



Tecnura

ISSN: 0123-921X

tecnura@udistrital.edu.co

Universidad Distrital Francisco José de  
Caldas  
Colombia

Vargas Torres, Germán Andrés; Castillo Estepa, Ricardo Andrés  
Desarrollo de algoritmo para detección y comando de robots humanoides en tareas de  
recolección

Tecnura, vol. 19, núm. 45, julio-septiembre, 2015, pp. 127-139

Universidad Distrital Francisco José de Caldas  
Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=257040047011>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



## Desarrollo de algoritmo para detección y comando de robots humanoides en tareas de recolección

### Development of an Algorithm for Humanoid Robot Detection & Command in Gathering Tasks

Germán Andrés Vargas Torres\*, Ricardo Andrés Castillo Estepa\*\*

**Fecha de recepción:** 21 de agosto de 2013

**Fecha de aceptación:** 5 de diciembre de 2014

**Como citar:** Vargas Torres, G. A., & Castillo Estepa, R. A. (2015). Desarrollo de algoritmo para detección y comando de robots humanoides en tareas de recolección. *Revista Tecnura*, 19(45), 127-139. doi:10.14483/udistrital.jour.tecnura.2015.3.a10

#### RESUMEN

Este artículo presenta un algoritmo para comandar un grupo de robots humanoides Bioid, organizándolos alrededor de un objeto de interés, detectado previamente por un sistema de visión externa. Los robots conforman un Sistema Multi-Agente (SMA) orientado hacia tareas de recolección cooperativa. Se detalla el desarrollo del SMA, así como cada componente del algoritmo de organización y la simulación en ambiente virtual. El algoritmo se subdivide en dos hilos dedicados: uno se encarga de visión de máquina (filtrado, detección de contornos y clasificación logrados a través de librerías de EmguCV) y cálculos en el espacio operacional, y el otro opera la comunicación inalámbrica ZigBee con los robots. Adicionalmente, los robots poseen su propio código embebido que les permite traducir una secuencia de instrucciones recibidas a patrones de marcha que los dirigen hacia el objeto de interés. El tiempo total de ejecución es tomado como la medida global de desempeño de interés.

**Palabras clave:** control de robots, robots humanoides, Sistemas Multi-Agente, sistemas de visión para robots, simulación por computador.

#### ABSTRACT

This article presents an algorithm which commands a group of Bioid humanoid robots in order to organize them around an object of interest, previously detected by an external vision system. The robots form a Multi-Agent System (MAS) oriented towards cooperative gathering tasks. Development of the MAS, as well as each of the organization algorithm's components and simulation inside a virtual environment, are all detailed. The algorithm is subdivided in two dedicated threads: one of which handles machine vision (filtering, contour detection and classification achieved through EmguCV libraries) and operational space calculations; while the other operates ZigBee wireless communication with the robots. Furthermore, the robots possess their own embedded code which enables them to translate a sequence of

\* Ingeniero en Mecatrónica. Asistente de investigación, Universidad Militar Nueva Granada, Bogotá, Colombia. Contacto: [u1800960@unimilitar.edu.co](mailto:u1800960@unimilitar.edu.co)

\*\* Ingeniero en Mecatrónica, magíster en Ingeniería Mecánica. Docente TC/Investigador Principal, Universidad Militar Nueva Granada, Bogotá, Colombia. Contacto: [ricardo.castillo@unimilitar.edu.co](mailto:ricardo.castillo@unimilitar.edu.co)

received instructions into gait patterns which allow them to move towards the object of interest. Total execution time for the gathering task is chosen as the global performance measure to evaluate.

**Keywords:** computer simulation, humanoid robots, Multi-Agent Systems, robot control, robot vision systems.

## INTRODUCCIÓN

El dominio de los Sistemas Multi-Agente (SMA) comprende todo sistema computarizado en el cual interactúan múltiples agentes racionales (Woolbridge, 2009). Un objetivo frecuente al hacer uso de Sistemas Multi-Agente en ingeniería es aprovechar la interacción entre los agentes para realizar tareas avanzadas mediante cooperación (Shoham & Leyton-Brown, 2007). Independiente del tipo de tarea por realizar, Cheng, Raja y Lesser (2013) indican que dicha cooperación se centra en optimizar una medida de desempeño global.

Los agentes de un SMA se caracterizan por poseer información y capacidades insuficientes para resolver el problema global. Puesto que los agentes “habitan” en un entorno complejo y dinámico, De Giacomo, Lespérance, Levesque y Sardina (2009) recalcan que los agentes establecen y ejecutan planes en tiempo real a fin de cumplir el conjunto de objetivos requeridos en la aplicación. En su caso, el desarrollo y uso del idioma de programación *IndiGolog* permite a los agentes planear mediante búsquedas destinadas a encontrar una serie de acciones que conformen una ejecución válida para el entorno estocástico (no determinístico).

Uno de los eventos que más ha gestado nuevos desarrollos en Sistemas Multi-Agente son las competencias RoboCup Soccer, tanto en el ámbito de simulación, donde se busca simular estrategias de juego basadas en inteligencia artificial (Ping-Huan & Li, 2011), como en los juegos con los robots humanoides físicos, donde se usan técnicas tales como Razonamiento Basado en Casos (Kuo & Lin, 2008) para generar estrategias y planes que se adapten al estado estocástico del juego. Mediante dicha técnica, los casos (ejecuciones) anteriores

son evaluados con base en la solución deseada a fin de extraer y retener las acciones/decisiones de mayor utilidad.

La ingeniería de sistemas ha adoptado en gran medida los SMA, dado que la aplicación de sus herramientas garantiza que el software final emulará un comportamiento organizacional. Por ejemplo, Fu-sheng, Hong-bo, Guang-ping, Yang y Xu-yan (2009) crearon un modelo de coordinación basado en agentes, el cual fue aplicado en la industria metalúrgica. Dicho modelo aplicó planeación de grupos parciales (PGP; Partial-Group Planning), lo cual le permite al sistema cambiar de centralización a descentralización y viceversa a voluntad. En otro ámbito, existe un gran interés por la obtención de comportamientos cooperativos en equipos conformados por distintos tipos de robots; Zhiguo, Junming, Xu, Zhiliang y Jun (2011) usan un modelo basado en un protocolo IGRS (Intelligent Grouping and Resource Sharing) para integrar un robot móvil (Charlie-II) y un robot tipo perro (Cailun-II) mediante el uso de un lenguaje estándar para efectos de distribución de recursos, servicios colaborativos y conformación de grupos transitorios.

Las capacidades de los robots humanoides en términos de hardware y software los convierten en los agentes más avanzados que pueden conformar un SMA. Uno de los principales obstáculos para su implementación radica en la generación de patrones de marcha, dada su complejidad desde la óptica de los sistemas dinámicos. Su resolución se ha abordado tradicionalmente emulando los patrones de marcha de los seres humanos (Malmberg, 2011), dadas las similitudes en términos de fases de marcha y fuerzas en actuación. Mukovskiy, Slotine y Giese (2010) destacan que dichos patrones deben modificarse cuando múltiples humanoides

se desplazan e interactúan en el ambiente de trabajo, a fin de garantizar la estabilidad de las formaciones que establezcan. Los investigadores abordaron las modificaciones bajo la óptica de la Teoría de Contracción, en la cual la estabilidad de un agente individual permite derivar condiciones que garanticen la estabilidad de los demás agentes con los cuales interactúe en un determinado escenario.

En el caso de los SMA aplicados a tareas de recolección, las limitaciones de los robots humanoides en términos de destreza conllevan la necesidad de adecuar los objetos por recolectar para facilitar su manipulación, tal como se aprecia en McGill y Lee (2011). Un conjunto de  $n$  humanoides manipulando un objeto adquieren un comportamiento dinámico diferente; dicha complejidad adicional puede ser resuelta a nivel global en los SMA centralizados y a nivel de cada agente en los SMA descentralizados. El segundo caso implica la necesidad de un carácter adaptativo, tal como Gu, Thobbi y Sheng (2011) demuestran mediante la aplicación de aprendizaje por refuerzo: el agente hace uso de Q-Learning para aprender una trayectoria correcta para sus brazos según distintos tipos de inclinaciones del objeto por levantar, con base en la recolección por parte de un instructor humano.

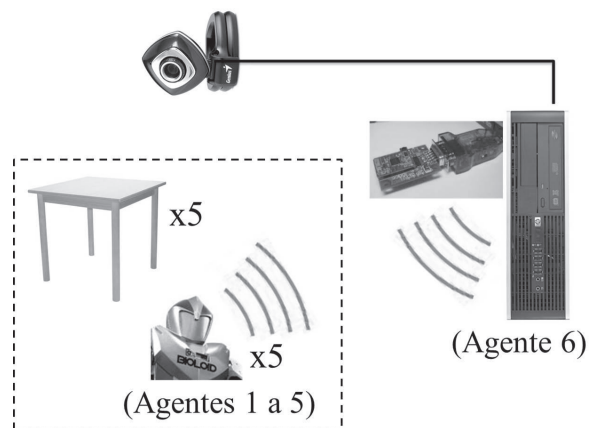
La ejecución de tareas de recolección con robots se beneficia significativamente de la colaboración entre agentes en dos instancias: las negociaciones son vitales para distribuir y asignar adecuadamente los agentes a objetos de interés y la coordinación es vital para evitar colisiones entre agentes y garantizar manipulación correcta de los objetos cuando su peso o sus dimensiones requieren más de un agente para su recolección.

En este caso, la medida global de desempeño de interés es el tiempo que toma a los agentes completar la recolección de todos los objetos que se distribuyen en el ambiente. Adicionalmente, es importante evaluar el funcionamiento en paralelo de los procesos de comunicación, visión y ejecución de caminatas y garantizar la inexistencia de interferencias entre estos.

El presente artículo se organiza de la siguiente manera: la Sección 2 detalla los diversos elementos que componen el Sistema Multi-Agente que se desarrolló para tareas de recolección. Acto seguido, la Sección 3 desglosa el algoritmo implementado con base en las tareas ejecutadas por cada uno de sus hilos (subprocesos), así como el código ejecutado por los robots humanoides Bioloid. Los resultados obtenidos para cada subproceso y su validación se presentan en la Sección 4, y, finalmente, la Sección 5 enuncia las conclusiones inferidas del algoritmo.

## DESARROLLO DEL SISTEMA MULTI-AGENTE

El presente Sistema Multi-Agente se desarrolló bajo el marco del proyecto de investigación Desarrollo de Estrategia con Aprendizaje On-Line para Comando de Grupo de Robots Humanoides Utilizando la Plataforma Robótica DARwIn-OP, perteneciente al Grupo de Investigación DAVINCI de la Universidad Militar Nueva Granada. Sus componentes de hardware se ilustran en la figura 1.



**Figura 1.** Componentes de Hardware en el SMA.

**Fuente:** elaboración propia.

El equipo de recolección está conformado por cinco robots humanoides Bioloid, los cuales interactúan con cinco mesas de diferentes

dimensiones, detalladas en la tabla 1. Los robots y las mesas se distribuyen en el entorno o espacio de trabajo; un área rectangular de 1,80 m por 1,35 m.

**Tabla 1.** Dimensiones de los objetos por recolectar

Tipo de Mesa N.º	Cantidad	Forma	Dimensiones (m)
1	2	Cuadrado	0,3 x 0,3
2	1	Rectángulo	0,3 x 0,1
3	1	Rectángulo	0,5 x 0,3
4	1	Triángulo Equilátero	0,35 por cada lado

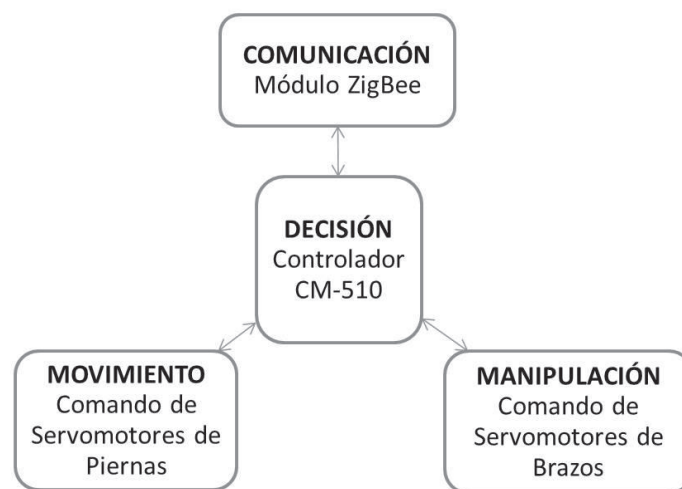
**Fuente:** elaboración propia.

Debido a que los robots Bioloid no poseen visión de máquina, se incorpora un sistema de visión externa mediante una webcam ubicada en el techo del laboratorio para proveer una vista de planta superior del ambiente de trabajo. El computador que procesa el *stream* de video obtenido de la cámara se convierte en un sexto agente, encargado de transferir los datos procesados a los robots vía comunicación inalámbrica ZigBee. Tal como

se resaltó en la figura 1, el sexto agente es el único que no se encuentra en el interior del espacio de trabajo y opera en una ubicación remota.

Dada la complejidad de los robots humanoides, se desarrolla una representación abstracta de la arquitectura del agente en términos de componentes lógicos y sus interrelaciones, tal como se ilustra en la figura 2. Cada bloque agrupa una serie de subsistemas presentes en el robot físico y permiten simplificar la integración de dichos subsistemas, tanto en el humanoide físico como en la simulación.

Como ejemplo, el bloque de movimiento incluye el conjunto de servomotores que accionan las articulaciones de las piernas en el robot humanoide y su control de bajo nivel. Cuando se recibe la orden de ejecutar un determinado patrón de caminata desde el bloque de decisión, el bloque de movimiento se encarga de transformar la orden en las secuencias de posiciones y velocidades angulares que cada servomotor debe ejecutar para generar el movimiento correcto del robot humanoide. Este SMA se caracteriza por poseer una arquitectura de comando heterárquica en el plano de actuación, en la cual cada uno de los cinco robots tiene el



**Figura 2.** Representación de agentes mediante componentes lógicos.

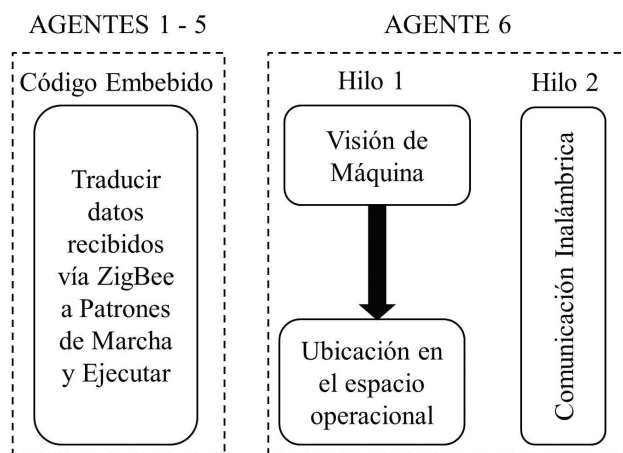
**Fuente:** elaboración propia.



mismo grado de autoridad al interactuar con un objeto en proceso de recolección. En el plano de decisión, el SMA clasifica como centralizado (Xuan & Lesser, 2002) dada la importancia del rol del sexto agente para el cumplimiento del objetivo global.

## ALGORITMO DE ORGANIZACIÓN

Dada la complejidad del Sistema Multi-Agente y la tarea por realizar, es de vital importancia evitar cuellos de botella en los diversos procesos que el algoritmo lleva a cabo que puedan conllevar pérdida/desactualización de información. Para este fin, el algoritmo se subdivide en una serie de componentes que se ejecutan en forma paralela gracias a la implementación de programación por hilos. Los robots humanoides se encuentran en continuo movimiento durante la ejecución del SMA, actualizando sus acciones en función de la información que reciben. Por ende, su código embebido actúa de forma paralela a los hilos del sexto agente. La figura 3 presenta la organización de cada proceso simultáneo.



**Figura 3.** Procesos ejecutados de forma paralela.

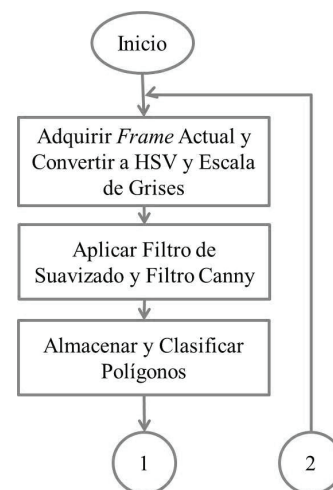
**Fuente:** elaboración propia.

A continuación se detallan la implementación de cada uno de estos procesos, así como la recreación del SMA mediante un Ambiente Virtual. Dado que un mecanismo deliberativo es vital para evitar conflictos

entre acciones paralelas, se realiza una distribución de tareas emergentes, bajo la cual el sistema toma en cuenta la cercanía de cada agente a los objetos de interés a fin de asignar las tareas correspondientes.

## Visión de máquina

El componente de visión está a cargo de extraer los elementos de interés (los objetos por recolectar y los robots humanoides) del *stream* de video entregado por la cámara. Se escoge implementar las librerías de EmguCV para simplificar el procesamiento de video, debido a su facilidad de uso y documentación disponible. Teniendo en cuenta que los robots humanoides Bioloid poseen un diseño homogéneo, se hace necesario dotarlos con marcadores triangulares de diversos colores que permitan al sistema distinguir la posición y orientación de cada uno en el siguiente componente del primer hilo. Asimismo, las mesas por recolectar poseen formas geométricas conocidas, por lo cual este segmento del código se enfoca hacia la extracción de figuras cuadradas y triangulares, lograda mediante la detección de contornos. El diagrama de flujo de este componente se presenta en la figura 4.



**Figura 4.** Diagrama de flujo para componente de visión.

**Fuente:** elaboración propia.

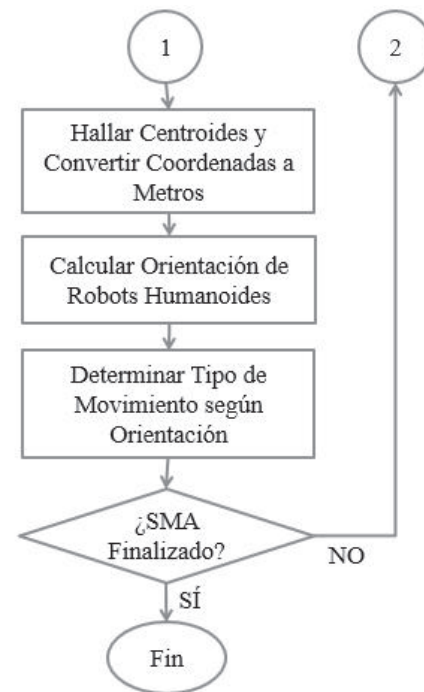
En primer lugar, la cámara captura 30 *frames* por segundo (fps) a una resolución de 640 por 480 píxeles. Al contrastar dicho valor con el tamaño del entorno de trabajo (1,80 m por 1,35 m), cada pixel tiene un tamaño equivalente a un cuadrado de 2,81 mm de lado, lo cual es una resolución elevada para la tarea por desarrollar. Acto seguido, el algoritmo crea dos copias del cuadro actual, las cuales se convierten a escala de grises y HSV respectivamente. La primera es la base para detectar contornos, mientras que la segunda permite clasificar a los cinco robots humanoides en la fase final del procesamiento. La imagen en escala de grises se somete a dos etapas de filtrado: un suavizado por descomposición de pirámide gaussiana, seguido por un filtro Canny.

Una vez obtenida la imagen posterior al filtrado Canny, los contornos cerrados detectados se asocian en polígonos. Se procede a almacenar los polígonos que cumplan las condiciones requeridas: tres caras para el caso de los triángulos o cuatro caras con ángulos cercanos a  $\pi/2$  radianes para el caso de los cuadriláteros. El sistema distingue la mesa triangular de los marcadores de los robots mediante evaluación del área del polígono. Finalmente, para cada uno de los triángulos almacenados, el sistema recurre a la imagen HSV, de la cual adquiere el valor de matiz (*Hue*) en las coordenadas del triángulo a fin de determinar su color, y por ende, el identificador de robot correspondiente (1 a 5).

## Ubicación en el espacio operacional

El componente de ubicación se centra en obtener la posición y orientación de los robots humanoides y las posiciones de las mesas, así como el cálculo del ángulo entre la orientación actual de cada robot y la orientación deseada (el robot observando en la dirección del objeto por recolectar más cercano). El diagrama de flujo correspondiente se muestra en la figura 5. El sistema toma las coordenadas del centroide para cada figura detectada en el componente de visión de máquina; dichas coordenadas

en píxeles pueden ser transformadas a coordenadas en metros con base en las relaciones expuestas en la ecuación (1). Es importante resaltar que el origen [0, 0] en la imagen corresponde a la esquina superior izquierda, contrario al origen del plano cartesiano que se sitúa en la esquina inferior izquierda; esto se refleja en el cálculo de la coordenada Y.



**Figura 5.** Diagrama de flujo para componente de ubicación.

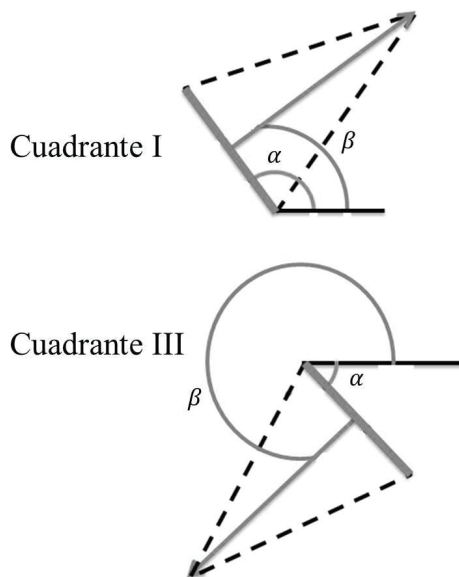
**Fuente:** elaboración propia.

$$\begin{aligned} X_m &= X_{\text{píxeles}} * 0,00281 & [m] \\ Y_m &= 1,35 - (Y_{\text{píxeles}} * 0,00281) & [m] \end{aligned} \quad (1)$$

Los marcadores para los robots humanoides son triángulos isósceles, ubicados con el vértice de menor ángulo apuntando hacia el frente del robot. Para calcular la orientación de los robots, el sistema detecta el lado del triángulo de menor longitud y halla su ángulo  $\alpha$  mediante la función  $\text{atan2}()$  (adicionalmente, el ángulo se convierte a  $\pi/2$  radianes automáticamente si se detecta que la línea es vertical). A continuación, el sistema evalúa la ubicación del vértice de menor ángulo

respecto al lado de menor longitud; dicho proceso permite determinar el cuadrante correcto y calcular la orientación verdadera  $\beta$ . Para el caso de  $\alpha = \pi/2 \text{ rad}$ , el vértice a la derecha de la línea significa  $\beta = 0 \text{ rad}$  y el vértice a la izquierda de la línea significa  $\beta = \pi \text{ rad}$ . Para un valor de  $\alpha \neq \pi/2 \text{ rad}$ , los cálculos se realizan con base a la ecuación (2); el signo de  $\alpha$  junto con la ubicación del vértice permite identificar el cuadrante. La figura 6 ilustra un ejemplo de este proceso para diferenciar los cuadrantes I y III.

$$\begin{aligned} \text{Cuadrante I:} & \quad \beta = |\alpha| - \pi/2 \\ \text{Cuadrante II:} & \quad \beta = |\alpha| + \pi/2 \\ \text{Cuadrante III:} & \quad \beta = 3\pi/2 - |\alpha| \\ \text{Cuadrante IV:} & \quad \beta = 3\pi/2 + |\alpha| \end{aligned} \quad (2)$$



**Figura 6.** Detección de la orientación  $\beta$ .

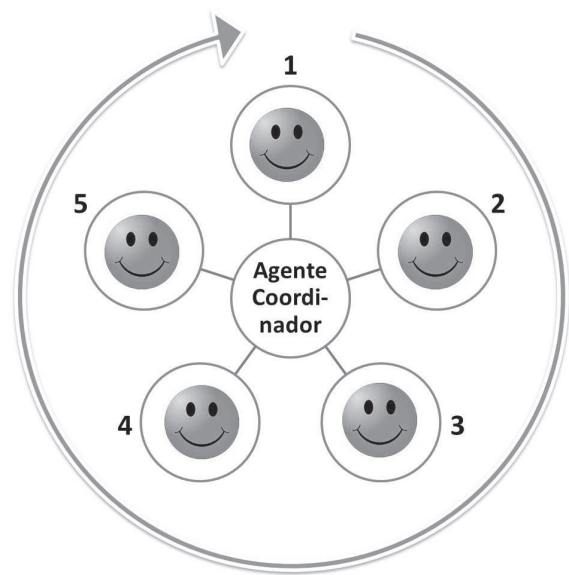
**Fuente:** elaboración propia.

Mediante cálculo de la distancia euclídea, se halla y asocia la mesa que se encuentre a mayor proximidad para cada uno de los robots humanoides. Por último, el sistema traza una línea entre los centroides del robot y la mesa; su ángulo es comparado con la orientación del robot para determinar la dirección en la cual el robot debe girar a fin

de alinear su curso de movimiento. Un resultado positivo indica que el robot debe girar en sentido horario, un resultado negativo indica que debe girar en sentido antihorario y un resultado dentro del margen de tolerancia  $[-\pi/12, \pi/12]$  radianes indica que debe avanzar en línea recta mientras su distancia al objetivo sea superior a 0,2 m.

## Comunicación inalámbrica

Este componente se encarga de entablar comunicación bilateral con todos los robots humanoides, otorgándoles los comandos de movimiento necesarios para aproximarlos a los objetos por recolectar. Se hace uso de las librerías para comunicación ZigBee creadas por el fabricante Robotis®, así como el adaptador de ZigBee a USB previamente ilustrado en la figura 1. A fin de organizar la comunicación entre los cinco humanoides y el sexto agente, se crea la topología tipo estrella de la figura 7; se asigna un identificador único a cada robot y el sistema realiza un bucle infinito, cumpliendo un ciclo de intercambio de información (envío-recepción) con un agente distinto en cada iteración.

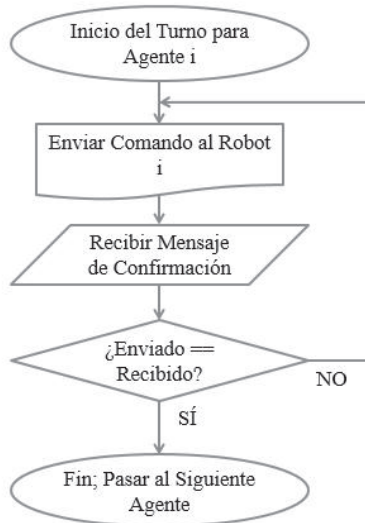


**Figura 7.** Topología para transporte de información.

**Fuente:** elaboración propia.



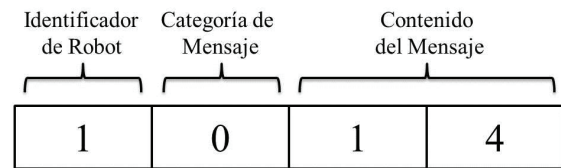
Se establece un mecanismo de “apretón de manos” (*handshake*) para garantizar que cada robot reciba la información correcta, en el cual este debe responder con el mismo dato que recibió. Si el sexto agente no recibe el dato equivalente, continuará enviando el dato original hasta confirmar su recepción, y así pasar al siguiente robot. Dicho proceso se presenta en la figura 8.



**Figura 8.** Mecanismo de “apretón de manos” (*handshake*).

**Fuente:** elaboración propia.

Puesto que los microcontroladores de los robots Bioloid poseen poder de procesamiento limitado, no es posible aplicar el paradigma de la programación orientada a objetos con los agentes, y por ende no se pueden implementar protocolos de comunicación FIPA. A fin de estandarizar los mensajes en el SMA, la información es enviada mediante una trama compuesta por un entero de cuatro dígitos, tal como se indica en la figura 9. El primer dígito identifica el robot destinatario (1 a 5), el segundo dígito corresponde a la categoría del mensaje ('0' indica que la categoría es “comando de movimiento”) y los dos últimos dígitos poseen el contenido del mensaje ('14' indica “realizar giro en sentido antihorario”).

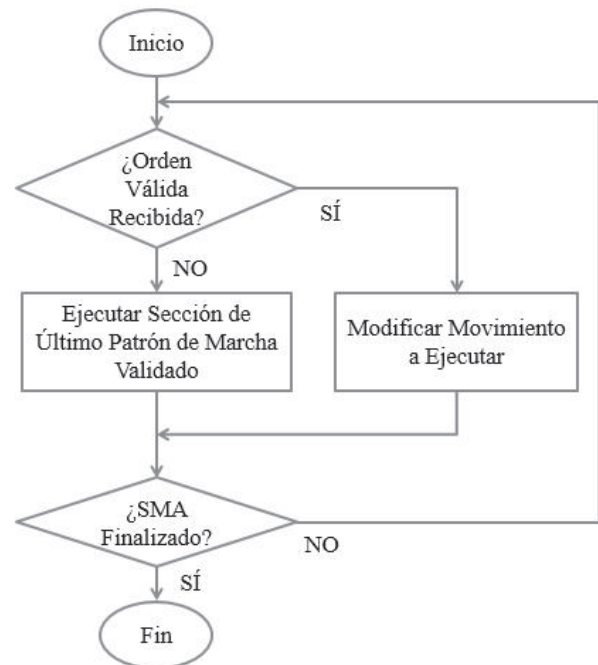


**Figura 9.** Trama de información.

**Fuente:** elaboración propia.

## Ejecución de patrones de marcha

A diferencia de los componentes anteriores, este proceso corresponde a un algoritmo en C ejecutado por los microcontroladores de los robots humanoides. Su código es idéntico para los cinco robots; la única modificación es el valor de la variable de identificador, la cual permite distinguir cuáles son los mensajes destinados a cada robot. La estructura del código ejecutado por los robots se detalla en la figura 10.



**Figura 10.** Componente de ejecución de patrones de marcha.

**Fuente:** elaboración propia.

En cada ciclo de ejecución, el proceso revisa si ha llegado un nuevo dato. En caso afirmativo, se verifica que el identificador concuerde con el robot, tras lo cual se extrae el contenido del mensaje y se verifica si es distinto al último comando de movimiento ejecutado por el robot. En caso negativo, el robot permanece estático o continúa ejecutando el último comando de movimiento tipo marcha recibido. Es importante distinguir movimientos finitos (sentarse, levantarse, levantar el objeto) de los patrones de marcha, los cuales se ejecutan de forma continua.

El kit de desarrollo que ofrece el fabricante incluye software para desarrollar movimientos y patrones de marcha, el cual incluye dieciséis patrones prediseñados que permiten al robot moverse en una variedad de direcciones. Para ejecutar dichas secuencias desde código en C, se implementan las librerías y documentación desarrolladas por el Humanoid Lab del "Instituto de Robòtica i Informàtica Industrial, CSIC-UPC". Debido a que todos los procesos del robot quedan en espera hasta terminar la ejecución de un movimiento, se hace necesario subdividir los patrones de marcha para permitir al robot leer periódicamente el buffer de recepción ZigBee y verificar si ha llegado un nuevo comando.

A fin de permitir a los robots desplazarse mientras cargan un objeto, se crean copias de los patrones de marcha predefinidos en las cuales se modifica la posición de los brazos del robot. Tal como se indica en Saab, Souères y Fourquet (2009) y Motoi, Ikebe y Ohnishi (2007), el centro de masa del robot varía al combinar manipulación y locomoción, por lo cual dicha variación se debe compensar cambiando la inclinación del robot a nivel de su articulación dorsal.

## Elaboración de ambiente virtual

Dada la complejidad de los Sistemas Multi-Agente compuestos por robots, su simulación suele llevarse a cabo únicamente por medio de Interfaces

Gráficas de Usuario (GUI) simples, con representación vía consola de texto o escenarios en dos dimensiones. Se hizo uso del software Webots© para facilitar la comprobación de la totalidad del algoritmo de comando del SMA, puesto que se elimina la necesidad de programar el entorno gráfico y los modelos 3D de los agentes.

Se escoge evaluar el algoritmo con dos robots humanoides y una única mesa cuadrada por recolectar; dicho número de agentes es óptimo dadas las dimensiones de la mesa. Se hace uso del modelo del robot humanoide DARwIn-OP para reemplazar los robots Bioloid, puesto que son fabricados por la misma empresa y la ejecución de sus patrones de marcha es análoga. Dado que Webots© está basado en la programación orientada a objetos, tanto los agentes como la mesa se representan creando objetos de sus clases correspondientes ("robot" y "sólido", respectivamente).

Siguiendo la representación abstracta previamente establecida en la figura 2, el componente central de decisión se representa mediante el archivo principal del controlador programado y los componentes restantes se asocian al incluir librerías predefinidas y llamar sus funciones correspondientes. La mesa es importada a partir de un modelo CAD, lo cual permite asociar adecuadamente sus fronteras (*bounding object*) y simular reacciones físicas de contacto mediante el motor de física del software.

## METODOLOGÍA

A fin de evaluar el desempeño del Sistema Multi-Agente, se requiere definir una situación estándar de exploración del ambiente y recolección de objetos: se ubica una única mesa y dos agentes recolectores en el interior del ambiente. Las variables iniciales de los agentes desplegados son la orientación y la distancia respecto a la mesa. Esta situación requiere el uso por parte de los agentes de todos los subprocesos que componen el algoritmo de organización.

Se realizan múltiples ejecuciones modificando las posiciones y orientaciones de los agentes de forma aleatoria, y se analizan las subsecuentes mediciones de los parámetros de desempeño (tiempo de ejecución, tasa de errores, entre otros). La situación estándar de exploración y recolección, así como los parámetros de desempeño se detallan con mayor profundidad en la siguiente sección.

## RESULTADOS

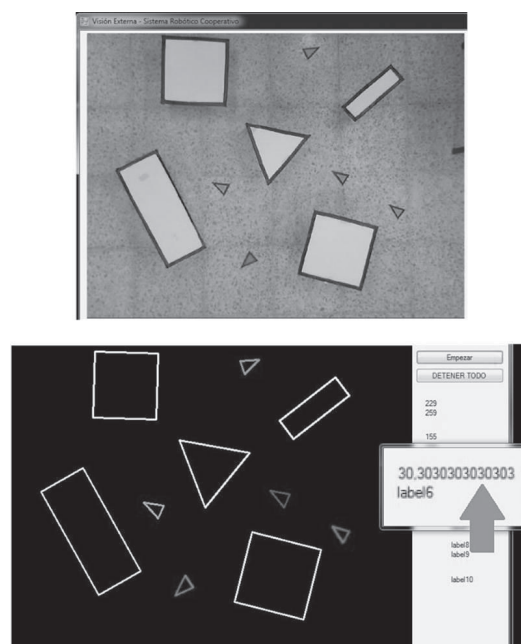
Los resultados de desempeño del Sistema Multi-Agente en la situación estándar son analizados en tres etapas. Una etapa preliminar corresponde a la valoración individual del proceso de visión de máquina, el cual es el componente de mayor complejidad en el algoritmo de organización. La segunda corresponde a la simulación del algoritmo completo en el interior del ambiente virtual. Finalmente, se verifica la ejecución de todos los componentes de manera simultánea en la tarea de recolección.

### Resultados de visión de máquina

Para evaluar el proceso de visión, se distribuyen todos los marcadores de los robots y las mesas por recolectar en posiciones y orientaciones aleatorias a lo largo del espacio de trabajo. El componente de visión es denominado robusto si posee la capacidad de detectar los marcadores y las mesas sin ser afectado por el ruido debido al patrón de baldosas del suelo y la iluminación artificial del laboratorio. La figura 11 muestra la imagen resultante tras el procesamiento, en la cual se han extraído y clasificado exitosamente todos los elementos de interés del espacio de trabajo. Es importante destacar la eficiencia del código implementado, puesto que la tasa de *frames* por segundo (fps) nunca disminuye de 30, valor óptimo para aplicaciones en robótica.

### Simulación en ambiente virtual

Se definen los siguientes parámetros iniciales para configurar la situación estándar definida en la



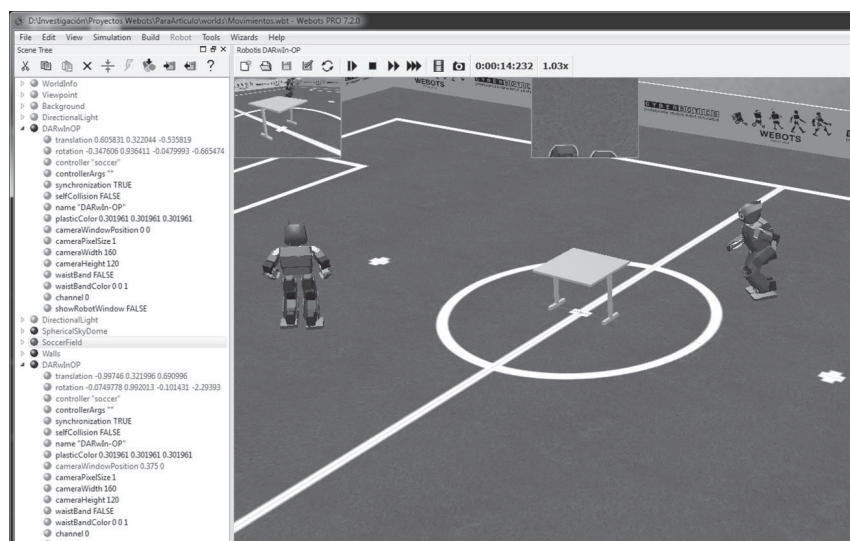
**Figura 11.** Extracción de elementos de interés.

**Fuente:** elaboración propia.

metodología: usando un generador de números aleatorios, ambos robots son ubicados a distancias de la mesa oscilando entre 0,75 y 1,25 metros y ángulos entre 0 y  $\pi$  radianes. Dicho rango de distancias se escoge con el objetivo de maximizar el desplazamiento de los robots sin exceder el límite inferior del espacio de trabajo (1.35 metros de ancho).

Como medida de desempeño global, se escoge el tiempo total de ejecución, medido desde el inicio de la simulación hasta el posicionamiento adecuado de ambos robots alrededor de la mesa para recolección. La posición y orientación final del robot se determina como fallida si, al levantar sus brazos, ambas manos no establecen contacto con la mesa.

Un ejemplo de ejecución de la simulación se presenta en la figura 12, y todos los resultados de ejecuciones se muestran en la tabla 2, donde  $d_1$ ,  $d_2$  y  $\theta_1$ ,  $\theta_2$  indican la distancia a la mesa y el ángulo de orientación de cada robot, y “tiempo” indica el tiempo total de ejecución en segundos. A pesar



**Figura 12.** Ejecución de la simulación en Webots®.

**Fuente:** elaboración propia.

de un mínimo de aproximadamente 20 segundos, la tarea fue frecuentemente completada entre 32 y 36 segundos.

**Tabla 2.** Resultados de simulación.

Prueba	$d_1$ (m)	$\theta_1$ (rad)	$d_2$ (m)	$\theta_2$ (rad)	Tiempo (s)
1	1,23	0	0,94	3,1416	35,464
2	1,21	1,3298	1,0837	2,74	32,664
3	1,0679	2,8668	1,1677	1,216	33,544
4	0,7702	0,4286	1,2457	2,1501	19,976
5	0,9727	3,1321	0,9394	1,5521	33,48

**Fuente:** elaboración propia.

## Validación en tarea de recolección

La ejecución simultánea de todos los componentes se realizó con la misma medida de desempeño global (tiempo total de ejecución) y la misma cantidad de elementos utilizados en la simulación (2 robots humanoides Bioloid y una mesa de geometría cuadrada). Se realizaron seis pruebas, en las cuales los robots y la mesa fueron ubicados en posiciones aleatorias, garantizando que la distancia de los

robots a la mesa estuviera en un rango entre 0,75 y 1,25 metros y la orientación inicial de los robots no coincidiera con la dirección hacia la mesa.

Los datos obtenidos a partir de las pruebas se presentan en la tabla 3, donde la columna “Correcciones” corresponde al total de correcciones de orientación realizadas por los robots durante la prueba, y la columna “Error de Detección” corresponde al número de veces que el algoritmo no detectó adecuadamente el cumplimiento de la orientación deseada por parte del robot y por ende se produce *overshoot*.

**Tabla 3.** Resultados de pruebas en el SMA.

Prueba	Duración (s)	Correcciones	Error de Detección
1	18,48	2	1
2	23,08	4	2
3	19,52	6	1
4	29,24	5	3
5	25,42	6	2
6	17,15	0	1

**Fuente:** elaboración propia.



El menor tiempo en el cual se completó la secuencia de movimientos fue de aproximadamente 17 segundos. Se observó que los robots humanoides presentan un grado significativo de oscilación lateral mientras se desplazan; puesto que los marcadores se ubican sobre la cabeza de los robots, dicha oscilación afecta la medición de su orientación y puede causar cambios de sentido de giro innecesarios una vez el robot se encuentra con una orientación cercana a la deseada.

## CONCLUSIONES

La implementación de componentes paralelos mediante programación por hilos, junto con el uso de las librerías de EmguCV, permitió simplificar considerablemente el desarrollo del sexto agente y garantizar el rendimiento del sistema al evitar interferencia mutua entre el proceso de comunicación inalámbrica ZigBee y el proceso de visión de máquina a 30 *frames* por segundo (fps).

El diseño de las tramas de datos y los mecanismos para verificar el mensaje y el nuevo comando de movimiento optimizan la ejecución de los patrones de marcha en los robots humanoides Bio-loid, al evitar la ejecución de órdenes destinadas a agentes diferentes o la repetición de reenvío/ejecución de órdenes previamente realizadas.

Las mediciones del SMA llevadas a cabo por el algoritmo (corrección de orientación y distancia al objetivo) son susceptibles a variaciones inducidas por la oscilación de los robots al desplazarse y las condiciones del entorno de trabajo (patrón de baldosas, luz artificial). Dichas variaciones pueden extender el tiempo que toma a los robots cumplir la tarea de recolección.

## FINANCIAMIENTO

Este artículo se deriva del proyecto de investigación ING 1198: Desarrollo de Estrategia con Aprendizaje On-Line para Comando de Grupo de Robots Humanoides Utilizando la Plataforma Robótica DARwIn-OP, del Grupo de Investigación DAVINCI,

perteneciente al Centro de Investigaciones de la Facultad de Ingeniería de la Universidad Militar Nueva Granada. El proyecto posee el aval y financiamiento de la Universidad Militar Nueva Granada.

## REFERENCIAS

- Cheng, S., Raja, A. & Lesser, V. (2013). Multiagent meta-level control for radar coordination. *Web Intelligence and Agent Systems: An International Journal*, 11(2) (Jul.), 81-105.
- De Giacomo, G., Lespérance, Y., Levesque, H., & Sardina, S. (2009). IndiGolog: A high-level programming language for embedded reasoning agents. En R. Bordini, M. Dastani & A. El Fallah Seghrouchni, *Multi-agent programming: languages, tools and applications* (pp. 31-72). New York, United States of America: Springer.
- Fu-sheng, Z., Hong-bo, W., Guang-ping, Z., Yang, Y. & Xu-yan, T. (2009). Multi-softMan coordination model and its application in metallurgical construction project management. *New Trends in Information and Service Science, 2009. NISS '09. International Conference on*, Jun. 30-Jul. 2 (pp. 963-968).
- Gu, Y., Thobbi, A. & Sheng, W. (2011). Human-robot collaborative manipulation through imitation and reinforcement learning. *Information and Automation (ICIA), 2011 IEEE International Conference on*, 6-8 Jun (pp. 151-156).
- Kuo, J. & Lin, H. (2008). Cooperative RoboCup agents using genetic case-based reasoning. *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, 12-15 Oct. (pp. 613-618).
- Malmberg, R. (2011). A comparison between robot gait and human gait with the help of literature research and Gait Analysis of the Nao Robot H25. Recuperada de <http://www.personeel.unimaas.nl/roos/robotlab/files/Roos%20Malmberg.pdf>
- McGill, S. & Lee, D. (2011) Cooperative humanoid stretcher manipulation and locomotion. *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 26-28 Oct. (pp. 429-433).
- Motoi, N., Ikebe, M. & Ohnishi, K. (2007). Real-Time Gait Planning for Pushing Motion of Humanoid



- Robot. *Industrial Informatics, IEEE Transactions on*, 3(2) (May.), 154,163.
- Mukovskiy, A., Slotine, J. & Giese, M. (2010). Design of the dynamic stability properties of the collective behavior of articulated bipeds. *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, 6-8 Dec. (pp. 66-73).
- Ping-Huan, K., & Li, T. (2011). Development of simulator for kid-sized humanoid soccer in RoboCup. *System Science and Engineering (ICSSE), 2011 International Conference on*, 8-10 Jun. (pp. 155-160).
- Saab, L., Souères, P. & Fourquet, J. (2009). Coupling manipulation and locomotion tasks for a humanoid robot. *Advances in Computational Tools for Engineering Applications, 2009. ACTEA '09. International Conference on*, 15-17 Jul. (pp. 84-89).
- Shoham, Y., & Leyton-Brown, K. (2007). *Multiagent systems: algorithmic, game-theoretic, and logical foundations*. New York, United States of America: Cambridge University Press.
- Wooldridge, M. (2009). *An introduction to multiagent systems* (2.<sup>a</sup> Ed.). Glasgow, United Kingdom: Wiley.
- Xuan, P. & Lesser, V. (2002). Multi-Agent Policies: From Centralized Ones to Decentralized Ones. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems* (Vol. Part 3), 15-19 Jul. (pp. 1098-1105).
- Zhiguo, S., Junming, W., Xu, L., Zhiliang, W. & Jun, T. (2011). IGRS protocol based mobile service robot positioning and multi-robot collaboration for smart home. *Control Conference (CCC), 2011 30th Chinese*, 22-24 Jul. (pp. 4741-4746).



