View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by El Servicio de Difusión de la Creación Intelectual

CONTROLADORES INTELIGENTES APLICADOS A ROBÓTICA MÓVIL

Enrique A. Sierra, Alejandro A. Hossian, Gustavo E. Monte

Grupo de Estudios en Inteligencia Artificial (GEIA). Unidad Académica Confluencia. UTN
enriquesie@yahoo.com.ar, hossi@ciudad.com.ar

1. INTRODUCCIÓN

Los robots móviles son una de las tecnologías que más interés ha despertado en la industria por cuanto su posible aplicación a una gran diversidad de tareas de forma cooperante con el ser humano [1]. Conforme la tecnología se ha ido desarrollando ha crecido en importancia el concepto de "Autonomía", el cual sobrepasa el concepto de sistema automático. La autonomía es un requerimiento adicional importante. Para definir la autonomía, podemos recurrir a las definiciones suministradas por el etólogo Smithers [18] que dice: "La idea central del concepto de autonomía se identifica en la etimología del término: autos (propio) y nomos (ley o regla). Se aplicó por primera vez en la antigua Grecia para referirse a aquellas ciudades o estados que se regían por leyes propias en lugar de vivir acorde al poder de un gobierno externo. Es útil contrastar el concepto de autonomía con el de sistema automático. Los sistemas automáticos se autorregulan pero ellos no
establecen las leyes que sus reguladores intentan satisfacer. Estas leyes les son suministradas o están inmersas en su construcción. Los sistemas automáticos son capaces de conducirse a lo largo de un camino corrigiendo y compensando los efectos de las perturbaciones externas. Los sistemas autónomos son capaces de generar por ellos mismos las leyes y estrategias con las que regularan su comportamiento: se autogobiernan y se autorregulan. Determinan el camino a seguir y se conducen sobre él". Esta definición recoge la cuestión esencial, para ser autónomo primero hay que ser automático. Esto implica sentir el entorno y ejercer acciones sobre él de manera beneficiosa para el agente y las tareas que debe desarrollar. Pero la autonomía va más allá que el automatismo, porque se supone que la base de autorregulación se genera desde la propia capacidad del agente de componer y adaptar sus principios de comportamiento. Más aún, el proceso de construcción y adaptación es algo que tiene lugar mientras el agente opera en su entorno. Para conseguir esta propiedad el robot requiere una serie de capacidades que en gran medida se agrupan bajo el concepto de inteligencia.

2. PROBLEMÁTICA DE LOSCONTROLADORES INTELIGENTES

La inclusión de controladores inteligentes (control fuzzy, redes neuronales, algoritmos genéticos, control neuro-fuzzy, etc.) no asegura un comportamiento adecuado en todas las situaciones, en otras palabras, es una manera de facilitar la implementación de los controladores en el robot. Para asegurar un comportamiento "inteligente" es necesario dotar de algoritmos que aseguren la resolución de los problemas derivados con los que se puede encontrar el robot. Uno los principales problemas cuando se implementan controladores inteligentes, mediante la *Toolbox* de Matlab, es la cantidad de recursos necesarios para poder ejecutar el código. La inclusión de computadores potentes y de programas más elaborados hace que este inconveniente desaparezca, a costa de necesitar una potencia de cálculo superior.

3 SIMULACIÓN DE UN ROBOT MÓVIL

La simulación del comportamiento del robot cuando se le proporciona el control inteligente es fundamental para la validación del controlador. Paralelamente al desarrollo sobre sistemas reales,

actualmente es posible utilizar herramientas software mediante las cuales el ciclo de análisis, diseño, prototipado y desarrollo de sistemas de control puede verse acelerado [16]. Entre una de las más utilizadas encontramos el entorno Matlab-Simulink el cual, en conjunción con el conjunto de *Toolbox* y *Blockset*, proporcionan una herramienta potente para analizar, diseñar y, testear sistemas tanto desde un punto de vista clásico como mediante las técnicas expuesta anteriormente (redes neuronales, *fuzzy* y algoritmos genéticos). Adicionalmente al entorno Simulink se le ha dotado de la capacidad de generación de código de los sistemas simulados para un conjunto de targets concretos (sistemas PC, microcontroladores, sistemas empotrados y DSP). Dicha capacidad hace este entorno muy adecuado para la herramienta que se propone, ya que permite acelerar los desarrollos al tiempo que proporciona un basto entorno de sistemas susceptibles de ser simulados y controlados con el mismo (desde una simulación de un sensor hasta el comportamiento de un conjunto de robots en un entorno común). En concreto existe un simulador de robot móvil Khepera para Matlab, que se explica en el punto 3.2, el cual permite desarrollar estrategias de control tanto de modo simulado como conectado al robot real. Estas plataformas permitirán realizar la labor de investigación en un entorno más controlado y de menor costo temporal, en paralelo al desarrollo en el robot real del entorno necesario para su implementación.

4 HERRAMIENTA DESARROLLADA

La herramienta que se ha desarrollado está basada en el simulador KiKS y permite, de forma modular, la incorporación de distintos controladores y de funciones necesarias para el cálculo del comportamiento del robot.

Para ello se han programado distintas funciones y bloques que permite el cálculo de los parámetros del robot a partir de la información proporcionada por los sensores y encoders. Estos módulos permiten desacoplar la tarea de cálculo del controlador con la de cálculo de posición y orientación, haciendo más flexible la incorporación de nuevos comportamientos.

Los módulos principales que se han programado son los siguientes:

- Almacenamiento del recorrido seguido por el robot en una matriz.
- Seguimiento y representación de las velocidades de cada rueda, así como del ángulo de orientación del robot en función del tiempo.
- Almacenamiento y representación de la medición de distancias de cada sensor de infrarrojos en función del tiempo.
- Evolución de la velocidad lineal del robot seguida en función del tiempo.
- Calculo de la posición en función de la información suministrada por los encoders de cada rueda.

Cada uno de estos módulos puede ser llamado en el programa principal, facilitando la implementación del código del programa y haciendo más flexible el diseño de nuevos comportamientos.

5 IMPLEMENTACIÓN DE LA HERRAMIENTA

Dentro de este apartado tenemos dos puntos principales, por un lado los controladores y por el otro los comportamientos que podemos conseguir con la configuración de dichos controladores.

5.1 CONTROLADORES INTELIGENTES

5.1.1 Controladores fuzzy

Para implementar este tipo de controladores la herramienta básica que se ha utilizado para generar los módulos es el *Fuzzy Logic Toolbox* de Matlab. Mediante esta herramienta se generan las funciones de

pertenencia de las entradas y de las salidas y se desarrolla la relación que existe entre las mismas. En el código del programa controlador implementado se carga éste módulo para ser utilizado cuando se necesita conocer las salidas a partir de la información sensorial de las entradas. Para validar el módulo y verificar que funciona correctamente se han probado las soluciones propuestas por Godjevac [6] [7], Zhang [21] y Maaref [9].

Ejemplo. Implementación de un controlador fuzzy para evitar obstáculos

Lo primero que tenemos que hacer es mediante el *Fuzzy Logic Toolbox* de Matlab generarnos las funciones de pertenencia de las entradas y las salidas, así como la relación que existe entre ellas. Para ilustrar este ejemplo se ha desarrollado un regulador fuzzy basado en la solución propuesta por Godjevac [6] [7].

En las figura 5 se muestran las entradas y salidas consideradas. Lo que se ha hecho es agrupar los sensores en izquierda, derecha, frente y atrás, de manera que se facilitan las reglas entre las entradas y salidas que hay que considerar más tarde. Una solución más completa en la que se tienen en cuenta todos los sensores es la que propone Zhang [21].

Como salidas se ha considerado los motores izquierdo y derecho, es decir, lo que se ha hecho es unir directamente las entradas con las salidas por medio de relaciones entre ambas. En este sentido una solución más elaborada es la que propone Maaref [9], en la que la salida que proporciona el controlador fuzzy la orientación y la velocidad global que hay que darle al robor para evitar el obstáculo.

La siguiente figura muestra las reglas que se deben cumplir en las entradas y salidas, estás reglas, como se ha comentado anteriormente, están basadas en la solución propuesta por Godjevac [6][7] pero modificándolo con 4 reglas más, de manera que podamos también seguir por un pasillo y evitar obstáculos en movimiento que provengan de la parte trasera.

Por último en las figuras 6 y 7 se representa las funciones de pertenencia de las entradas y de las salidas respectivamente.

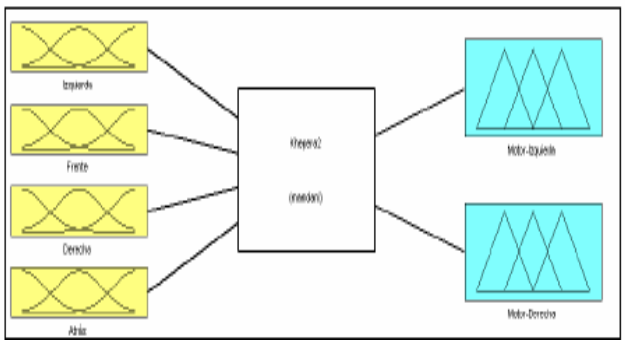

Fig. 5. Entradas y salidas consideradas en el controlador fuzzy de Godjevac

	Distancias				Velocidades	
	Izquierda	Frente	Derecha	Atrás	Motor Izquierdo	Motor derecho
Regla 1	Grande	Grande	Grande	Grande	Adelante medio	Adelante medio
Regla 2		Pequeña			Adelante medio	Atrás medio
Regla 3	Grande	Grande	Grande	Pequeña	Adelante rápido	Adelante rápido
Regla 4	Pequeña	Grande	Grande		Adelante rápido	Adelante lento
Regla 5	Grande	Grande	Pequeña		Adelante lento	Adelante rápido
Regla 6	Pequeña	Grande	Pequeña		Adelante lento	Adelante lento

Tabla 1. Relación entre las entradas y salidas en el regulador fuzzy de Godjevac modificado

5.1.2 Redes neuronales

La implementación de este tipo de controladores, para robots tan sencillos como el que nos ocupa, consiste en el entrenamiento de una matriz de pesos en los cuales a partir del comportamiento deseado se obtienen las entradas de dicha matriz. En este caso la implementación de estas matrices es muy similar a la idea de conectar los sensores con los actuadores propuesta por Braitenberg [2].

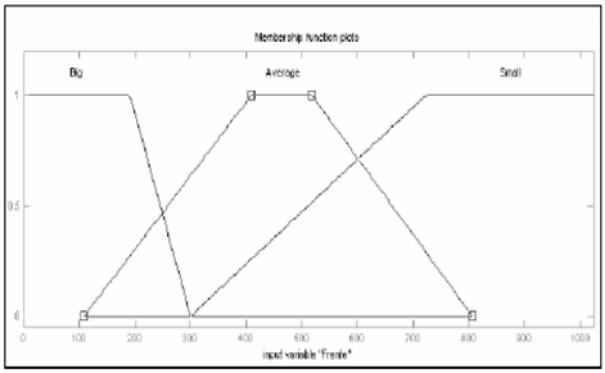

Fig. 6. Funciones de pertenencia de las entradas

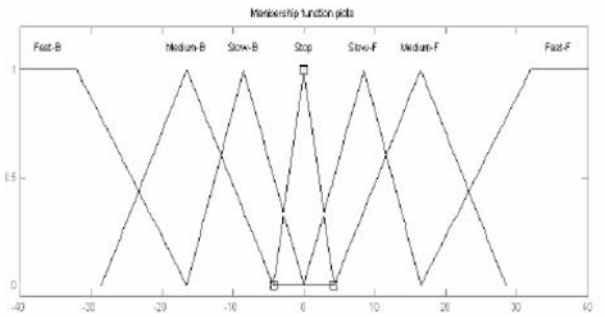

Fig. 7. Funciones de pertenencia de las salidas

5.1.3 Braitenberg

Con los vehículos de Braitenberg [2] el autor propuso estudiar los principios de la inteligencia construyendo sucesivamente agentes más complejos. Con este planteamiento inicial es fácil construir vehículos que reaccionen de una manera determinada dependiendo de la información sensorial que reciben en cada instante.

La idea principal es la de conectar los sensores con las actuadores. Cada medición del sensor se pondera por un peso, que será el que consiga la contribución de esa medida en la velocidad de las ruedas. En los siguientes apartados se muestran como se han ponderado los pesos de la medida de cada sensor para conseguir el comportamiento deseado.

5.1.3.1 Evitar obstáculos

SENSOR	S0	S1	S2	S3	S4	S5	S6	S7	LIMITE
Motor 1 (izq.)	0.2	0.2	0.3	-0.72	-0.46	-0.2	0.26	0.15	0.30
Motor 2 (der.)	-0.26	-0.51	-0.77	0.31	0.2	0.2	0.15	0.26	0.30

Tabla 2: Matriz de pesos del vehículo de Breitenbeg en el comportamiento de evitar obstáculos

5.1.3.2 Empujar objetos

SENSOR	S0	S1	S2	S3	S4	S5	S6	S7	LÍMITE
Motor 1 (izq.)	2	3	4	4	3	2	0	0	10
Motor 2 (der.)	-2	-3	-4	-4	-3	-2	0	0	10

Tabla 3: Matriz de pesos del vehículo de Breitenbeg en el comportamiento de empujar objetos

5.1.3.3 Seguir una pared

SENSOR	S0	S1	S2	S3	S4	S5	S6	S7	LÍMITE
Motor 1 (izq.)	0	2	3	3	0	0	0	0	5
Motor 2 (der.)	0	0	-3	-3	0	0	0	0	7

Tabla 4: Matriz de pesos del vehículo de Breitenbeg en el comportamiento de seguir una pared

5.1.3.4 Exploración del entorno

SENSOR	S0	S1	S2	S3	S4	S5	S6	S7	LÍMITE
Motor 1 (izq.)	2	4	6	6	4	2	4	4	10
Motor 2 (der.)	-2	-4	-6	-6	-4	-2	4	4	10

Tabla 5: Matriz de pesos del vehículo de Breitenbeg en el comportamiento de exploración del entorno

5.2 COMPORTAMIENTOS

Mediante los controladores mostrados en el punto anterior se ha implementado los siguientes comportamientos básicos: Evitar obstáculos, Llegar a objetivos, Seguir paredes, Atracción por la luz, Empujar objetos. Estos comportamientos básicos pueden ser combinados para conseguir un comportamiento más inteligente, de esta manera podemos tener: Evitar obstáculos + Llegar a objetivos ó Evitar obstáculos + Atracción por la luz. Sin embargo la combinación de otros comportamientos puede ser no deseada, por ejemplo: Evitar obstáculos + Empujar objetos. La implementación de la combinación de varios comportamientos puede implementarse mediante el módulo de un controlador fuzzy, una red neuronal o de otras tecnologías inteligentes.

5.2.1 Ejemplo. Evitar obstáculos

Para desarrollar este comportamiento se han implementado varios controladores que proporcionan el objetivo deseado, como se ha visto en 5.1.1. Para tener una mayor dificultad en este apartado se le ha añadido, tanto en simulación como en la realidad, un objeto en movimiento que dificulta la evolución del robot sobre el entorno, y que permite validar los controladores con el comportamiento de evitar obstáculos tanto estáticos como en movimiento.

El ejemplo presentado tiene una duración de 60 segundos, en los primeros 30 segundo se trabaja con un controlador fuzzy y durante los últimos 30 segundos con una red neuronal implementada mediante matrices de ponderación de las medidas de los sensores conectadas directamente a la salida de los motores.Para interpretar los resultados de cada uno de los controladores se han añadido los módulos que muestran una representación de los resultados obtenidos.

En las siguientes figuras se muestran los resultados de la prueba explicada anteriormente. En ellos puede apreciarse un comportamiento diferente del robot en función del controlador aplicado. Mientras que el controlador fuzzy implementado es más agresivo, acercándose más a las paredes (como demuestra la figura 11) y teniendo una velocidad media superior al controlador de red neuronal, éste tiene un margen de seguridad más grande ya que evita el contacto con los obstáculos mucho antes.

5.2.2 Ejemplo. Llegar a objetivos

Para implementar este comportamiento se utilizada la información de los encoders de las ruedas para saber en todo momento la posición del robot en el entorno de trabajo. Para ello se ha tenido que realizar un módulo que transforma la medida de los sensores en la nueva posición que va a ocupar el móvil. En la figura 9 puede apreciarse el resultado del comportamiento de llegar a 5 objetivos, conseguido con un controlador fuzzy de 6 reglas. Las medidas de las distancias recorridas por el robot se han tomado con el módulo mencionado anteriormente. Éste módulo puede utilizarse en sucesivos diseños y de esta forma no perdemos flexibilidad en la implementación de los comportamientos deseados.

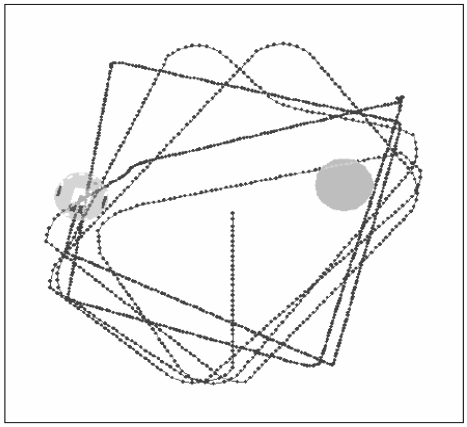

Fig. 8. Algoritmo para evitar obstáculos con controladores fuzzy distintos

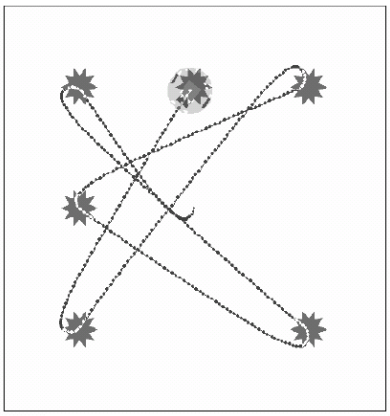

Fig. 9: Algoritmo para llegar a 5 objetivos

6. CONCLUSIONES

Mediante esta herramienta, de carácter modular, se facilita el diseño, implementación y pruebas de controladores inteligentes, aplicados al caso concreto de un robot móvil con movimiento diferencial. Al utilizar, tanto para la simulación como para el diseño, las *Toolbox* de Matlab, se permite el uso de toda la capacidad de programa, no limitándose a los módulos implementados, si no que es posible crear otros nuevos, así como sacar otro tipo de resultados que el usuario considere más interesantes. Este entorno hace que el desarrollo y diseño de pruebas sea muy sencillo y además pueda ser utilizado como una herramienta muy válida en el ámbito docente, debido a la facilidad de uso y por desarrollarse en un entorno de programación conocido y accesible por los alumnos como es Matlab.

REFERENCIAS

[1] Blanes J.F., (2000) *Percepción yRepresentación del entorno en robótica móvil.* Tesis Doctoral. Universidad Politécnica de Valencia.
[2] Braitenberg V., (1984) *Vehicles: Experiments in Synthetic Psycology.* MIT Press. Cambridge.
[3] Brooks R.A., (1986) *A robust layered controlarchitecture for a mobile robot.* IEEE Journal of robotics and automation nº2, vol 1. Pag 14-23.
[4] Fukuda T., (1999) *Computational Intelligence for the Robots and Automation.* Procedings IEEE ISIE'99.
[5] Fukuda T., Hasegawa Y., (2000) *Learning and adaptation for Intelligent Automation.* 4th IFAC International Symposium SICICA 2000. Buenos Aires, Argentina.
[6] Godjevac J., (1995) *Comparative study of fuzzy control, neural network control and neuro-fuzzy control.* Research Index.
[7] Godjevac J., (1995) *A Learning Procedure for a Fuzzy System: Application to Obstacle Avoidance.* Research Index.
[8] Ledin J.A. (1999) *Hardware in the loop simulation.* Embedded Systems Programming.
[9] Maaref H., Barret C., (2000) *Sensor-based fuzzynavigation of an autonomous mobile robot in an indoor environment.* Control Engineering 8: 757-768
[10] Martín del Brío B., Sanz Molina A., (2001) *Redes Neuronales y Sistemas Borrosos.* Ra-Ma.
[11] Mondana F., Franzi E., Ienne P., (1994) *Mobilerobot miniaturisation : A tool for investigation in control algorithms.* Third Int. Symposium on Experimental Robotics.
[12] Newell A., (1981) *The Knowledge Level.* Journal od Artificial Intelligence. 18 (1), pp 87-121
[13] Nilsson T., *KiKS is a Khepera Simulator.* Master Thesis. Umeå University.
[14] Oriolo G., Ulivi G., Vendettelli M., (1997) *Fuzzy maps : a new tool for mobile robot perception and planning.* Journal of Robotics Systems, 14(3).
[15] Penrose R., (1990) *The Emperor's new mind.*Oxford Univerty Press. Oxford.
[16] Simarro R., Navarro J. L., (2001) *Simulación de una terminal marítima de contenedores.* Workshop en modelado y simulación de sistemas. Bellaterra.
[17] Simó J.E., (1997) *Una arquitectura basada en motivaciones para el control de robots móviles.*
Tesis Doctoral. Universidad Politécnica de Valencia.
[18] Smithers T., (1991) *Taking eliminative materialism seirously: a metodology for autonomous systems research.* Procedings for the first European Conference on Artificial Life.
[19] Steels L., (1993) *Intelligence, Dynamics and Representation.* Proceedings of the NATO
Advanced Study. Institute on the Biology and Technology of Intelligence and Autonomous
Agents. Trento. Italy.
[20] Steels L., Brooks R., (1994) *The Artificial Life, Route to Artificial Intelligence. Buildings*
Situated Embodied Agents. Lawrence Erlbaum Ass., New Haven.
[21] Zhang J., Wille F., Knoll A., (1996) *Fuzzy Logic Rules for Mapping Sensor Data to RobotControl.* Research Index.