

CONTROL DE SISTEMAS PARALELOS INSPIRADO EN LA NATURALEZA

Carlos Iván Camargo – Universidad Nacional de Colombia
cicamargoba@unal.edu.co

Abstract—En la actualidad los dispositivos electrónicos hacen parte de nuestras actividades, a diario estamos en contacto con decenas de dispositivos digitales, ellos se encuentran integrados a aparatos de uso común como electrodomésticos, sistemas de comunicaciones, sistemas de entretenimiento, etc. Esta “invasión” digital va en aumento gracias al gran desarrollo de la industria de los semiconductores y a la disponibilidad de herramientas de programación. En un futuro no muy lejano, millones de dispositivos serán integrados en calles, puentes, y en nuestras propias casas, estos dispositivos formarán redes intercomunicadas que permitirán supervisar y controlar variables interesantes como tráfico, variables ambientales, identificación de personas. Sin embargo, es necesario dar respuestas a nuevas preguntas y afrontar nuevos retos que se presentan en el momento de diseñar este tipo de sistemas. La naturaleza es una fuente de inspiración para resolver este tipo de problemas ya que podemos encontrar ejemplos de sistemas con millones de componentes que interactúan para cumplir tareas que benefician al sistema como un todo. Un buen ejemplo de sistemas masivamente paralelos lo encontramos en los insectos sociales, vemos como millones de hormigas o termitas trabajan coordinadamente para suplir las necesidades de sus colonias, este tipo de sociedades no poseen elementos de coordinación, ni existen individuos especializados para dicho fin, el comportamiento emerge a partir de interacciones locales. El sistema inmune de los mamíferos es otro buen ejemplo de sistema masivamente distribuido en el que sus componentes actúan de forma coordinada para luchar contra agentes externos. En este trabajo se presenta al sistema inmune artificial como elemento de control de un sistema distribuido, cuyos individuos son robots móviles.

Index Terms—Sistemas Bio-Inspirados, Sistemas Multi-Robot, Sistemas Embebidos.

I. INTRODUCCIÓN

En la actualidad existe un número creciente de investigadores que ven a la naturaleza como fuente de inspiración para realizar sus diseños (por ejemplo: aplicaciones inspiradas en, los sistemas inmunes artificiales, algoritmos genéticos, autómatas celulares). La naturaleza ha creado sistemas muy complejos que están formados por millones de elementos, los cuales ejecutan sus funciones en paralelo, cada elemento realiza una determinada tarea para lograr un objetivo común. Los seres vivos cuentan con mecanismos que supervisan a diferentes niveles, el estado y funcionamiento de sus componentes; si se detecta una anomalía, estos mecanismos son capaces de reparar componentes defectuosos (regeneración) o de iniciar procesos de eliminación de elementos nocivos [1]; ante la presencia de un organismo externo el Sistema Inmune inicia un proceso de eliminación, el cual consiste en la especialización de ciertas células para la detección del agente nocivo, este proceso se realiza de forma paralela

y distribuida, lo cual mejora el tiempo de respuesta. Estas células especializadas finalmente generan mecanismos para la eliminación de dichos elementos. Los diseñadores de sistemas computacionales encuentran especialmente atractivas las propiedades de auto-replicación, auto-reparación y asociación; en la actualidad existen varias líneas de investigación cuyo objetivo es crear metodologías de diseño que busquen trasladar estas propiedades al campo de los sistemas digitales y computacionales [2], [3], [4], [5], [6].

Por otro lado, es muy interesante la forma en que comportamientos complejos emergen de la interacción local de millones de elementos sencillos. Ejemplos de este tipo se observan a diario en la naturaleza: los cardúmenes de peces o las bandadas de aves. Existen muchas personas que creen que el comportamiento de las aves o los peces obedecen al seguimiento de líderes, esto debido a la forma de pensar centralizada de los seres humanos, sin embargo, el comportamiento global depende de la reacción individual a acciones de sus vecinos inmediatos. Otro ejemplo se puede observar en el comportamiento de las colonias de insectos, cada individuo realiza una función específica para contribuir a un objetivo general: la supervivencia y la conservación de la especie; a partir de interacciones locales estas colonias crean estructuras complejas, las cuales no podrían ser realizadas por un solo individuo. La respuesta a la pregunta ¿Cómo emerge un determinado comportamiento global a partir de interacciones locales? ha sido la inspiración de un gran número de investigaciones [7], [8], [9], [10]. Sin embargo, esta pregunta no ha sido completamente respondida y existe un largo camino por recorrer hasta que seamos capaces de formular reglas locales que permitan obtener de forma colectiva la solución a cualquier problema complejo.

Existe un considerable número de trabajos en los que se explotan las características distribuidas de los sistemas inmunes artificiales (AIS) para controlar un gran número de robots en tareas de búsqueda y rescate, y detección de minas [11], [12], [7], [13]; otros investigadores comienzan a estudiar la forma de explotar las capacidades de los AIS en el entorno de computación ubicua [14], [15], por otro lado, la utilización de los AIS como medio para obtener tolerancia a fallos en sistemas digitales, ya está siendo estudiada [16], [17], [18]. Esta gama de aplicaciones de los AIS hace pensar en la posibilidad de su utilización para formular un nuevo paradigma computacional o para complementar los ya existentes, por ejemplo, adicionando la capacidad de detección y remoción de fallas. Sin embargo, existen muchas consideraciones que deben tomarse en cuenta en el desarrollo de estos sistemas

[19] antes de obtener un progreso significativo en la utilización de los AIS como medio para obtener la tolerancia a fallos.

Ejemplos como los anteriores evidencian el creciente interés en el estudio y desarrollo de sistemas que utilizan metáforas Biológicas en su funcionamiento; tal vez una de las características más importantes de estos sistemas bioinspirados (desde el punto de vista electrónico) es su capacidad intrínseca de adaptación a entornos cambiantes, lo cual les permite exhibir propiedades de tolerancia a fallos. El campo de aplicación de estos sistemas no se limita solo al campo de la electrónica, sino que se extiende a un gran número de actividades humana, como por ejemplo: actividades económicas, políticas comerciales. Por otro lado, proporcionan una fuente de inspiración y una motivación para trabajar con sistemas distribuidos masivos. Sin embargo, para poder ser capaces de crear dispositivos que se comporten de forma similar, primero debemos cambiar la forma de ver el mundo y cambiar nuestra tendencia al centralismo y entrar a una era descentralizada. Sin embargo, el interés en la descentralización no es nuevo; las ideas de la descentralización se expanden hoy más rápidamente y penetran más profundamente en diferentes dominios de las actividades humanas: organizaciones, tecnologías, modelos científicos y teorías del conocimiento. En el caso de las tecnologías actuales como los MEMs (Micro-Electro Mechanical devices) es posible la construcción de millones de dispositivos computacionales equipados con sensores y actuadores y su integración en materiales, estructuras y entornos [20], [21]. Se estima que en el futuro aumenten las aplicaciones de estas tecnologías, por lo que su programación será un reto constante [22]; al igual que entender cómo diseñar y programar grandes sistemas descentralizados.

Sin embargo, para estar en condiciones de crear sistemas digitales inspirados en las *sociedades naturales* es necesario dar respuestas a nuevas preguntas y afrontar nuevos retos que se presentan en el momento de diseñar este tipo de sistemas. A continuación se enumeran los principales problemas a los que nos debemos enfrentar:

- **Diseño:** ¿Qué técnicas y métodos permiten el diseño y control de estos sistemas?
- **Organización:** ¿Cómo pueden reorganizarse y coordinar su comportamiento?
- **Uso:** ¿Qué usos se darán a estos sistemas?
- **Adaptación:** ¿Cómo conforman funciones colectivas y como las memorizan?
- **Desarrollo:** ¿Qué tipos de protocolos de interacción y comunicación, que modos de organización podrán crear sistemas seguros, confiables y flexibles?
- **Validación:** ¿Cómo evaluar la validez de estos sistemas?

Para dar respuesta a estas preguntas realizaremos un estudio basado en *Colonias de Robots*, las cuales constituyen un entorno de trabajo adecuado para explorar las diferentes variables que intervienen en el problema; en el presente trabajo no nos limitaremos solo a obtener resultados experimentales sino que estos resultados serán llevados al mundo físico con robots construidos por nosotros, esto se hace con el fin de adoptar nuevas metodologías de diseño de sistemas digitales y evaluar la validez de las soluciones propuestas.

La robótica Móvil representa un campo de investigación que crece rápidamente. En la actualidad existe un gran número

de grupos de investigación que trabajan en el desarrollo de técnicas de auto-organización para sistemas Multi-Robot, entre los que se encuentran: El proyecto Swarm-bots [23], [24], Interaction Labs (USC) [25] [26], Autonomous System Lab (EPFL), MIT Computer Science and Artificial Intelligence Laboratory [27]. De las experiencias obtenidas de estos proyectos, se deduce, que para desarrollar algoritmos de inteligencia artificial, es necesario contar con plataformas Hardware y Software que permitan validar los algoritmos y modelos computacionales propuestos. La característica común de los proyectos ya mencionados es la implementación física de los algoritmos sobre robots reales, lo cual separa estos trabajos de investigaciones similares en el área de los Sistemas Multi-Agente, los cuales son desarrollados en plataformas Software. En la primera parte de este artículo se describirá la plataforma Hardware utilizada para realizar las pruebas, en la segunda se explicarán las bases del modelo utilizado.

II. ECBOT: PLATAFORMA ABIERTA PARA ROBÓTICA BASADA EN LINUX

ECBOT está formado por la unión de dos grandes componentes: El Componente Hardware y el Componente Software, el primero constituye el dispositivo físico mediante el cual se interactúa con el entorno y sobre el cual el componente Software implementa un determinado algoritmo.

A. Arquitectura Software

La columna vertebral del componente Software es el proyecto Player/Stage, el cual, es el resultado de un proyecto de investigación del Grupo de Investigación en Robótica de la University of Southern California. Este proyecto esta dividido en tres partes:

- 1) **Player** Es un servidor que proporciona una interfaz flexible a una gran variedad de sensores y actuadores. Como puede verse en la Figura 1 utiliza un modelo cliente/servidor basado en sockets TCP, su principal característica es que permite el manejo de los sensores y actuadores a través de una interfaz de programación de alto nivel, lo cual permite que los programas de control del robot sean escritos en cualquier lenguaje y puedan ser ejecutados en cualquier plataforma¹ (cliente) que posea una conexión de red con el robot (servidor). Además, Player soporta múltiples conexiones concurrentes de clientes, lo cual es muy útil en estudios de control descentralizado.
- 2) **Stage** Es un simulador escalable; que simula robots que se mueven y realizan operaciones de sensado en un entorno de dos dimensiones, estos robots son controlados por Player. Stage proporciona robots virtuales los cuales interactúan con dispositivos simulados. En la Figura 2 se puede observar una simulación multi-robot utilizando las librerías de Stage; una de las características más atractivas del proyecto Player-Stage es que permite la creación de sensores y actuadores que simulan el comportamiento de los dispositivos reales.

¹También es posible que el cliente y el servidor se ejecuten en el mismo robot.

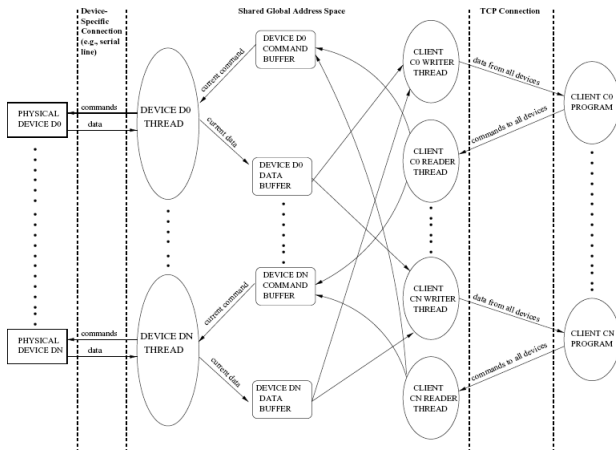


Fig. 1. Arquitectura de Player. [28]

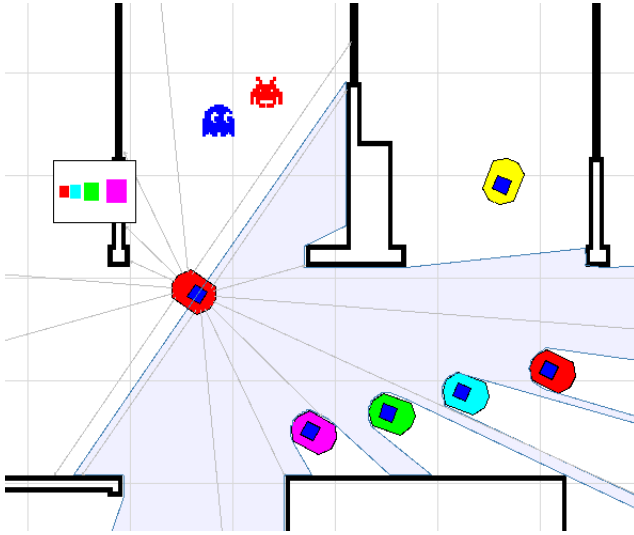


Fig. 2. Captura del simulador Stage.

1) *Arquitectura Hardware:* ECBOT [29] [30] [31] reúne la mayoría de las características de las plataformas comerciales disponibles ². Esta plataforma permite la ejecución de aplicaciones Linux [32] [33] y permite realizar de forma fácil la implementación de algoritmos de control. Gracias a su capacidad de comunicación inalámbrica no es necesario detener el robot y llevarlo a un sitio donde sea programado, ECBOT permite cambiar su configuración de forma remota. Este robot es el primero en su tipo diseñado totalmente en Colombia, tanto el HW como el SW fueron adaptados para él por ingenieros Colombianos, esto ayuda a disminuir la brecha tecnológica que existe en nuestro país, y permite que en nuestras instituciones académicas se utilice tecnología de punta, eliminando de esta forma tecnologías obsoletas utilizadas hasta el momento. En la figura 3 se muestra el Diagrama de Bloques de ECBOT,

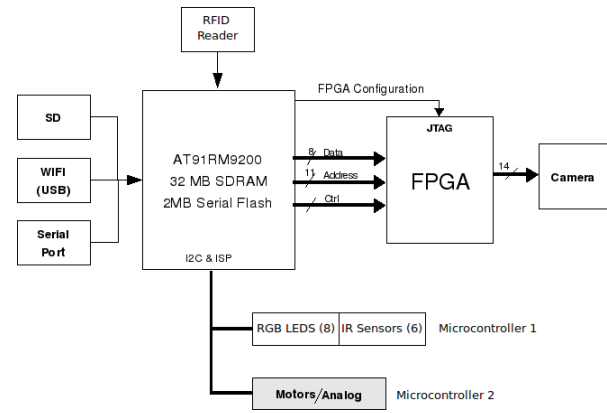


Fig. 3. Diagrama de Bloques del componente Hardware de ECBOT.

B. Especificaciones de ECBOT

ECBOT fué diseñado pensando en que pueda ser utilizado como una plataforma para el estudio de los sistemas Embebidos, y puede ser utilizado en cualquier campo, es ideal como herramienta didáctica ya que como puede verse en 3 permite implementar tareas HW en una FPGA, lo cual no se enseña en los programas académicos de la Universidades Colombianas. Adicionalmente permite la utilización de Linux como sistema operativo, lo cual facilita el desarrollo de aplicaciones gracias al gran número de librerías y aplicaciones disponibles. El principal aporte de este trabajo es la adopción de estas nuevas tecnologías y metodologías de diseño al entorno académico e industrial Colombiano, proporcionando de esta forma herramientas que permitan a nuestros profesionales competir de forma más eficiente con los productos producidos en el exterior. Por otro lado, este es el primer paso hacia la independencia tecnológica que nos ha convertido en importadores de tecnología, esta plataforma representa un avance de más de 30 años en el diseño de sistemas digitales, y gracias a su carácter libre puede ser utilizado por empresas e instituciones académicas como punto de partida en el estudio de los sistemas Embebidos modernos.

1) *Sensores y Actuadores:* ECBOT está diseñado de tal forma que se le pueda adaptar cualquier sensor que posea la interfaz I2C, y permite controlar hasta 10 servomotores, sin embargo, cuenta con una serie de sensores básicos que le permiten realizar las siguientes tareas:

- Detección de obstáculos: 6 sensores infrarojos de proximidad.
- Control de posición: Control de 2 motores DC utilizando PWM y BEMF.
- Sensor de Imágen: Permite capturar imágenes en formato subQCIF (129 x 96)
- Leds RGB: Permite cambiar el color del robot.

cada robot posee un detector de *blobs* de colores, lo cual permite la identificación de ciertos lugares del entorno (Almacenamiento de alimentos, Hogar) y el estado de otros robots.

2) *Comunicaciones:* Cada plataforma robótica puede comunicarse de forma directa con otros robots utilizando la interfaz de red inalámbrica, o de forma indirecta comunicando su estado interno a través de los LEDS RGB, cada color

²Roomba: <http://www.irobot.com>, Khepera: <http://www.k-team.com>, Pioneer: <http://www.activrobots.com/>

representa un estado interno, este estado puede ser detectado con el sensor de imagen. En el presente estudio se prefiere esta última forma de comunicación ya que si observamos el comportamiento de los sistemas naturales y biológicos ellos no utilizan comunicación directa. Adicionalmente cada robot cuenta con un sensor RFID que permite de forma indirecta comunicarse con otros robots de la misma forma que lo hacen las feromonas en la naturaleza.

En la Figura 4 se muestra una fotografía de la tarjeta base de la plataforma robótica, en la que podemos observar la localización de los componentes, esta arquitectura está fuertemente influenciada por el robot *e-puck* [34] del EPFL.

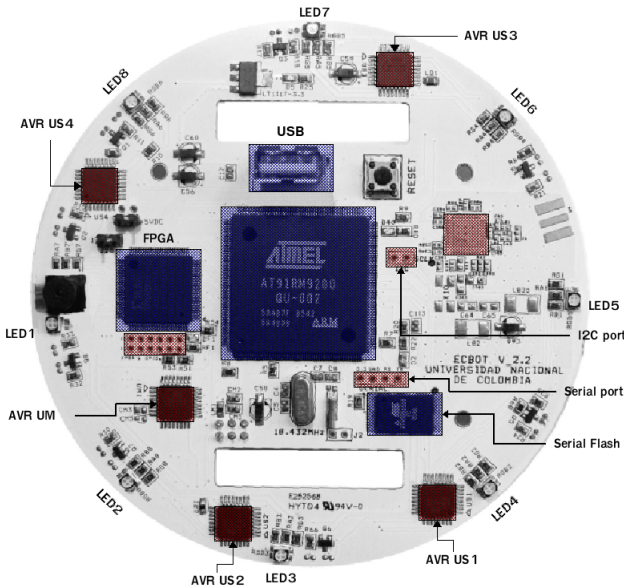


Fig. 4. Lado de Componentes de la tarjeta principal de ECBOT.

III. ALGORITMO DE CONTROL

El trabajo aquí presentado tiene dos fuentes de inspiración tomadas de la naturaleza: El sistema inmune de los mamíferos y las feromonas utilizadas por las hormigas como medio de comunicación.

A. Sistema Inmune Artificial

Especificidad, inducibilidad, diversidad, memoria, distinción entre propio y no propio (non self), autoregulación; son las principales características del sistema inmune humano, y las que permiten que utilizarlo en la solución de problemas en ingeniería.

El marco de control organiza un grupo de agentes en un entorno dinámico, se desarrolla con base en el mecanismo distribuido de la inmunidad biológica. El Sistema Inmune (SI), es un tipo especial de sistema multi-agente donde cada elemento inmune tiene un patrón de comportamiento y funciones específicas para un antígeno particular. El comportamiento de dicho agente, depende del entorno, así como del comportamiento individual; además posee un grupo de capacidades específicas que determinan su inteligencia fundamental. Esta

inteligencia puede ser alcanzada a través de comunicación entre agentes o a través de exploraciones del entorno.

Se adoptó una red de control distribuida basada en el comportamiento, la cual se basa en la transición de estado de comportamiento de los agentes. [35] Propone un nivel de descripción, instanciado en el llamado “comportamiento básico”, bloques constructores para síntesis y análisis de funcionamiento complejo de grupo en un sistema multi-agente. La Biología proporciona evidencia de soporte de unidades de funcionamiento básico en una variedad de niveles. Controlar una pierna de una rana, o un brazo humano, es una tarea compleja, especialmente si se realiza a bajo nivel. Con el fin de reducir la complejidad, la naturaleza impone una abstracción. Musaa - Ivaldi & Giszter [36] muestran que un grupo relativamente pequeño de campos vectoriales básicos, encontrados en la espina de la rana, genera el repertorio completo del comportamiento motriz, aplicando combinaciones apropiadas de los vectores básicos. Tomando esto como idea, se definen funcionamientos como leyes de control que encapsulan grupos de restricciones, de tal forma que se logran metas particulares. Los funcionamientos básicos se definen como un grupo mínimo de tales comportamientos, con propiedades apropiadas, que toman ventaja de la dinámica de el sistema dado, para cumplir de forma efectiva con su repertorio de tareas.

Los comportamientos básicos son una herramienta que pretenden describir, especificar y predecir el funcionamiento del grupo. Si se seleccionan de forma adecuada estos pueden generar comportamientos grupales repetibles y predecibles. La idea de los comportamientos básicos es general: Ellos pretenden ser primitivas para estructurar, sintetizar y analizar el funcionamiento del sistema, así como, bloques constructores para control, planeamiento y aprendizaje. Están relacionados con los atractores dinámicos, estados de equilibrio y otros términos utilizados para describir comportamientos estables, repetibles y primitivos de cualquier sistema [37].

1) *Comportamientos Básicos Para Movimiento en un Plano*: Los comportamientos básicos propuestos en [35] definen grupos de comportamientos sin especificar reglas particulares para implementarlas. El comportamiento de grupo en el dominio espacial puede verse como patrones espacio-temporales de la actividad de los agentes. Ciertas organizaciones de agentes espacialmente fijas son relevantes, así como ciertos patrones espacio-temporales. Las organizaciones de agentes espacialmente fijas corresponden a logros de metas, mientras los patrones espacio-temporales corresponden al mantenimiento de las mismas. La Tabla I muestra una lista de comportamientos que constituyen un grupo básico para un repertorio flexible de interacciones de grupo.

2) *Combinación de Comportamientos Básicos*: Los comportamientos básicos están diseñados para ser un sustrato para una variedad de comportamientos de grupo más complejos para un determinado dominio. Generar funcionamientos complejos requiere aplicar algún tipo de operadores de combinación cuyas propiedades sean bien conocidas y produzcan el funcionamiento compuesto deseado. Este es uno de los retos del control *Basado en el Comportamiento*, esto es, coordinar la actividad de múltiples comportamientos de entrada para

Safe Wandering	La habilidad de un grupo de agentes para moverse alrededor, mientras evitan colisiones entre ellos y otros obstáculos. La naturaleza homogénea de los agentes puede utilizarse para evitar colisiones entre agentes. Esto es, se pueden divisar dos estrategias diferentes; una para evitar colisiones entre agentes del mismo tipo, y otra para evitar colisiones con todo lo demás.
Following	La habilidad de dos o más agentes para moverse mientras se mantiene uno detrás del otro.
Dispersion	La habilidad de un grupo de agentes para dispersarse sobre un área con el fin de establecer y mantener una distancia mínima predeterminada.
Aggregation	La habilidad de un grupo de agentes de reunirse con el fin de establecer y mantener una distancia máxima predeterminada.
Homming	La habilidad para alcanzar una región o lugar específico.

TABLE I

GRUPO DE COMPORTAMIENTOS BÁSICOS PARA EL DOMINIO ESPACIAL

producir el comportamiento de salida deseado. En la figura 5 se muestran las combinaciones necesarias para obtener los comportamientos *herding* y *surrounding*

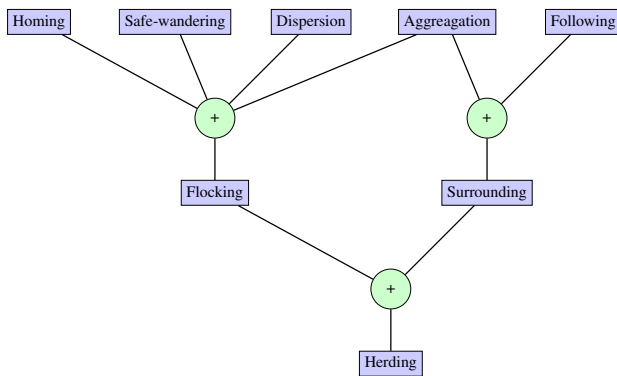


Fig. 5. Ejemplo de combinación de comportamientos básicos

La operación de los agentes no está predeterminada, pero se altera dinámicamente con el fin de adaptarse a un entorno de trabajo dinámico. Este control distribuido y no-determinístico hereda las siguientes funciones del sistema inmune humano:

- 1) **Auto-organización** Los agentes tienen la capacidad única de determinar respuestas para alcanzar objetivos comunes a través de toma de decisiones independientes y utilizando comunicaciones.
- 2) **Reconocimiento de Self/Non-self** Los agentes identifican tareas durante su exploración aleatoria; para evaluar la estimulación de una tarea específica se utiliza la función de afinidad (un índice para identificación non-self). Este reconocimiento depende únicamente de la estimulación proporcionada por la tarea, los agentes no poseen información previa sobre su localización o complejidad. Cada agente posee un set predefinido de capacidades, lo cual los hacen extremadamente flexibles para abordar diferentes tipos de problemas.
- 3) **Adaptabilidad** Los agentes ajustan sus comportamientos al atacar un problema manipulando la mejor respuesta, evaluando la especificidad de sus capacidades.

Se adquieren nuevos conocimientos o funcionalidades cuando se enfrentan a variaciones de tareas y entorno.

- 4) **Robustez** La falla de un agente, mientras ejecuta una operación no paralizará el sistema; ya que el marco de control está encaminado a ser totalmente descentralizado y no se asignan dependencias fijas entre tareas y agentes.
- 5) **Diversidad** El sistema es capaz de aprender de la experiencia, manipulando sus capacidades y realizando su inteligencia fundamental para resolver diferentes problemas.

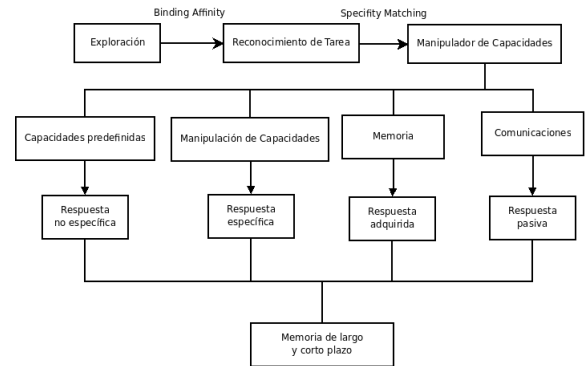


Fig. 6. Flujo de control de un agente del sistema inmune

El marco de control proporciona un grupo de reglas que guían y determinan el comportamiento de un agente en respuesta a un entorno dinámico. A través de la manipulación de las reglas se pueden investigar de forma efectiva eventos desconocidos y variaciones dinámicas del sitio de trabajo. La figura 6 muestra el flujo de control de un agente SI. Primero realizan una exploración del entorno circundante dentro de su rango sensorial (SR_S). Basándose en la medición de afinidad, los agentes reconocen y se enfocan en una tarea. Una vez reconocida una tarea, el agente manipula sus capacidades con una función que recibe el nombre de *matching específico* esto permite que el agente realice respuestas apropiadas al abordar diferentes tareas. Se definen cuatro tipos de respuestas inspiradas en el sistema inmune, ellas son: No específicas, específicas, adquiridas y pasivas. La respuesta no específica se encarga de tareas generales, los agentes ejecutan capacidades predefinidas cuando abordan este tipo de tareas; para tareas más complicadas, los agentes requieren la generación de nuevas capacidades específicas a estas tareas, los agentes modifican sus capacidades fundamentales para abordar este tipo de tareas, estas nuevas capacidades son almacenadas en memoria en la forma de respuesta adquirida. La respuesta pasiva se presenta en tareas que requieran cooperación de más de un agente, es este caso, el primer agente que solicita ayuda es el único que puede dar instrucciones a los otros agentes, los cuales exhiben un comportamiento pasivo.

Los cambios de estado son gobernados por un grupo de reglas evento-condición-acción, dichas reglas permiten que los agentes SIA realicen respuestas bajo diferentes situaciones; de acuerdo con estas reglas, los agentes obtienen información a través de comunicación entre ellos, esta información incluye localización de tareas, señales de auxilio, y estados de com-

portamiento de otros agentes; toda esta información permite a los agentes coordinar planes de acción para diferentes tipos de tareas. La Figura 7 muestra el árbol de decisiones que define las acciones de los agentes SIA, este árbol muestra como los agentes se adaptan a cambios en el entorno y se comportan de forma autónoma al abordar las tareas.

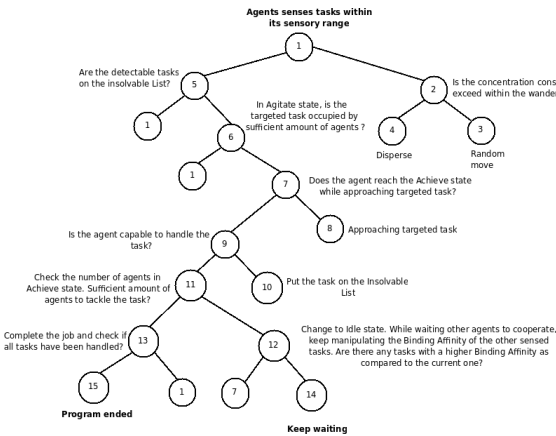


Fig. 7. Árbol de decisiones que define las acciones de un agente SIA [38]

B. feromonas Artificiales

Uno de los principales problemas al trabajar con sistemas masivamente distribuidos es la comunicación entre los individuos, permitir la comunicación directa entre los miembros del sistema y difundir información del entorno a todos los integrantes hace que el proceso de comunicación requiera mucha capacidad de almacenamiento y procesamiento, lo cual dificulta escalar los resultados obtenidos en una forma directa. Por esta razón, se prefiere la comunicación local [39], en la que cada individuo es capaz de comunicarse con los elementos más cercanos a él. Otro problema que surge al tratar de llevar a la práctica los experimentos de sistemas multi-robot, es que la mayoría de los simuladores asume que la posición del robot es conocida en todo momento y que cada robot conoce la localización de ciertos sitios importantes, esto en la práctica es difícil de implementar. Algunos trabajos reportan algoritmos que permiten la creación de sistemas de coordenadas de forma dinámica [40], sin embargo, solo tienen validez cuando los integrantes del sistema se encuentran estacionarios.

Algunos insectos sociales como las hormigas, muestran un gran desempeño en diversas actividades [41]. No existe un control central en estas comunidades. Son capaces de determinar la localización de comida con la ayuda de sustancias químicas llamadas feromonas, la feromona, es dejada en la superficie de la tierra para una movilización eficiente. Las hormigas construyen caminos utilizando feromonas para formar redes que conectan su nido con las fuentes de comida. El primer camino creado no es el óptimo, sin embargo, los caminos más cortos recibirán mayor concentración de feromonas (debido a que el tiempo requerido para recorrerlo es menor) y en los más largos la concentración de feromonas se reducirá. [42] Esto es muy interesante desde el punto de vista de comunicación multi-agente ya que un agente puede

dejar múltiples mensajes en el entorno. Este tipo de sistema de comunicación indirecta no solo omite la comunicación centralizada, sino que establece una relación entre información y posición en el entorno. [43]. Este fenómeno ha sido estudiado en varios campos de investigación [44] [45] [46] [47]. Debido a que las feromonas son sustancias químicas, son difíciles de utilizar en aplicaciones reales.

Algunos investigadores han utilizado el concepto de feromona virtual para imitar las características de las feromonas, [46] [48] [49]. En [50] y [42] implementan sistemas físicos que se aproximan a las características que se desean extraer de las feromonas utilizando la tecnología RFID. Esta tecnología utiliza variaciones de campo magnético para la comunicación; posee 2 componentes principales, los *tags* y los *readers*. Un *tag* es un dispositivo de identificación que posee un identificador único y una pequeña cantidad de memoria disponible. Un *reader* es un dispositivo que puede reconocer la presencia de un *tag* y hacer operaciones de lectura y escritura sobre él.

Utilizando una red de *tags* distribuidos a lo largo del espacio ocupado por los robots, es posible imitar el comportamiento de las feromonas de las hormigas y de esta forma definir caminos que sean útiles para la ejecución de una determinada tarea. Adicionalmente, los robots pueden utilizarlos para marcar sitios de interés general.

IV. CONCLUSIONES Y TRABAJO FUTURO

V.

REFERENCES

- [1] P. Raven and G. Johnson. *Biology*, chapter 50 The Immune System. Mc Graw Hill.
- [2] F. Gruau. and P. Malbos. The blob: A basic topological concept for hardware-free distributed computation. In Cristian Calude, Michael J. Dinneen, and Ferdinand Peper, editors, *Unconventional Models of Computation, Third International Conference, UMC 2002, Kobe, Japan, October 15-19, 2002, Proceedings*, volume 2509 of *Lecture Notes in Computer Science*, pages 151–163. Springer, 2002.
- [3] R. Nagpal. Organizing a Global Coordinate System from Local Information on an Amorphous Computer. AI Memo No 1666 MIT, 1999.
- [4] M. Wooldridge and M. Jennings. *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, 1995.
- [5] R. L. King; A. B. Lambert; S. H. Russ; and D. S. Reese. The Biological Basis of the Immune System as a Model for Intelligent Agents. *IPDPS 1999 Workshop*, 1999.
- [6] Y. Thoma, D. Roggen, E. Sanchez, and J. Moreno. Prototyping with a bio-inspired reconfigurable chip. In *15th IEEE International Workshop on Rapid System Prototyping (RSP 2004)*. IEEE Computer Society, Los Alamitos, California, 2004, 2004.
- [7] S. Singh. S. Thayer. Kilorobot Search and Rescue Using an Immunologically Inspired Approach. In *Distributed Autonomous Robotic Systems*. Springer-Verlag, June 2002.
- [8] E. D'Hondt. Exploring the Amorphous Computing Paradigm. Technical report, VUB, Bruxelles, 2000.
- [9] B. Bonabeau C. Meyer. *Swarm Intelligence: A whole new way to think about business*. Harvard Business Review, 2001.
- [10] M. Resnick. *Turtles, Termites and Traffic Jams Explorations in Massively Parallel Microworlds*. The MIT Press Cambridge, Massachusetts, 1994.
- [11] F. Sahin S. Sathyanath. AISIMAM - An Artificial Immune System Based Intelligent Multi-Agent Model and its Application to a Mine Detection. *ICARIS 2002*, September 2002.
- [12] S. Singh S. Thayer. Development of an Immunology-Based Multi-Robot Coordination Algorithm for exploration and Mapping Domains. *IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

- [13] S. Thayer S. Singh. Immunology Directed Methods for Distributed Robotics: A Novel, Immunity-Based Architecture for Robust Control & Coordination. *Proceedings of SPIE: Mobile Robots XVI*, November 2001.
- [14] M. Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 1993.
- [15] P. Mohr, N. Ryan, and J. Timmis. Exploiting Immunological Properties for Ubiquitous Computing Systems. *ICARIS 2004*, September 2004.
- [16] A. Avizienis and R. Avizienis. An Immune system paradigm for the design of fault tolerance systems. *Workshop 3: Evaluating and Architecting Systems for Dependability (EASY)*, 2001.
- [17] A. Avizienis. Toward systematic design of fault tolerant systems. *Computer*, 1997.
- [18] D. Bradley and A. Tyrrell. A Hardware Immune System for Benchmark State Machine Error Detection. In *Congress on Evolutionary Computation*, 2002.
- [19] J. Timmis, M. Ayara, and R. Duncan. Towards Immune Inspired Fault Tolerance In Embedded Systems. In *Proceedings of 9th International Conference on Neural Information Processing*. IEEE, 2002.
- [20] W. Buttera and V. Bove. Literally embedded processors. In *SPIE Media Processors*, 2001.
- [21] A. Berlin. *Towards Intelligent Structures: Active Control of Bucling*. PhD thesis, MIT. Department of Electrical Engineering and Computer Science, May 1994.
- [22] R. Nagpal. *Programmeable Self-Assembly: COConstructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics*. PhD thesis, MIT. Department of Electrical Engineering and Computer Science., June 2001.
- [23] F. Mondada, G. C. Pettinaro, I. Kwee, A. Guignard, L. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. SWARM-BOT: A swarm of autonomous mobile robots with self-assembling capabilities. In C.K. Hemelrijk and E. Bonabeau, editors, *Proceedings of the International Workshop on Self-organisation and Evolution of Social Behaviour*, pages 307–312, Monte Verità, Ascona, Switzerland, September 8-13, 2002. University of Zurich.
- [24] G. Baldassarre, S. Nolfi, and D. Parisi. Evolving mobile robots able to display collective behaviours. In C.K. Hemelrijk and E. Bonabeau, editors, *Proceedings of the International Workshop on Self-Organisation and Evolution of Social Behaviour*, pages 11–22, Monte Verità, Ascona, Switzerland, September 8-13, 2002. University of Zurich.
- [25] C. Jones and M. Mataric. Adaptive Division of Labor in Large-Scale Minimalist Multi-Robot Systems. *Proceedings of the IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS)*. Las Vegas, Nevada, 2003.
- [26] Jones and Maja J Mataric. *Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications*, chapter Behavior-Based Coordination in Multi-Robot Systems. Marcel Dekker, Inc, 2005.
- [27] J. McLurkin. Speaking Swarmish. *AAAI Spring Symposium*, 2006.
- [28] B. Gerkey, K. Stoy, and R. T. Vaughan. Player Robot Server. Technical report, USC Robotics Labs, 22 November 2000.
- [29] C. Camargo. Implementación de Sistemas Digitales Complejos Utilizando Sistemas Embebidos. *Memorias del XI Workshop de Iberchip ISBN 959-261-105-X*, 2005.
- [30] C. Camargo. ECBOT y ECBAT91 Plataformas Abiertas para el Diseño de Sistemas Embebidos y Co-Diseño HW/SW. *VIII Jornadas de Computación Reconfigurable y Aplicaciones, Madrid España*, 18 September 2008.
- [31] C. Camargo. ECBOT: Arquitectura Abierta para Robots Móviles. *Séptima Conferencia Iberoamericana en Sistemas, Cibernética e Informática, Miami, USA*, 2 July 2008.
- [32] C. Camargo. Linux como Herramienta de Desarrollo de Sistemas Embebidos. *XII Taller IBERCHIP, San José, Costa Rica*, March 2006.
- [33] C. Camargo. Linux como Plataforma de Desarrollo de Sistemas Embebidos. *Colombian Workshop on Circuits and Systems Bogotá, Colombia*, November 2007.
- [34] EPFL. e-puck EPFL Education robot. <http://www.e-puck.org/>.
- [35] Maja J Mataric. *Interaction and Intelligent Behavior*. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, May 1994.
- [36] Mussa-Ivaldi F. A. and S. Giszter. Vector field approximation: a computational paradigm for motor control and learning. *Biological Cybernetics* 67, 1992.
- [37] M. Mataric. Integration of Representation Into Goal-Driven Behavior-Based Robots. *IEEE Transactions on Robotics and Automation* 8(3), 1992.
- [38] Henry Y. K. Lau and Vicky W. K. Wong. An Immunity Based Distributed Multiagent-Control Framework. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, January 2006.
- [39] D. Coore H. Abelson, D. Allen. Amorphous Computing. <http://www.ai.mit.edu/publications/pubsDB/pubs.html>, 29 August 1999.
- [40] D. Coore. Establishing a Coordinate System on an Amorphous Computer. <http://swiss.ai.mit.edu/projects/amorphous>, 1997.
- [41] B. Holldobler and E. Wilson. *The Ants*. Springer-Verlag, 1990.
- [42] Herianto, T. Sakakibara, and D. Kurabayashi. Artificial Pheromone System Using RFID for Navigation of Autonomous Robots. *Journal of Bionic Engineering, Elsevier Limited and Science Press*, 2007.
- [43] D. Kurabayashi. Toward Realization of Collective Intelligence and Emergent Robotics. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 1999.
- [44] Edelstein-Keshet L. imple model for trail-following behaviour; trunk trails versus individual foragers. *Journal of Mathematical Biology*, 1994.
- [45] Payton D, Daily M, Estowski R, Howard M, and Lee C. Pheromone robotics. *Autonomous Robots*, 2001.
- [46] Sugawara K, Kazama T, and Watanabe T. Foraging behavior of interacting robots with virtual pheromone. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
- [47] Panurak H V D. Go to the ant. *Engineering principles for natural multi-agent systems. Annals of Operations Research*, 1997.
- [48] Tanaka K, Yo Ishigaki, Inoue H, and Itoh M. Safety monitoring system by autonomous mobile sensors utilizing pheromone communication. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation; International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 2005.
- [49] Mamei M, Quagliari R, and Zambonelli F. Making tuple space physical with RFID tags. *Proceedings of ACM Symposium on Applied Computing*, 2006.
- [50] Mamei M and Zambonelli F. Physical deployment of digital pheromones through RFID technology. *Swarm Intelligence Symposium*, 2005.



Carlos Camargo es Ingeniero Electricista de la Universidad Nacional de Colombia, Magister en Ingeniería Eléctrica de la Universidad de los Andes y Candidato a Doctor en Ingeniería Eléctrica en la Universidad Nacional de Colombia. Es profesor del Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia en el área de los sistemas Digitales.