que permiten tales modificaciones provienen del Módulo Inteligente.

Módulo Inteligente:

- **Procesador:** Se encarga de ejecutar las acciones requeridas de acuerdo a la programación, a las señales provistas por los sensores y a las señales enviadas por otros robots; tiene en cuenta además los estados previos del sistema.
- **Memoria de datos:** Almacena los datos que puedan ser requeridos para futuras acciones ante posibles condiciones externas y/o internas.
- **Memoria de programa:** Almacena los algoritmos que determinan las estrategias de navegación y los protocolos de comunicación.
- **Acondicionadores de señal:** Adecúan las señales de entrada y de salida de acuerdo a la naturaleza y uso de estas.

Módulo De Comunicación:

- **Receptor:** Recibe las señales enviadas por otros robots y las entrega al Módulo Inteligente
- **Transmisor:** Emite las señales generadas por el sistema inteligente que pueden ser útiles para los demás robots.

PERCEPCIONES DEL MICRO-ROBOT

El micro-robot presenta diferentes comportamientos de acuerdo a las percepciones del entorno y de sus estados internos. En cuanto a las percepciones de los objetos del entorno, estas liberan comportamientos e inhiben otros, la prioridad de un determinado comportamiento sobre otro, depende de las probabilidades de éxito que estos proporcionen a la misión del robot (y por supuesto la “supervivencia” del robot influye en gran medida sobre la probabilidad de éxito). Dado que los comportamientos están directamente ligados a las percepciones del robot, las señales que provienen de los sensores se dirigen a los subsistemas del robot correspondientes (reflejo, instintivo, inteligente). La Figura 97 ilustra el flujo de las señales y las interacciones entre los subsistemas.

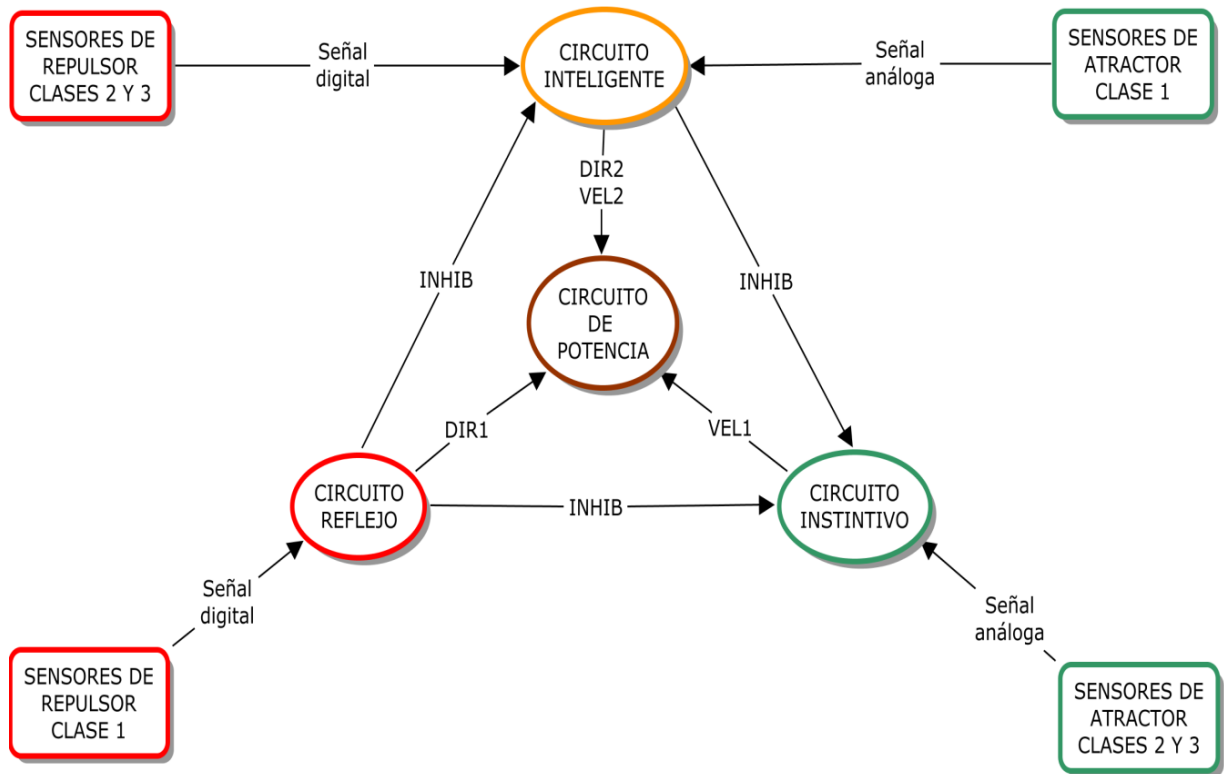


Figura 97. Flujo de señales entre los subsistemas de la arquitectura propuesta. INHIB: señal de inhibición de un subsistema a otro. DIR#: señales que indican dirección de giro del robot (izquierda o derecha) y de su desplazamiento (hacia adelante o hacia atrás). VEL#: señales que indican la velocidad de desplazamiento del robot

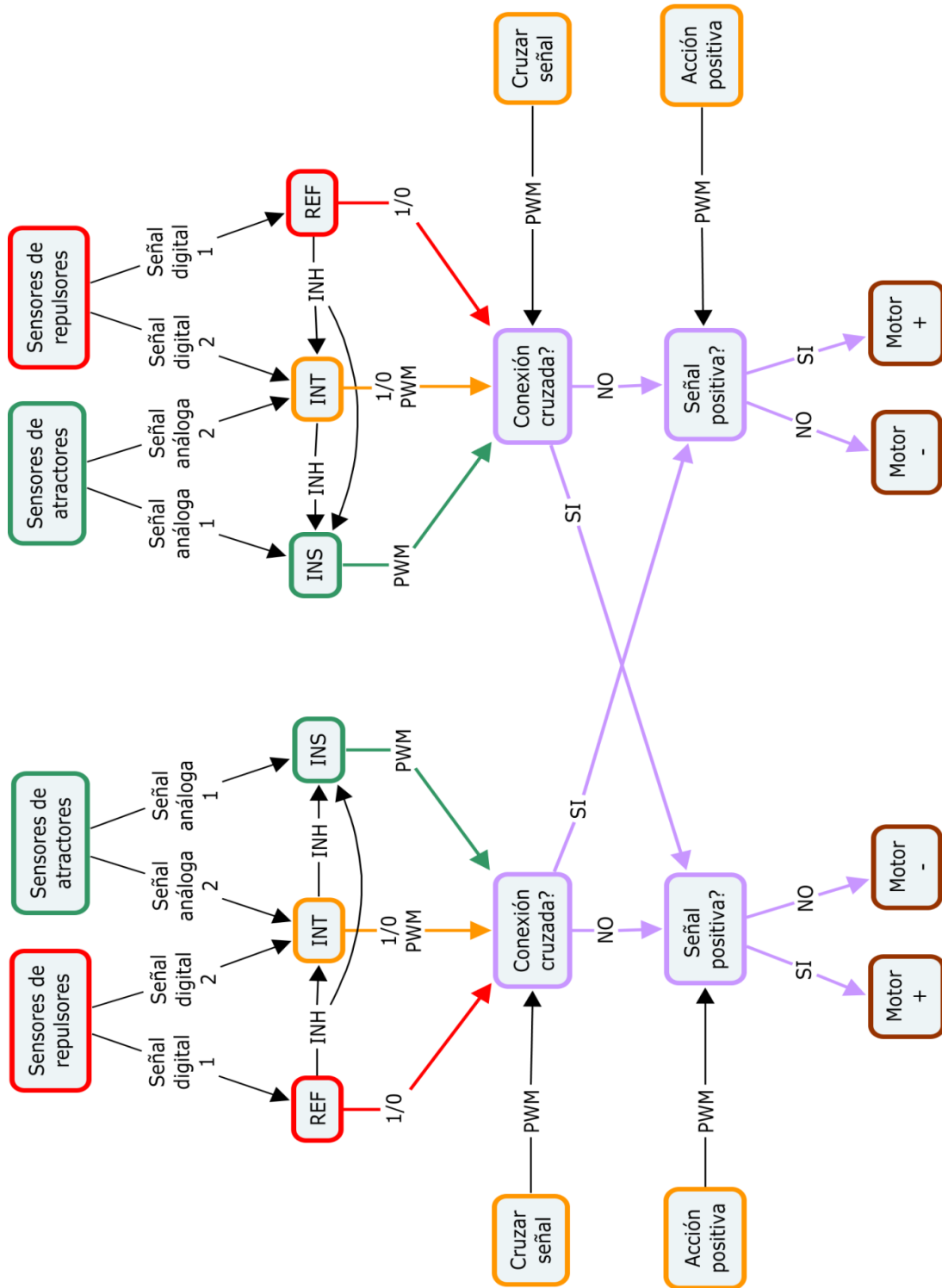


Figura 98. Conexiones Braitenberg, flujo de las señales según su naturaleza.

CONFIGURACIONES CINEMÁTICAS:

Existen diferentes configuraciones cinemáticas para los robots móviles con ruedas, estas dependen principalmente de la aplicación hacia dónde va enfocado, de manera general se tienen varias configuraciones, como se observa en la Figura 99.

Dependiendo de la configuración cinemática que lo conforme, los robots móviles con ruedas utilizan cuatro tipos de ruedas para su locomoción, estas son convencionales, tipo castor, ruedas de bolas y omnidireccionales.

En el marco de las configuraciones cinemáticas posibles y las ruedas que estas utilizan, los robots documentados en la literatura utilizan comúnmente la configuración de tracción diferencial, donde se utilizan ruedas convencionales como ruedas motrices, y una o dos ruedas locas, de bola u omnidireccionales, para proveer estabilidad. Estos robots son llamados *uniciclo* (Silva et al 2007).

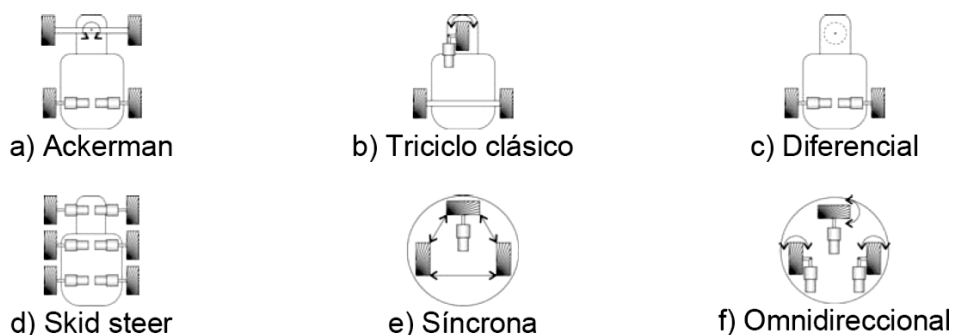


Figura 99. Configuraciones cinemáticas para los robots móviles con ruedas: a) Ackerman, b) Triciclo clásico, c) Tracción diferencial, d) Skid steer, e) Síncrona y f) Tracción omnidireccional Fuente: Silva Ortigoza, R., García Sánchez, R., Barrientos Sotelo, R., Molina Vilchis, M. A., Hernández Guzmán, V. M., & Silva Ortigoza, G. (2007). Una panorámica de los robots móviles. *Télématique*, 6(3), 1-14.

El término *uniciclo* es frecuentemente usado en la robótica y la teoría de control para denotar un carro que se mueve en dos dimensiones, siendo este uso distinto del sentido literal de "robot de una sola rueda".

Un vehículo tipo *uniciclo* en general podría ser cualquier vehículo capaz de tener simultáneamente rotación y traslación arbitrarias. Los robots *uniciclo* físicamente realizables, son sistemas holonómicos. Se trata de sistemas en el que el retorno a la configuración interna inicial no garantiza el retorno a la posición original del sistema. En otras palabras, el resultado del sistema es dependiente de la trayectoria.

Se puede decir que un robot es holonómico si es capaz de modificar su dirección instantáneamente (en esta consideración se considera masa nula), y *sin necesidad de rotar previamente*. Un robot es holonómico si tiene los mismos grados de libertad efectivos que controlables. En general, un robot no holonómico tiene menos grados de libertad controlables que número total de grados de libertad, mientras que si sucede lo contrario, el robot es redundante. En general, a mayor diferencia entre grados de libertad controlables y grados totales, más difícil será controlar el robot.

Con el objeto de hacer más tratable el problema del modelado en las configuraciones cinemáticas, se suelen establecer algunas suposiciones de diseño y de operación (Silva et al 2007).

Por una parte, dentro de las suposiciones de diseño generalmente se toman tres:

- Las partes dinámicas del robot son insignificantes, es decir que no contiene partes flexibles, de esta manera pueden aplicarse mecanismos de cuerpo rígido para el modelado cinemático.
- La rueda tiene máximo un eslabón de dirección, esto reduce la complejidad del modelado.
- Todos los ejes de dirección son perpendiculares a la superficie, de esta manera se reducen todos los movimientos a un solo Plano.

Por otra parte, respecto a las suposiciones de operación, al igual que en las de diseño, se toman tres:

- Se descarta toda irregularidad de la superficie donde se mueve el robot.
- La fricción de traslación en el punto de contacto de la rueda con la superficie donde se mueve, es lo suficientemente grande para que no exista un desplazamiento de traslación del móvil.
- La fricción rotacional en el punto de contacto de la rueda con la superficie donde se mueve, es lo suficientemente pequeña para que exista un desplazamiento rotatorio.

Aunque las suposiciones mencionadas son realistas, el deslizamiento que ocurre en el punto de contacto de las ruedas con la superficie se ha convertido en un tópico importante debido a las repercusiones que tiene sobre el móvil.

Actuadores: Relativo a los actuadores utilizados para dotar de movimiento a los robots móviles con ruedas, es común que se utilicen motores de corriente directa (CD), porque su modelo es lineal, lo que facilita enormemente su control, y específicamente los de imán permanente debido a que el voltaje de control es aplicado al circuito de armadura y el circuito de campo es excitado de manera independiente.

En este tipo de motores, se tienen dos tipos: con escobillas y sin escobillas; y aunque los motores sin escobillas tienen algunas ventajas significativas sobre los motores con escobillas, estas pueden llegar a ser irrelevantes en ciertos proyectos de robótica móvil, y además salen a relucir algunas desventajas que hacen a los motores con escobillas como la mejor opción:

- En los motores sin escobillas no se puede invertir el sentido de giro cambiando la polaridad de sus terminales, esto agrega complejidad y costo a su manejo.
- Los motores sin escobillas son más caros.
- Se requiere un sistema adicional para la conmutación electrónica.
- El controlador de movimiento para un motor sin escobillas es más costoso y complejo que el de su equivalente con escobillas.

Al igual que en el arreglo cinemático, cuando se modela un motor DC se asumen algunas consideraciones, de esta forma se establece que la única fricción presente es la viscosa, aunque en la práctica se involucran otros tipos de fricción no lineales, sin embargo, la suposición es válida al elegir un motor cuyo efecto de las fricciones no lineales sea muy pequeño.

Control: En la actualidad, el tema del control de los robots móviles con ruedas ha acaparado la atención de gran cantidad de investigadores (Silva et al 2007). Desde el punto de vista de la teoría de control, estos se encuentran en el área que se conoce como control de sistemas *no-holónomos*, estos sistemas se caracterizan por tener un número menor de grados de libertad controlables respecto al número de grados de libertad totales, en el caso de un robot móvil con ruedas de tracción diferencial, el número total de grados de libertad es 3 (posición x , y y su orientación θ) sin embargo únicamente se puede controlar el desplazamiento hacia adelante y hacia atrás así como su orientación, quedando como incontrolable el desplazamiento transversal.

Matemáticamente se dice que el sistema está sujeto a restricciones no integrables en las velocidades, es decir, su plano de velocidades está restringido. El control del movimiento estos robots, a grosso modo, se puede clasificar en cuatro tareas fundamentales (Silva et al 2007):

- Evasión de obstáculos.
- Planificación de trayectoria,
- Seguimiento de la trayectoria
- Localización,

En cuanto a la evasión de obstáculos existen distintos métodos, los más relevantes son:

- Por detección de bordes
- Por construcción de mapas
- Por descomposición en celdas
- Por campos potenciales artificiales

CONSTRUCCIÓN DEL HARDWARE

Durante el desarrollo del proyecto se ha hecho una analogía de los circuitos del microbot con el sistema nervioso de un insecto el cual, para facilitar su análisis se ha subdividido en tres sistemas: Sistema Reflejo, Sistema Instintivo y Sistema Inteligente; de acuerdo a las funciones de los circuitos que los conforman.

El diagrama en bloques de la Figura 100 muestra la conexión de los sistemas que conforman el micro-robot, cada uno de ellos recibe las señales de sus respectivos sensores para percibir las condiciones del entorno y envía señales, a través de un bus, hacia los sistemas que inhibe y hacia la etapa de potencia que maneja los actuadores, con los cuales se realiza el desplazamiento del micro-robot.

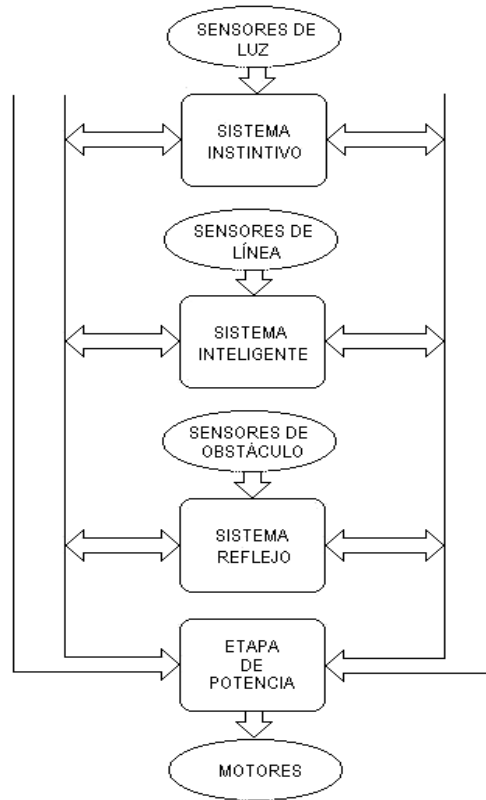


Figura 100. Diagrama en bloques propuesto para un microbot

Sistema Reflejo:

El Sistema Reflejo comprende los circuitos que perciben estímulos y envían señales directamente a los efectores (motores) sin pasar por el cerebro (microcontrolador) con el fin de efectuar, en forma rápida, las acciones pertinentes y garantizar así la pronta respuesta un estímulo que puede ocasionar daños al microbot. Concretamente, este sistema tiene las funciones de evasión de obstáculos y corresponde a la Capa 0 de la Arquitectura de Sumisión.

El Sistema Reflejo está conformado por una *red nerviosa* conformada por compuertas lógicas que toman las señales de los sensores de choque y envían la orden de invertir el giro de los motores; el mismo circuito se encarga de recibir las señales que los sistemas instintivo e inteligente envían para el control de los motores (Figura 101).

El circuito permite la conmutación de las velocidades de las ruedas. Cuando el micro-robot choca con un obstáculo, gira sobre el mismo punto para dejar el obstáculo a un lado; para que esto sea posible, ambas ruedas deben tener la misma velocidad pero girando en sentido contrario, por esta razón cada vez que una rueda invierte su giro toma la velocidad de la rueda opuesta.

Los capacitores y resistencias conectados a cada sensor de choque aseguran la permanencia de un estado bajo durante un tiempo después de perder contacto con el obstáculo, esto permite al micro-robot evadirlo con mayor eficiencia.

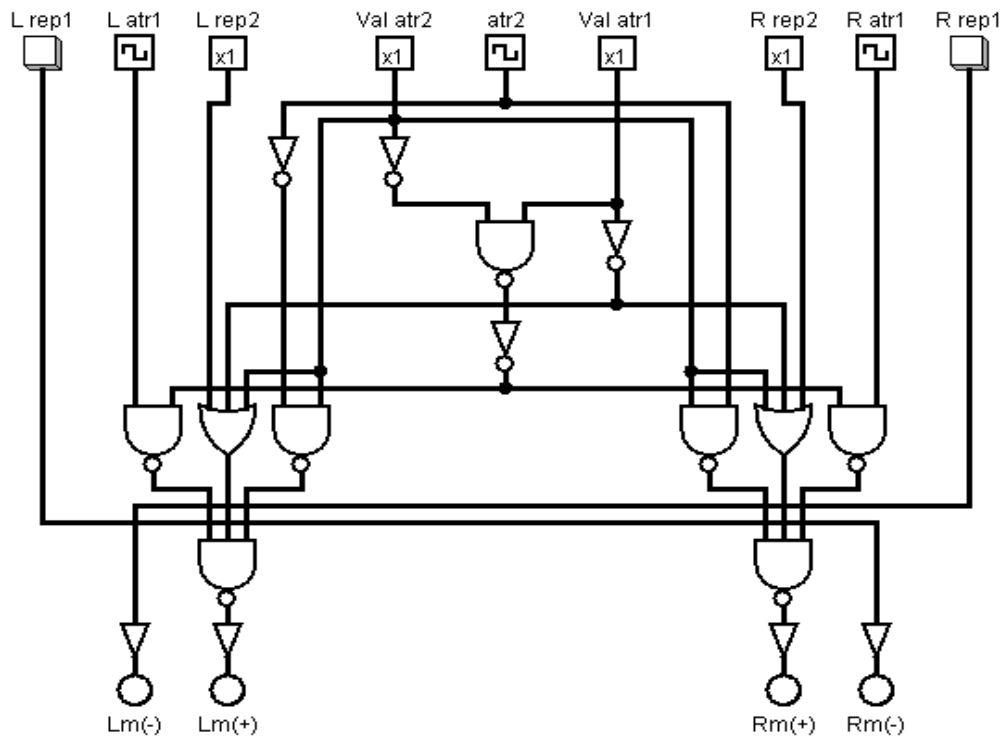


Figura 101. Diagrama del circuito del Sistema reflejo

Sistema Instintivo:

Provoca la búsqueda de la fuente de luz más potente; este comportamiento es producto de la electrónica del microbot, diseñada para procesar la información provista por el medio y enviar señales a los efectores para el desplazamiento hacia la luz más potente. Este sistema corresponde a la Capa 1 del microbot según la Arquitectura de Sumisión.

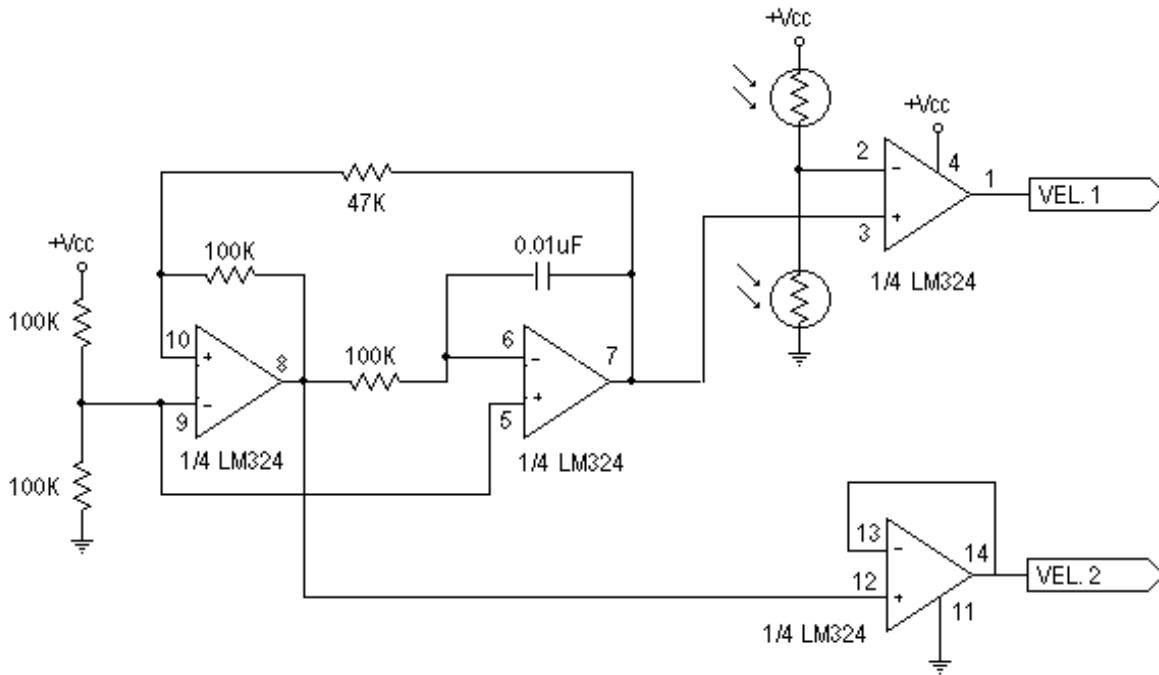


Figura 102. Diagrama del circuito del Sistema Instintivo

El circuito de PWM está hecho con 4 amplificadores operacionales, como se muestra en la Figura 102, dos de ellos forman un generador de onda cuadrada y otra triangular con una frecuencia idealmente cercana a 400Hz. Otro amplificador, configurado como comparador, se encarga de generar el pulso de ancho variable dependiendo del voltaje de control que es suministrado por el divisor de tensión formado por las dos fotorresistencias; esta es la señal de PWM, la cual se utiliza para proveer una velocidad variable a los motores.

El amplificador operacional restante está configurado como seguidor de tensión y se encarga de suministrar en su salida la onda cuadrada del generador; esta es una señal con 50% de ciclo de trabajo (aproximadamente), utilizada para proveer a los motores una velocidad constante e igual en ambos cada vez que el sistema inteligente inhibe el sistema instintivo; la razón de esto es evitar la prolongada sobrealimentación de los motores que se obtendría al utilizar una señal constantemente en estado alto.

Sistema Inteligente. Comprende los circuitos que permiten el aprendizaje a partir de la experiencia; si ocurren cambios en el medio, el microbot al igual que un insecto, “es capaz en este caso de modificar su conducta para acomodarse a los cambios, en lugar de responder siempre de un modo idéntico (y quizá ineficaz) a los estímulos particulares (Carthy, 1970).

Como eje central para este sistema se propone el uso de un sistema embebido que presente facilidad de conexión y comunicación con los circuitos propuestos. Se deja a consideración del lector la elección de este componente, con el ánimo de dejar abierta la posibilidad de expansión y crecimiento de prestaciones de la arquitectura propuesta.

ANEXO 5. PRUEBAS ESTADÍSTICAS

(Ver carpeta Pruebas Estadísticas)