ARQUITECTURA DE CONTROL CONDUCTUAL PARA AGENTES INTELIGENTES

ARCHITECTURE OF BEHAVIORAL CONTROL FOR INTELLIGENT AGENTS

Joel Ricardo Jiménez Cruz
Universidad Autónoma Metropolitana
jcjr@xanum.uam.mx

Resumen

En este trabajo se simula, por medio del lenguaje de programación NetLogo, el comportamiento adaptativo de un agente inteligente ante su medio ambiente. El comportamiento está regido por una arquitectura de control conductual de inspiración biológica que se implementa a partir de máquinas de estado. Con este tipo de arquitectura, se aborda la problemática de que el agente elija la respuesta conductual más apropiada en función de las circunstancias de su entorno y de la estimulación recibida. Se reporta y compara el funcionamiento del agente a partir de dos experimentos que utilizan 5 escenarios y 4 controladores. Las simulaciones de este comportamiento inteligente se pueden implementar en robots móviles autónomos, en agentes asistentes o tutores, o en aquellos agentes que buscan y recuperan información en bases de datos o en Internet (softbots).

Palabras Claves: Agentes inteligentes, arquitectura de control conductual, comportamiento adaptativo, NetLogo, selección de acción.

Abstract

This work simulates, through the NetLogo programming language, the adaptive behavior of an intelligent agent in its environment. The behavior is governed by a behavioral control architecture of biological inspiration that is implemented from state machines. With this type of architecture, the problem addressed is that the agent chooses the most appropriate behavioral response depending on the circumstances of its environment and the stimulation received. The performance of the agent is reported and compared from two experiments using 5 scenarios and 4

controllers. The simulations of this intelligent behavior can be implemented in autonomous mobile robots, assistant agents or tutors, or in those agents that search and retrieve information in databases or the Internet (softbots).
Keywords: *Action selection, adaptive behavior, behavioral control architecture, intelligent agents, Netlogo.*

1. Introducción

La simulación en computadora permite recrear los comportamientos de organismos en los entornos o nichos en los que viven, modelando sus sensores, el procesamiento de información que llevan a cabo y las tareas que realizan [Jiménez, 2018]. Una vez realizada la simulación se puede proceder a implementarla en robots físicos o virtuales o agentes computacionales (sofbots) que tiene acceso a fuentes de información y son capaces de verificar y manipular la información obtenida de estas fuentes de acuerdo a las consultas planteadas por los usuarios con el fin de obtener los recursos esperados. Las fuentes de información pueden ser de muchos tipos, incluyendo, por ejemplo, las bases de datos y la información en Internet.

El estudio del comportamiento animal y los experimentos neurobiológicos pueden servir de guía para el diseño y la construcción de sistemas artificiales que logren un comportamiento autónomo y adaptativo: Cuando un animal confronta su entorno, su comportamiento se ajusta continuamente para afrontar las situaciones, necesidades y condiciones del medio ambiente tanto internamente como externamente [Berns, 1996]. Por ejemplo, el alimentarse, pelear, huir y aparearse son cuatro impulsos básicos (conocidos en biología como las 4 Fs) que el organismo puede enfrentar y que determinan sus comportamientos, en este caso, de sobrevivencia [Lorenz, 1986].

En el procesamiento de información que realizan los organismos y los sistemas artificiales están involucrados el cerebro (el controlador), el cuerpo (interacciones internas y externas) y las relaciones biunívocas con su medio ambiente. En los agentes artificiales esta relación tripartita, está representada por la arquitectura de control que incluye: la percepción y actuación selectiva sobre el medio ambiente;

el procesamiento de la información interna y externa y; los sensores, actuadores o efectores.

La selección de la conducta más apropiada por parte del agente es un tema que se conoce como “selección de acción” y es muy importante cuando se desea que el agente exhiba una conducta inteligente y adaptable que considere la predicción y la evaluación de las consecuencias de sus acciones [Vanderelst, 2018]. El mecanismo de selección se coloca en una capa de la arquitectura o se distribuye en sistemas más pequeños, más manejables y flexibles que permiten realizar el arbitraje y priorización de las conductas y que gestionan el mantenimiento, la inhibición y la alternancia de comportamientos cuando se tienen objetivos conflictivos múltiples [Gaudla, 2018].

Las arquitecturas de control inteligente se pueden clasificar en varias categorías: por el tipo de interacción entre módulos de control (por ejemplo, arquitecturas jerárquicas o centralizadas), por el tipo de funcionalidad asignada a cada módulo (por ejemplo, de propósito general o específico) y por la forma en que los módulos interactúan con el entorno externo (p. ej., comportamiento basado en eventos o en procedimientos). La última categoría se puede clasificar en deliberativa, reactiva, híbrida y conductual [Tzafestas, 2018].

La arquitectura de control deliberativo está basada en un conocimiento previo y en la observación interna de los estados y de las acciones, en ella, se genera un plan basado en un objetivo y un modelo estático del entorno. La arquitectura de control reactivo se basa en un modelo estímulo-respuesta que permite al robot o agente ocuparse de entornos muy dinámicos e impredecibles, los comandos de control se generan durante su operación en función de la interacción del agente con el entorno. Una arquitectura de control híbrida combina las ventajas de los enfoques reactivos y deliberativos, al tiempo que disminuye sus inconvenientes individuales, pero su complejidad computacional es mayor. El agente deliberativo genera de forma estática planes que se ejecutarán en tiempo real, reaccionando al entorno. Finalmente, la estructura de una arquitectura de control conductual se implementa a partir de una colección de comportamientos independientes que se auto-organizan para llevar a cabo las tareas encomendadas [Cadavid, 2017].

De estas, la arquitectura reactiva es muy popular por su sencillez, porque responde de manera inmediata a los estímulos del medio ambiente, aun cuando no tienen la capacidad de anticipación, memoria o aprendizaje [Ben-Ari, 2018]. Entre los ejemplos de las arquitecturas reactivas se encuentran las tortugas de Grey Walter en los años 50s, los vehículos de Valentino Braitenberg y las criaturas de Rodney Brooks de la década de los 80s.

Las tortugas de Walter son máquinas especulativas que exploraban su medio ambiente, por medio de sensores de luz y de contacto, así como dos motores uno para propulsarlo y el otro para poder guiarlo en el medio ambiente. El procesamiento se hacía con un circuito analógico de bulbos. Las tortugas exhiben conductas complejas (buscar una luz, alejarse de ella y, evitar obstáculos) [Walter, 1950], [Walter, 1951].

Los vehículos de Braitenberg utilizan conexiones simples excitatorias e inhibitorias entre sensores y motores produciendo comportamientos como agresión, miedo, amor, exploración [Hogg, 1991]. Las criaturas de Brooks están basadas en la arquitectura de sumisión (subsumption), constituida por capas que trabajan en paralelo y que se organizan en forma priorizada por medio de un mecanismo de coordinación competitivo. Los comportamientos muestran un alto grado de inteligencia sin que exista, en realidad, un procesador central [Brooks, 1986], [Jones, 1993].

Actualmente, la arquitectura conductual es la base del diseño de muchos agentes inteligentes, con la cual el agente elige y coordina la conducta más adecuada de acuerdo a los estímulos recibidos y al contexto en el que se encuentra. Por ejemplo, [Cao, 2018] presentan un sistema de control conductual para la interacción del ser humano con robots sociales personalizables que adaptan sus comportamientos a los perfiles de los usuarios y a las respuestas durante la interacción humano-robot. La programación de las conductas de un agente se puede inspirar en la forma en que los animales resuelven el conflicto de la selección de una conducta al momento en que varias unidades funcionales entran en competencia por un conjunto limitado de acciones y sólo una unidad funcional toma el control de las salidas motoras [Jiménez, 2000], [Grillner, 1997].

En este trabajo se explora y efectúa la simulación en NetLogo de un agente que utiliza una arquitectura conductual cuyos módulos se implementan a partir de máquinas de estado. Netlogo se ha convertido en una plataforma estándar para el modelado y la simulación de agentes que permite la exploración de fenómenos emergentes en una gran variedad de campos del conocimiento [Tisue, 2004], [Raglin, 2018].

Las conductas que se exploran son la búsqueda de alimentos, la evitación de obstáculos, el seguimiento de una orilla o pared. Se reporta y compara el funcionamiento del agente a partir de dos experimentos que utilizan 5 escenarios y 4 controladores. Una vez realizadas las simulaciones, la arquitectura conductual se puede implementar en robots móviles autónomos, agentes asistentes o tutores, o en aquellos agentes que buscan y recuperan información en bases de datos o en Internet (softbots).

En las secciones que siguen se expone la teoría y la forma en que están constituidas estas arquitecturas de control conductual y se muestran las simulaciones del agente utilizando esta arquitectura. Finalmente se llevan a cabo la discusión y las conclusiones pertinentes.

2. Métodos

La arquitectura de control conductual se puede inspirar y desarrollar a partir de componentes simples y consistentes, observando la estructura de los sistemas biológicos e imitándola. En la implementación de la arquitectura, los comportamientos se van agregando incrementalmente al sistema, se ejecutan en paralelo y se activan en respuesta a condiciones externas y/o internas, entradas sensoriales y estados internos o mensajes de otros comportamientos. La dinámica de la interacción surge tanto dentro del propio sistema, a partir de la combinación e integración interna de los comportamientos, como de la relación de éstos con el entorno [Prescott, 1999].

Las arquitecturas de control conductual pueden contener componentes reactivos, sin utilizar representaciones complejas. De hecho, la estructura e interacción de los comportamientos internos, produce una representación del mundo externo y de

sus relaciones con él y pueden servir de base para el aprendizaje y la predicción. Esta información se almacena de manera distribuida entre los módulos de comportamientos que componen la arquitectura, a diferencia de los sistemas deliberativos e híbridos que incluyen un planificador deliberativo, un componente reactivo y a veces un sistema motivacional que utilizan un razonamiento o representación centralizada como un mapa global para poder realizar las tareas y planes de contingencia [Rodríguez, 2018].

El poder y la flexibilidad del control conductual proviene, no sólo de los comportamientos, sino también de la forma en que se combinan y se organizan, de acuerdo con las siguientes características [Mataric, 2007], [Feng, 2018)]:

- Logran y/o mantienen objetivos conductuales como buscar una ubicación, seguir la pared, esconderse de la luz, recargarse, buscar una pareja, etc.
- Tienen una duración extendida (no instantánea, como en el caso de los sistemas reactivos).
- Toman entradas de sensores y también de otros comportamientos, y envían salidas a efectores y a otros comportamientos, creando redes de comportamientos que "se comunican" entre sí.
- Ejecutan comportamientos en paralelo/concurrentemente en escalas de tiempo compatibles, lo que permite la respuesta en tiempo real.
- Utilizan redes de comportamientos para almacenar estados (presentes y futuros) de manera distribuida y construir modelos/representaciones que generan un comportamiento eficiente (no solo reactivo).
- La estructura y representación es uniforme en todo el sistema (sin conexiones intermedias entre capas, como en los sistemas híbridos).

Un ejemplo del control conductual y toma de decisiones es el trabajo de Randall Beer [Randall Beer, 1991] que implementa la coordinación de comportamientos de una cucaracha artificial (Periplaneta Computatrix) en su medio ambiente a partir de simulación de circuitos neuronales que contienen 78 neuronas y 156 conexiones.

Otro ejemplo es, el de [Jiménez, 2017] que modelan y simulan la navegación y la evitación de obstáculos de un insecto artificial por medio de redes neuronales de

espigas utilizando los conceptos de sistemas multiagentes y el entorno de programación Netlogo que permite monitorear y manipular en cada iteración de simulación el estado de cada elemento del circuito neuronal incluyendo:

- Las neuronas y sus variables internas.
- Las sinapsis y sus parámetros.
- Las características de las espigas.

En el presente trabajo se simulan por medio de autómatas o máquinas de estados algorítmica (ASM) las capacidades motoras, sensoriales y de procesamiento de un agente, la evitación de obstáculos, el deambular, el seguimiento de una orilla y la búsqueda y consumo de alimento. El agente tiene sensores de proximidad que modelan las antenas y que le permiten avanzar sin chocar o desviarse de su camino, un sensor de energía que simula el hambre y sensores olfativos y táctiles en la boca que señalan que el alimento está en la boca del agente y además que señala las acciones realizadas por el agente en función de sus condiciones internas y del medio ambiente (figura 1).

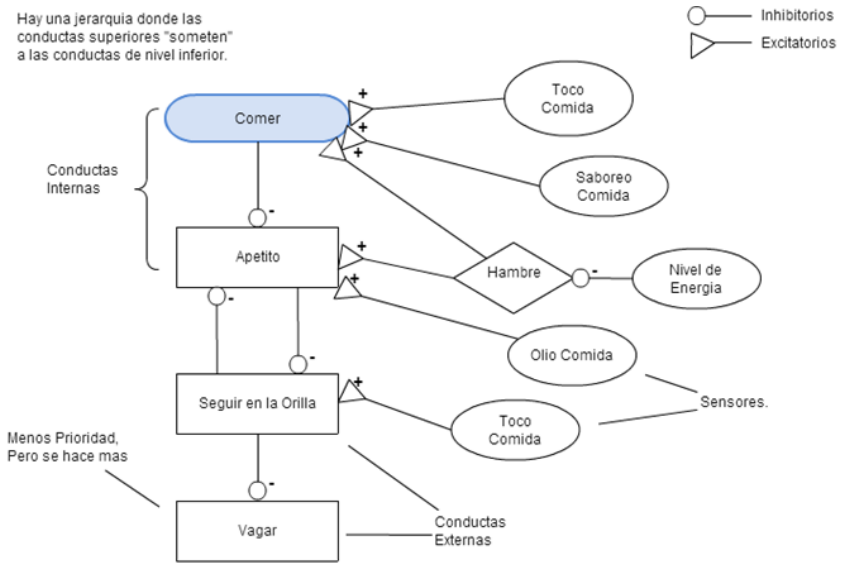


Figura 1 Máquina de estados de las conductas del agente.

Una máquina ASM es un sistema abstracto que puede estar en uno de un número finito de estados. El sistema está en un estado a la vez (el estado actual). Puede

pasar de un estado a otro cuando se produce un evento o condición desencadenante (reglas de transición) [Ben-Ari, 2018b].

Una máquina ASM se define por un conjunto de:

- Posibles estados.
- Posibles transiciones de cada estado.
- Condiciones de activación para cada transición.
- Reglas que rigen el comportamiento del sistema en cada estado.

La máquina de estados muestra las posibles acciones o conductas que el agente puede tomar en función de los estímulos que se van recibiendo. La máquina representa una jerarquía en la cual las conductas de nivel superior pueden inhibir a las conductas de nivel inferior.

Para codificar una conducta, ésta se puede representar por un conjunto (E, R, C), donde E denota a los estímulos, R el rango de posibles respuestas y C la función de mapeo: $C: E \rightarrow R$. Una respuesta particular r está dada por $r = D\ (G\ *\ C(E))$, donde G es la ganancia o fuerza relativa de cada conducta y D es una función de la elección de la conducta que puede ser competitiva o cooperativa. En la función competitiva, las conductas compiten por su activación y puede organizarse por prioridades en forma jerárquica que pueden incluir varias capas o por enlaces libres (redes de conductas) y la función cooperativa se caracteriza por la fusión o integración de conductas [Atkinson, 1998], [Muñoz, 2018].

En este trabajo se codificaron las conductas en forma competitiva, representadas por el siguiente pseudocódigo:

Repite
Vagar ( );
Si estás hambriento y tienes el alimento en la boca
    Consume la comida;
Si estas hambriento y hay olor
    Sigue el rastro;
Si hay un contacto con la antena
    Sigue la orilla;
Hasta el final

3. Resultados

Para la simulación de la arquitectura conductual del agente se utilizó el lenguaje de programación Netlogo (en su versión 6.0.3) que proporciona un entorno integrado de simulación y programación de modelos de agentes múltiples para el estudio del comportamiento emergente en sistemas complejos. El lenguaje de programación Netlogo ofrece un conjunto de primitivas que permite a los agentes percibir y modificar su mundo virtual y también comunicarse e interactuar con otros agentes u objetos [Sakellariou, 2008]. Además de su simplicidad, una de las principales ventajas de usar Netlogo, es que permite monitorear sensores, estímulos y manipular la representación del agente, de su medio ambiente y por ejemplo, analizar las conductas de movimiento, evitación de obstáculos, búsqueda y recolección de comida [Zaharija, 2014].

En el presente trabajo, se realizaron varias simulaciones de las reacciones del agente considerando distintos escenarios, circunstancias, distribuciones de alimento y obstáculos durante un tiempo de 1500 unidades (ticks). En la pestaña de información de Netlogo se van anotando los resultados y observaciones correspondientes a cada experimento.

Características y configuración del escenario y del agente

En la figura 2 se muestra el escenario donde se encuentra el agente, las líneas de color azul representan obstáculos, las áreas de color verde el olor de la comida y los puntos de color blanco indican la comida o pellets. El agente tiene sensores izquierdo y derecho para oler la comida y sensores de contacto en las antenas izquierda y derecha que se activan cuando el agente toca una línea azul. El agente se pone de color rojo cuando se activa la antena derecha, de color verde cuando se activa la antena izquierda y de color azul cuando se activan ambas antenas.

El score determina el número de pellets recolectados o comidos en un tiempo de 1500 ticks. Hay monitores que muestran los estados actuales y siguientes de la máquina de estados. Se tiene un botón que aleatoriza la disposición de la comida en función del escenario. Se disponen de 5 escenarios con los cuales se evalúa el desempeño del agente.

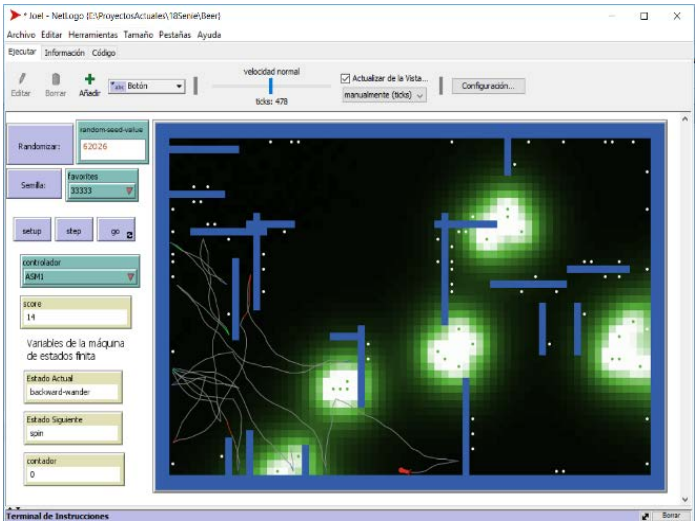


Figura 2 Simulación del agente en NetLogo.

Los números aleatorios que se generan son pseudo-aleatorios y en cierta manera determinísticos, porque repitiendo la misma semilla se generan los mismos resultados aleatorios que son útiles para replicar y constatar el funcionamiento de la simulación.

Experimentos

En esta sección se reporta el funcionamiento del agente a partir de dos experimentos que utilizan 5 escenarios y 4 controladores: Vagar, ASM1, ASM2, seguir la orilla o pared. El controlador de vagar, camina y cambia su orientación de manera aleatoria. ASM1 avanza hacia adelante, hacia atrás y gira. ASM2 avanza hacia adelante, hacia atrás, gira, sigue el aroma. Seguir la pared es un algoritmo que recorre la pared por la izquierda.

Experimento 1

Se escogieron 5 escenarios diferentes particularmente interesantes y desafiantes para el agente para que pueda recoger el mayor o el menor número posible de pellets, y así evaluar el desempeño del agente. Se anotó en una tabla: el valor de la semilla para cada escenario que se escogió; el score obtenido y las anotaciones de la conducta del agente (tabla 1).

Tabla 1 Cantidad de pellets recolectados de los controladores desarrollados.

Semilla/escenario	Controlador/Score				Observaciones
	Vagar	ASM1	ASM2	Seguimiento de la pared	
11111	21	24	14	51	Con FSM2 se queda encerrado en un pasillo, cuando sale se mueve en círculo intentando alcanzar un pellet
22222	11	26	22	56	Con Vagar se pierde mucho tiempo golpeando la pared, sin realizar un desplazamiento significativo
33333	29	22	18	44	Con FSM2 solo recorre la mitad izquierda del mundo
44444	24	26	29	58	Con Vagar no entra a un bloque que tenga pellets
55555	14	26	29	12	Con seguimiento de pared se quedó atrapado en una isla
Promedio	19.8	24.8	22.4	44.2	

Experimento 2

Otro experimento que se realizó consistió en anotar la cantidad de pellets consumida y las observaciones particulares, variando las semillas para cada una de las conductas; vagar (tabla 2), ASM1 (tabla 3), ASM2 (tabla 4) y seguimiento de la pared (tabla 5).

Tabla 2 Conducta de Vagar.

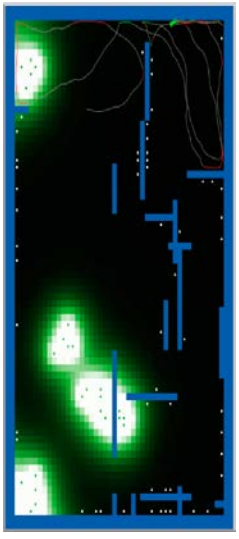
Semilla	Score	Anotaciones	
94225	18	Sólo quedó atorado en ocasiones	
38698	27	Vagó por casi todo el escenario	
94013	14	Quedó atorado en un “callejón sin salida”	
82613	25	Vagó por la mitad de escenario por un obstáculo a la mitad del escenario	
24720	21	Recorrió casi todo el escenario	

Tabla 3 Controlador ASM1.

Semilla	Score	Anotaciones
32032	24	Vagó por casi todo el escenario
36253	13	Se metió en un cuadro con una salida muy pequeña
83510	19	Vagó por casi todo el escenario
92875	22	Vagó por casi todo el escenario
74757	21	Se metió en un cuadro con una salida muy pequeña con varios pellets

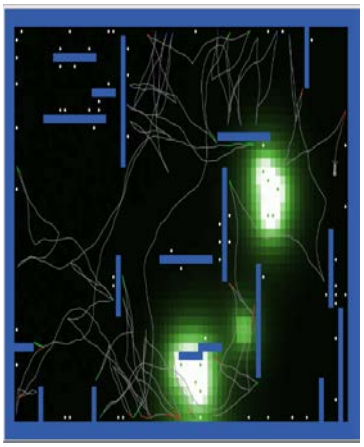


Tabla 4 Controlador ASM2.

Semilla	Score	Anotaciones
25432	22	Comió sólo en dos áreas donde había alimento
76209	27	Comió solo en tres áreas donde había alimento
61118	43	Vagó por casi todo el escenario devorando la mayoría de la comida cuando se encontraba aglutinada en una sola área
92875	22	Vagó por casi todo el escenario
12116	13	Se metió en un recuadro con salida angosta y sólo comió lo que estaba dentro

Tabla 5 Seguimiento de la pared.

Semilla	Score	Anotaciones
96012	4	Se quedó estancado en un obstáculo y sólo se quedó rodeando a éste
54944	43	Rodeó todo el escenario
87780	42	Rodeó todo el escenario
94412	9	Se quedó estancado en un obstáculo y sólo se quedó rodeando a éste
9914	48	Rodeó todo el escenario

4. Discusión

La arquitectura de control conductual que se implementó en este trabajo se estructuró en base a una organización de comportamientos que interactúan con el entorno del agente, con escalas de tiempo similares y sin un control central que opera en paralelo, gracias al procesamiento concurrente que ofrece NetLogo. En este sentido, el control combina y administra la estructura modular y el funcionamiento interno con la dinámica de interacción de los comportamientos del agente con su entorno. De esta manera, los patrones e historia de la interacción pueden interactuar para producir conductas más flexibles, complejas y emergentes.

En los experimentos realizados se observó que la cantidad de comida ingerida depende del controlador empleado y de la distribución de los obstáculos y de la comida. Si los obstáculos están muy cerca, el agente tarda más tiempo es salir de un área determinada, una vez que logra salir, puede explorar nuevamente su medio ambiente con el fin de localizar las fuentes de alimentación.

En los distintos escenarios que se probaron, la mejor estrategia resultó ser ASM2 debido a que incorpora, además del avance hacia atrás y hacia adelante y el giro, la detección del olor de la comida.

Un controlador muy útil, sin duda, es aquel que implementa la conducta de deambular o vagar de manera aleatoria en su medio ambiente. Es una estrategia utilizada por todos los animales y humanos cuando se trata de encontrar un recurso o fuente de alimentación. También resulta muy eficaz el comportamiento implementado por el controlador ASM1 que contiene una búsqueda especializada que consiste en avanzar hacia adelante y cuando el agente queda atrapado, avanza hacia atrás y gira. Por último, la conducta de seguir la orilla o pared también ejemplifica una técnica muy común utilizada por los animales para buscar un objeto o comida.

La arquitectura de control conductual que se implementó se basan en los principios y efectos de la filosofía reactiva sin limitarse a ellos y considerando que los otros paradigmas están definidos por la relación entre el sensar, el planear y el actuar [Mataric, 2007], [Lazzeri, 2018)]:

- Los sistemas deliberativos requieren mucho tiempo de procesamiento, una elevada capacidad de memoria y computacionalmente son complejos debido a su representación simbólica del mundo externo.
- Los sistemas híbridos (reactivos y deliberativos) requieren medios de combinación e interacción complejos entre sus componentes.

Como se observó en los experimentos realizados, el agente conductual no cuenta con una representación global del mundo. Se encuentra situado en su medio ambiente y no requiere de descripciones abstractas, cuenta con perceptores y efectores adecuados para experimentar e interactuar directamente con su entorno. Su "inteligencia" no está programada explícitamente sino más bien es resultado de las interacciones internas del agente y de las interacciones entre él y su medio a ambiente. La "inteligencia" se determina por la conducta total del sistema, es decir, la "inteligencia" emerge [Brooks, 1991].

5. Conclusiones

Las arquitecturas de control pueden ser computacionalmente complejas o sencillas, cada una con sus pros y sus contras, y dependiendo de la aplicación deseada. Cada una de las arquitecturas de control permite a los robots o agentes, en diferentes grados, realizar de manera autónoma o semiautónoma una variedad de tareas en diferentes dominios y aplicaciones [Lee, 2018], [Biro, 2018].

La selección de una arquitectura de control depende principalmente de las propiedades intrínsecas del problema, del tipo de tarea deseada, de la optimicidad requerida y de la información disponible. Además, en la implementación física del robot o agente, puede estar estrechamente relacionado con las restricciones del hardware y software, la morfología del robot y la interacción con el entorno [Melidis, 2018].

En este trabajo se ha implementado y mostrado las bondades de la arquitectura conductual, pero los otros paradigmas también son útiles en diversas aplicaciones. Por ejemplo, los sistemas deliberativos proporcionan un razonamiento y una planificación que viene dada por una representación precisa del entorno. Esto

implica que el entorno no cambia; por lo tanto, estos sistemas son ideales para entornos estructurados y altamente predecibles, especialmente en situaciones donde el agente realiza repetidamente una tarea determinada.

Por otro lado, los sistemas reactivos producen resultados satisfactorios cuando se trata de entornos cambiantes y donde se necesita una respuesta o reacción inmediata en el desempeño de la tarea, como la evitación de obstáculos. Por lo tanto, los entornos cambiantes (estocásticos) se manejan mejor con sistemas reactivos. Además, los controladores reactivos son muy poderosos en entornos y tareas que se han caracterizado de manera anticipada y en donde la carga computacional no es excesiva. Sin embargo, cuando se necesita la representación del mundo y el conocimiento de las acciones pasadas y futuras, se pueden agregar otros módulos que contengan planificación, aprendizaje o memoria [Herrero, 2017].

Finalmente, los sistemas híbridos han demostrado ser una buena opción para entornos que necesitan modelos internos y requieren de una planificación a largo plazo y que, en tiempo real, las demandas del sistema son suficientemente independientes de un razonamiento de alto nivel utilizando lenguajes y sistemas operativos específicos que puedan responder rápidamente en entornos dinámicos [Cadavid, 2017].

En este trabajo, las conductas de deambular, seguimiento de una orilla y búsqueda de alimento se implementaron de manera sencilla y principalmente con fines ilustrativos de la arquitectura conductual. Si se tratara de tareas más complejas, sería necesario incorporar o combinar otras arquitecturas que promuevan la planificación y la supervisión modular con el fin de mejorar la representación del mundo y la adaptación del agente o robot a sus tareas y su medio ambiente [González, 2018].

El control conductual se puede utilizar como una alternativa a los métodos de búsqueda tradicionales tales como tablas hash, búsqueda secuencial, búsqueda binaria, o los métodos empleados por la Inteligencia Artificial clásica como árboles de búsqueda, método de generación y prueba, métodos heurísticos. En ciertas aplicaciones, estos métodos son poco flexibles para responder exitosamente ante

diversas situaciones; no se obtienen buenos resultados cuando los datos son imprecisos, están incompletos o si se pretenden realizar búsquedas más versátiles (por ejemplo empleando sinónimos, o a través de una consulta del usuario). Algunos de los métodos de búsqueda, citados anteriormente, necesitan de una representación en memoria del espacio de búsqueda y tienen una carga computacional elevada [Caballero, 1998].

6. Bibliografía y Referencias

[1] Atkinson, J. (1998). Diseño de Agentes Autónomos Utilizando un Enfoque de Control Basado en Conductas. Revista Facultad de Ingeniería, núm. 5, pp. 45-54. http://www.redalyc.org/pdf/114/11400507.pdf

[2] Beer, R., Chiel, J., Sterling, L. (1991). An artificial insect. American Scientist, Vol. 79(5): pp. 444-452. http://www.jstor.org/stable/29774478?seq=1#page_scan_tab_contents

[3] Ben-Ari M., Mondada F. (2018a). Reactive Behavior. In: Elements of Robotics. Springer, Cham, pp. 39-53. https://link.springer.com/chapter/10.1007/978-3-319-62533-1_3

[4] Ben-Ari M., Mondada F. (2018b). Finite State Machines. In: Elements of Robotics. Springer, Cham, pp. 55-61. https://link.springer.com/chapter/ 10.1007/978-3-319-62533-1_4.

[5] Berns G.S., Sejnowski T.J. (1996). How the Basal Ganglia Make Decisions. The Neurobiology of Decision Making, Eds. Damasio A., Damasio H., ChristenY.,Springer-Verlag: https://papers.cnl.salk.edu/PDFs/How%20the%20Basal %20Ganglia%20Make%20Decisions%201996-2876.pdf.

[6] Biro, A. (2018). Combining adjustable autonomy and shared control as a new platform for controlling robotic systems with ROS on TurtleBot. Örebro University Report. Independent thesis Advanced level. http://www.diva-portal.org/smash/get/diva2:1178533/FULLTEXT01.pdf

[7] Brooks, R. (1986). Robust layered control system for a mobile robot, IEEE J. Robotics and Automation, RA-2, pp. 14-23. https://ieeexplore.ieee.org /document/1087032/.

[8] Brooks, R. (1991). Intelligence Without Reason. A.I. Memo No. 1293. MIT. http://people.csail.mit.edu/brooks/papers/AIM-1293.pdf

[9] Caballero, H. (1998). Diseño e Implementación de una Sociedad de Agentes Inteligentes Recolectores de Información. Tesis de Ingeniero en Computación. Universidad Tecnológica de la Mixteca. http://jupiter.utm.mx/~tesis_dig/5152.pdf.

[10] Cadavid H., Pérez A., Rocha C. (2017). Reliable Control Architecture with PLEXIL and ROS for Autonomous Wheeled Robots. In: Solano A., Ordoñez H. (eds) Advances in Computing. CCC 2017. Communications in Computer and Information Science, vol 735. Springer, Cham. https://www.researchgate.net/publication/319138847_Reliable_Control_Architecture_with_PLEXIL_and_ROS_for_Autonomous_Wheeled_Robots.

[11] Cao, H-L., Van De Perre, G., Kennedy, J., Senft, E., Gomez Esteban, P., De Beir, A., ... Vanderborght, B. (2018). A personalized and platform-independent behavior control system for social robots in therapy: development and applications. IEEE Transactions on Cognitive and Developmental Systems, PP(99), 1-13. DOI: 10.1109/TCDS.2018.2795343. https://ieeexplore.ieee.org/document/8263227/.

[12] Feng, Y., Jia, Q., Wei, W. (2018). A Control Architecture of Robot-Assisted Intervention for Children with Autism Spectrum Disorders. Journal of Robotics, vol. 2018, Article ID 3246708, 12 pages. https://doi.org/10.1155/2018/3246708 y https://www.hindawi.com/journals/jr/2018/3246708/.

[13] Gaudla, S., Bryson, J. (2018). The extended ramp model: A biomimetic model of behaviour arbitration for lightweight cognitive architectures. Cognitive Systems Research, Volume 50, Pages 1-9. https://doi.org/10.1016/j.cogsys.2018.02.001.

[14] González, A. Alves, M., Viana, G., Carvalho, L., Basilio, J. (2018). Supervisory Control-Based Navigation Architecture: A New Framework for Autonomous Robots in Industry 4.0 Environments. IEEE Transactions On Industrial Informatics, Vol. 14 (4): 1732-1743.

[15] Grillner S. Georgopoulos A., Jordan L. (1997). Selection and Initiation of Motor Behavior. En Neurons, Networks and Motor Behavior, eds. Stein P., Grillner S., Selverton A., Stuart D. The MIT Press.

[16] Herrero, I., Urdiales, C., Peula, J, Sandoval, F. (2017). CBR based reactive behavior learning for the memory-prediction framework. Neurocomputing. Volume 250, Pages 18-27. https://doi.org/10.1016/j.neucom.2016.10.075

[17] Hogg, D.W., Martin, F., Resnick, M. (1991). Braitenberg creatures. Technical report E&L Memo No. 13, MIT Media Lab. http://cosmo.nyu.edu/hogg/lego /braitenberg_vehicles.pdf.

[18] Jiménez J, Espinosa I. (2000). Arquitecturas de selección de acción inspiradas en el sistema nervioso de los vertebrados. Memorias del II simposio internacional en tecnologías inteligentes. ISIT 2000, Apizaco, Tlaxcala. http://sgpwe.izt.uam.mx/files/users/uami/jcjr/ArqSelapizaco00.pdf.

[19] Jiménez, J. (2018). Simulación de estrategias de búsqueda en animales con posibles aplicaciones en computación y robótica. Pistas Educativas, Vol. 39, Núm. 128, pp. 829-847. http://www.itcelaya.edu.mx/ojs/index.php/pistas /article/view/1168/984.

[20] Jiménez C., Johnson, J. (2017). SpikingLab: modelling agents controlled by Spiking Neural Networks in Netlogo. Neural Computing and Applications. Volume 28, Supplement 1, pp 755–764. doi: 10.1007/s00521-016-2398-1. https://www.researchgate.net/publication/303846142_SpikingLab_modelling _agents_controlled_by_Spiking_Neural_Networks_in_Netlogo

[21] Jones, J., Flynn, A. (1993). Chapter 9, Robot Programming (Subsumption architecture) in Mobile Robots, A. K. Peters, Ltd.

[22] Lazzeri, N., Mazzei, D., Cominelli, L., Cisternino, A., De Rossi, D. (2018). Designing the Mind of a Social Robot. Applied Sciences. Tomo 8, N.º 2: 302. DOI:10.3390/app8020302

[23] Lorenz, K. (1986). Biología del Comportamiento - Evolución y modificación de la conducta. Siglo XXI Editores, México D.F., México.

[24] Mataric, M. (2007). The Robotics Primer. MIT Press. http://neuron.tuke.sk /sabolpatrik/students/humanoids16/Cvicenia/RoboticsPrimer.PDF.

[25] Lee, G. y Chwa, D. (2018). Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. Intelligent Service Robotics, Volume 11, Issue 1, pp 127–138.

[26] Melidis, C., Iizuka, H. & Marocco, D. (2018). Intuitive control of mobile robots: an architecture for autonomous adaptive dynamic behaviour integration. Cognitive Processing 19(2): 245-264. doi: 10.1007/s10339-017-0818-5. https://www.ncbi.nlm.nih.gov/pubmed/28585090

[27] Muñoz, P., Moreno, M., Barrero, D., Ropero, F. (2018). MoBAr: a Hierarchical Action-Oriented Autonomous Control Architecture. Journal of Intelligent & Robotic Systems. pp 1–16. https://doi.org/10.1007/s10846-018-0810-z

[28] Prescott T., Redgrave P., Gurney K. (1999). Layered Control Architectures in robots and Vertebrates, Adaptive Behavior, Vol 7, pp. 99-127. http://eprints.whiterose.ac.uk/107032/1/Prescott%20Adaptive%20Behavior%201999%20preprint.pdf.

[29] Raglin, A., Metu, S., Howard, C. (2018). Understanding theoretical human information interaction, the development of a standard model using an agent based modeling framework, Proc. SPIE 10653, Next-Generation Analyst VI, 1065302; doi: 10.1117/12.2304533; https://doi.org/10.1117/12.2304533.

[30] Rodríguez, F., Matellán, V., Conde, M., Martín F. (2018). HiMoP: A three-component architecture to create more human-acceptable social-assistive robots. Cognitive Processing, 19 (2): 233-244. https://doi.org/10.1007/s10339-017-0850-5.

[31] Sakellariou, I., Kefalas, P., Tamatopoulou, I. (2008). Teaching Intelligent Agents using NetLogo. Proceedings of the ACM-IFIP IEEIII 2008 Informatics Education Europe III Conference, Venice, Italy. https://pdfs.semanticscholar.org/0d6f/2b8bc1015dda90e0d68939e388e43359d155.pdf.

[32] Tisue, S., Wilensky, U. (2004). NetLogo: Design and implementation of a multi-agent modeling environment. Proceedings ofAgent. https://ccl.northwestern.edu/papers/2013/netlogo-agent2004c.pdf.

[33] Walter, G. (1950). An imitation of life. Scientific American 182(5): 42-45. http://robotics.cs.tamu.edu/dshell/cs643/papers/walter50imitation.pdf.

[34] Tzafestas, S. (2018). Mobile Robot Control and Navigation: A Global Overview. Journal of Intelligent & Robotic Systems. Volume 91, Issue 1, pp 35–58. https://doi.org/10.1007/s10846-018-0805-9.

[35] Vanderelst, D., Winfield, A. (2018). An architecture for ethical robots inspired by the simulation theory of cognition. Cognitive Systems Research. Volume 48, Pages 56-66. https://doi.org/10.1016/j.cogsys.2017.04.002.

[36] Walter, G. (1951). A machine that learns. Scientific American 185(2): 60-63. http://robotics.cs.tamu.edu/dshell/cs643/papers/walter51learns.pdf.

[37] Zaharija, G., Grubač, A., Granić, A. (2014). LEaRN–LEgo Robot and Netlogo. Proceedings of CIET 2014, pp. 209-218. University of Split. https://www.researchgate.net/publication/273086230_LEaRN__LEgo_Robot_and_Netlogo.