XXV Jornadas de Automática
Ciudad Real, del 8 al 10 de septiembre de 2004

AUTOGUIADO DE ROBOTS MÓVILES MEDIANTE REDES NEURONALES

Mª Concepción Marcos Provecho
Escuela de Ingenierías Industrial e Informática. Universidad de León
comarpro@terra.es

Roberto Guzmán Martínez
Escuela de Ingenierías Industrial e Informática. Universidad de León
eiiinf@unileon.es

Rocío Alaiz Rodríguez.
Dpto. Ingeniería Eléctrica y Electrónica. Universidad de León
dierar@unileon.es

Resumen

El objetivo de este trabajo es la implementación de una estrategia de autoguiado de un robot móvil en entornos desconocidos empleando como núcleo de decisión una red neuronal. Se ha desarrollado un entorno en Matlab para la generación de trayectorias de entrenamiento, el entrenamiento propiamente dicho y la simulación de la red. En la fase de desarrollo se han tenido en cuenta las peculiaridades del microbot PICBOT3, microbot sobre el que se implementa la red neuronal y por tanto realizar las aproximaciones necesarias.
Los resultados experimentales ponen de manifiesto la capacidad de generalización de la red, consiguiendo alcanzar metas distintas desde orígenes diferentes sin colisionar con los obstáculos del camino.

Palabras Clave: Robots Móviles, Navegación Autónoma, Redes Neuronales.

1 INTRODUCCIÓN

La navegación autónoma de robots en entornos desconocidos constituye uno de los retos tecnológicos más importantes en el campo de la robótica móvil. Actualmente, un gran número de investigadores centran sus estudios en el desarrollo de técnicas de control y navegación de robots. De forma resumida, el problema que se plantea puede enunciarse de la siguiente manera: un robot móvil se encuentra situado en un determinado ambiente, que contiene una serie de obstáculos y una meta. Se trata de controlar el robot para que navegue por el entorno, evitando los obstáculos y alcanzando la meta. Existen varios enfoques para intentar resolver este problema, por un lado, se está trabajando en mejorar los métodos clásicos de planificación de trayectorias, como los grafos de visibilidad, los diagramas de Voroni, etc. [7]; y por otro, las investigaciones se encaminan hacia sistemas de control inteligentes, basados en técnicas de Inteligencia Artificial como son la Lógica Fuzzy [1],[6] y las Redes Neuronales [5],[8]. Estos sistemas han permitido abocar muchos de los problemas de control gracias a su capacidad de emular algunas características propias de los humanos.
En este trabajo se pretende aprovechar la capacidad de generalización de las redes neuronales para resolver los problemas que se plantean en la navegación de un robot móvil en un ambiente desconocido. Es un trabajo de simulación para su posterior implantación en el microbot PICBOT3.

Este artículo está dividido en seis partes: en la primera se presenta el problema objeto de estudio; en la segunda se describen las características del microbot utilizado; las condiciones en que se realizaron las pruebas experimentales se describen en la sección tres y en la cuarta se detallan las características de las redes neuronales empleadas; en la quinta se muestran los resultados obtenidos en el trabajo experimental; y en la última se exponen las conclusiones y las líneas de trabajo futuro.

2 DESCRIPCIÓN DEL MICROBOT

El microbot elegido para realizar las pruebas experimentales es el PICBOT 3 de la empresa Microsystems Ingeeniering mostrado en la Figura 1. Es un robot flexible y modular, cuyo sistema de control está basado en el microcontrolador PIC16F876. Su sistema de tracción es de tipo diferencial, con una rueda loca y dos ruedas motrices accionadas por servomotores FUTABA 3003. A este microbot se le han añadido cinco pares emisor-receptor de ultrasonidos en su parte delantera, para

detectar la presencia de obstáculos, y una brújula o compás magnético que le permite orientarse dentro del entorno de navegación.

Figura 1: Microbot PICBOT3

Para que un robot pueda navegar en un ambiente desconocido tiene que ser capaz de efectuar unos movimientos tan intuitivos como los que realizaría una persona que se encontrara en su misma situación. Los humanos no tenemos un mapa o un sistema de coordenadas en base al cual navegamos, simplemente observamos el entorno y vamos decidiendo en cada instante la acción que hemos de ejecutar para continuar avanzando hacia nuestro objetivo. Lo que se pretende es que el robot actúe de igual forma que una persona, es decir, que capte información sobre su entorno a través de los sensores y en función de ella "decida" que acción debe llevar a cabo para continuar su camino. No se puede pretender que el robot aprenda todas las situaciones que posiblemente se le vayan a presentar durante la navegación. A nivel de memoria se recargaría el sistema con demasiada información que se puede considerar redundante. Para resolver este problema las personas recurrimos al proceso de *generalización,* que nos permite solucionar un caso particular, cuando se asemeja a un caso general conocido. Una de las principales características de las redes neuronales es precisamente su capacidad de generalización, por tanto, son una herramienta ideal para proporcionar al robot la capacidad de decidir en cualquier situación, incluso cuando dicha situación no se le haya presentado anteriormente.

3 MODELADO DEL MICROBOT

El ambiente donde navega el robot es un cuadrado de 5,40 metros de lado, representado por una imagen de 120 x 120 píxeles. Por lo tanto, cada píxel equivale a 4,5 cm reales. En el interior del ambiente se distribuyen de forma aleatoria varios obstáculos que el robot tendrá que evitar.

Las variables de interés para el control del robot son las coordenadas (*x*,*y*) y el ángulo *phi* que representan respectivamente la posición y orientación del robot dentro del ambiente. Las coordenadas se van a expresar en píxeles, considerando como Sistema de Referencia el indicado en la Figura 2, donde el Origen se encuentra en la esquina superior izquierda de la imagen, el *eje x* aumenta de izquierda a derecha y el *eje y* de arriba hacia abajo. El ángulo *phi* se aproxima a una de los ocho posiciones señaladas en la Figura3.

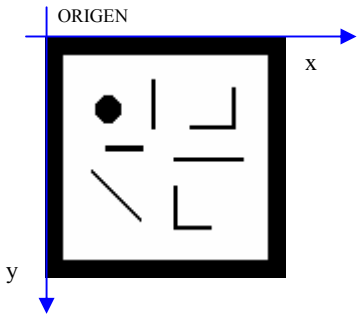

Figura 2: Sistema de Referencia

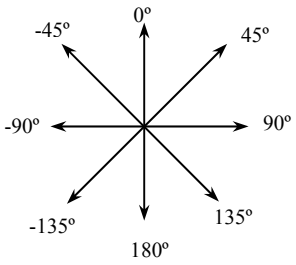

Figura 3: Ángulo de orientación del robot

Otra variable importante es el ángulo *theta* de dirección de la meta, que representa el giro que debe efectuar el robot para quedar perfectamente orientado hacia la meta. El valor de este ángulo puede variar entre –179º y 180º, considerando positivos los ángulos efectuados en sentido horario y negativos los realizados en sentido contrario. En la Figura 4 se representa un ejemplo de ángulo θ negativo.

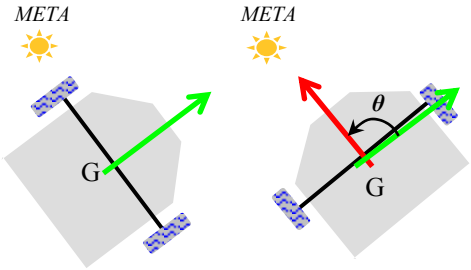

Figura 4: Ángulo de orientación a meta

Con el fin de facilitar el aprendizaje de la red, se va a codificar el rango de valores de este ángulo, de acuerdo con los siguientes criterios:

- Si el ángulo está comprendido entre -11º y –179º se codifica como –1.

- Si está comprendido entre –10º y +10º se codifica como 0.
- Si está comprendido entre +11º y +180º se codifica como 1.

Por tanto, el ángulo *theta* se representará por los valores –1,0, ó 1.

El robot está equipado con cinco sensores de ultrasonidos que se encuentran uniformemente distribuidos en su parte delantera (ver Figura 5). Estos sensores tienen la capacidad de detectar la presencia de un obstáculo y medir la distancia a la que se encuentra. En este caso, se ha considerado que únicamente son de dos estados, de forma que si se detecta un objeto se activará la señal lógica 1 y si no la señal activa será un 0. El campo de acción de los ultrasonidos es un ángulo de unos 33º de apertura y 22.5 cm de alcance, que se aproximará en la imagen del entorno por medio de las plantillas de la Figura 5.

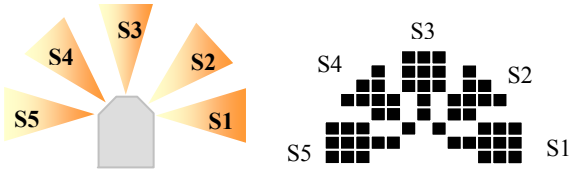

Figura 5: Disposición de los sensores en el microbot

En este trabajo se ha considerado que el robot sólo puede realizar tres movimientos básicos (Figura 6), que son:

- Avance sin giro.
- Giro de 45º y avance.
- Giro de – 45º y avance.

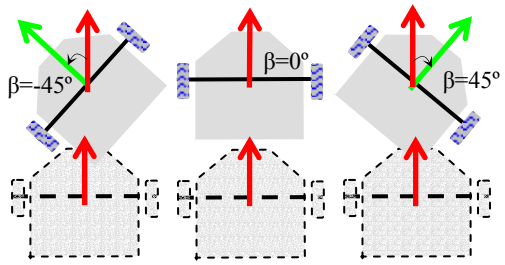

Figura 6: Acciones de Control

En cada una de estas acciones el avance que realiza el robot es de 4,5cm.

4 RED NEURONAL

La red neuronal elegida para el desarrollo del sistema de control del robot es un MLP (Multi-Layer Perceptron), debido a que este tipo de redes por sus características son muy válidas para resolver problemas de clasificación como el que nos ocupa. A esto cabe añadir que presentan una arquitectura relativamente sencilla para su implementación en el robot. La Figura 7 muestra la estructura de la red empleada, compuesta por: seis entradas, cuatro neuronas en la capa oculta y tres en la de salida; todas ellas con función de activación de tipo sigmoidal. Este tipo de función hace que los valores de salida de la red sean números decimales comprendidos en el intervalo [0 1], el bloque WTA (Winner Take All) tiene como función poner la salida de mayor valor a 1 y las otras dos a 0.

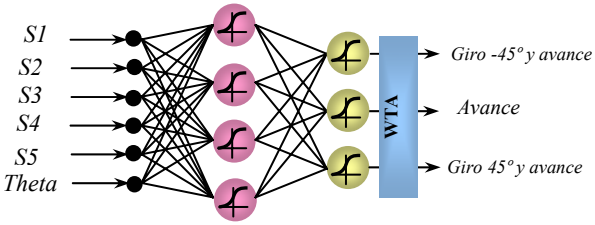

Figura 7: Estructura del MLP

A pesar de que la arquitectura de esta red es bastante sencilla, el microcontrolador en el que se va a implementar presenta unas características demasiado limitadas para realizar las operaciones que conlleva el funcionamiento de la red. Por esta razón, se ha entrenado también un Perceptrón Simple, cuya arquitectura puede verse en la Figura 8.

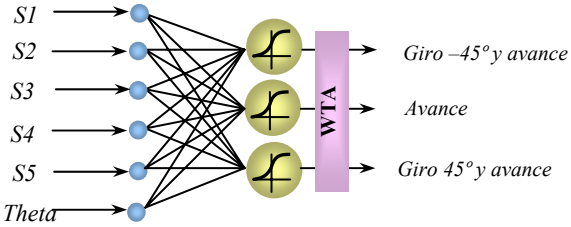

Figura 8: Estructura del Perceptrón Simple

Como puede observarse en las Figuras 7 y 8, los *vectores de entrada* a la red están compuestos por seis elementos, los cinco primeros corresponden al estado de los sensores ultrasónicos y pueden tomar valores 1 y 0, dependiendo de si están o no activos. El sexto es el ángulo de dirección de la meta codificado, cuyos valores pueden ser –1,0 ó 1.

El *vector de salida* está formado por tres elementos, correspondientes a cada una de las acciones básicas de control que puede realizar el robot. Los valores de estos elementos están codificados, de forma que el correspondiente al movimiento que se debe ejecutar es 1 y los otros dos 0. Así , la red ofrecerá como respuesta el vector (1,0,0) si la acción de control es girar – 45º; (0,1,0) si es avanzar sin girar y por último (0,0,1) si es girar 45º.

5 RESULTADOS EXPERIMENTALES

5.1 GENERACIÓN DE TRAYECTORIAS DE ENTRENAMIENTO

Una de las características más destacables de las redes neuronales, es que aprenden a partir de *ejemplos*. Para poder llevar a cabo el proceso de entrenamiento es necesario recopilar una gran cantidad de datos, los suficientes para que la red pueda aprender a desempeñar la tarea para la que ha sido creada. En este caso, los ejemplos van a ser una serie de trayectorias como la que se muestra en la Figura 9, a partir de las cual se van a extraer todos los parámetros que componen los *vectores de entrada* y *salida* de la red.

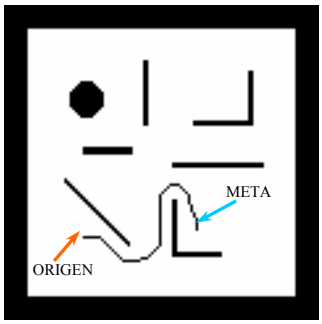

Figura 9: Ejemplo de Trayectoria de entrenamiento

Para generar estas trayectorias lo ideal sería poder captar la secuencia de movimientos que efectuaría una persona que guiase al robot. Como esto no es posible hay que recurrir a otras alternativas como utilizar una PDA con un interface adecuado para la obtención de datos [2] o simplemente, como se ha hecho en este trabajo, se dibujan en la imagen del ambiente ejemplos de trayectorias que podría realizar el robot y se programan una serie de funciones en Matlab que permitan extraer los datos de las trayectorias trazadas [3].

A la hora de dibujar una trayectoria hay que tener presente en todo momento que la única información sobre el entorno que tiene el robot es la proporcionada por los sensores y la dirección de la meta. También hay que considerar que las trayectorias deben ser continuas y que el robot no debe pasar más de una vez por el mismo punto. Es importante que las trayectorias generadas sean variadas, es decir, en ellas deben plantearse el mayor número de situaciones posibles en las que podría encontrarse el robot al navegar en un ambiente con obstáculos.

5.2 ENTRENAMIENTO Y TEST DE LA RED

Para la creación, entrenamiento y test de la red se ha utilizado la *toolbox Neural Networks* de Matlab, herramienta que dispone de una librería para trabajar de forma sencilla con redes neuronales [4]. Se puede implementar tanto un perceptrón multicapa (MLP) como arquitecturas más sencillas del tipo perceptrón.

El algoritmo completo de simulación se muestra en la Figura 10.

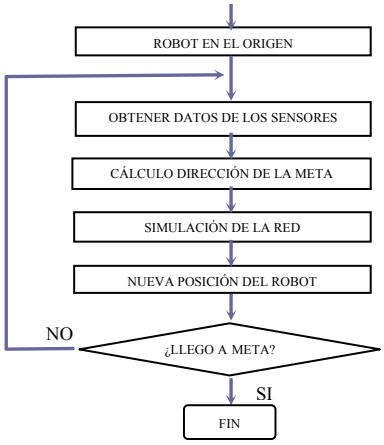

Figura 10: Algoritmo de simulación

Para poder realizar un análisis detallado de las trayectorias simuladas se ha realizado el Interface Gráfico de la Figura 11, utilizando la herramienta GUI de Matlab [3].

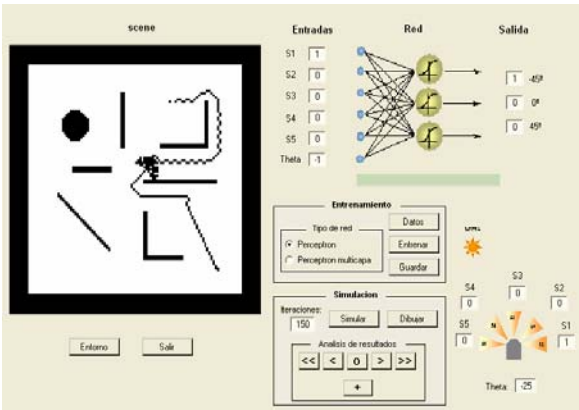

Figura 11: Interface de Simulación

Para simular una trayectoria el usuario debe en primer lugar, seleccionar el entorno en el que desea visualizarla y el tipo de red que quiere utilizar para realizar la simulación. Así mismo, tiene que introducir las coordenadas de los puntos donde desee situar el origen y la meta de la trayectoria. Una vez realizadas estas selecciones aparecerá dibujada en la imagen del entorno la trayectoria simulada. Este

Interface permite además de visualizar el recorrido realizado por el robot, analizar gráficamente el estado de los sensores, el ángulo de orientación a la meta y la acción de control proporcionada por la red para cada punto de la trayectoria.

5.3 IMPLEMENTACIÓN EN EL ROBOT

Para llevar a cabo la implementación de la red en el robot, se almacenan la matriz de pesos del perceptrón en la memoria EPROM no volátil, del microcontrolador. Esta matriz permitirá determinar las acciones de control (salidas de la red) en función del estado de los sensores y la orientación a la meta (entradas a la red).
Puesto que el microcontrolador utilizado es de 8 bits y sólo puede trabajar con números enteros comprendidos entre –127 y 128 será necesario normalizar los pesos de la red. Otro punto clave es la implementación de la función de transferencia de los nodos de salida puesto que la función sigmoidal no es fácilmente implementable en este microcontrolador. Por lo tanto es necesario realizar una segunda modificación que consiste en aproximar la función sigmoidal a una función lineal a tramos. Realizadas estas simplificaciones las únicas operaciones que conlleva el algoritmo de aproximación son sumas y productos con números enteros adaptados ya a la arquitectura del microcontrolador.

5.4 COMPARACIÓN DE RESULTADOS

Las primeras trayectorias se simularon en un entorno sin obstáculos, para comprobar que el robot se orientaba correctamente hacia la meta. Se hicieron pruebas situando el origen y el destino en diversos puntos del escenario, obteniendo resultados aceptables en cuanto a la optimización de la ruta tanto con el MLP y el Perceptrón Simple, como con el Perceptrón aproximado. Una vez verificada la orientación del robot, se añadieron una serie de obstáculos al entorno y se simularon nuevas trayectorias, con el fin de verificar si el robot detectaba y evitaba los objetos. En las Figuras 12, 13 y 14 se presenta un ejemplo de trayectoria, simulada con los dos tipos de red y con el algoritmo de aproximación, respectivamente.

Puede observarse que las respuestas, para las mismas coordenadas de origen y meta son diferentes, sin embargo, en los tres casos se cumple el objetivo propuesto, que es alcanzar la meta evitando colisionar con los obstáculos.

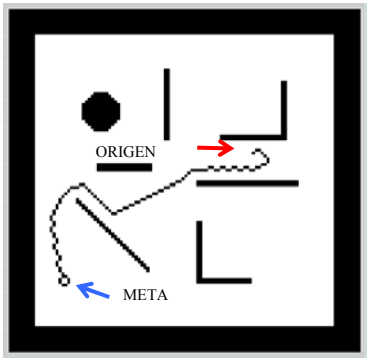

Figura 12: Trayectoria (MLP)

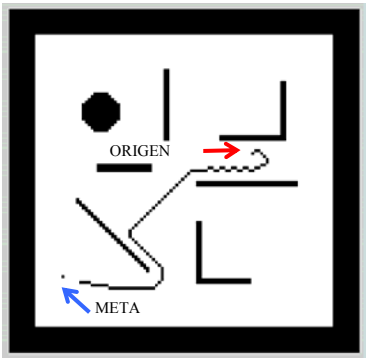

Figura 13: Trayectoria (Perceptrón Simple)

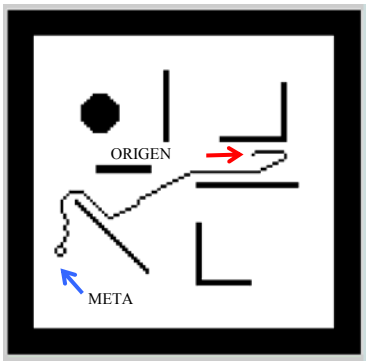

Figura 14: Trayectoria (Aproximación)

6 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se pueden resaltar como puntos más importantes los siguientes:

1. Se creó una librería en Matlab para conseguir el conjunto de datos de entrenamiento. Esta tarea resultó ser una de las más complejas e importantes del trabajo, ya que los ejemplos de entrenamiento son claves para el correcto aprendizaje de la red.
2. Se entrenaron dos tipos de redes: un Perceptrón Simple y un MLP, que se consideraron por su funcionamiento y por la sencillez de su topología

los más adecuados para el desempeño de la aplicación a realizar. El proceso de entrenamiento también se efectuó utilizando Matlab, encontrando las mayores dificultades en el ajuste del número de ciclos de entrenamiento y en el caso del MLP, del número de neuronas y capas ocultas. Fue necesario entrenar un gran número de redes para decidir los parámetros de entrenamiento que más se ajustaban a las necesidades de esta aplicación.

3. Se simularon numerosas trayectorias y se visualizaron los resultados proporcionados por las dos redes, comprobando que aunque las soluciones ofrecidas por ambas resultaron diferentes, en ambos fueron aceptables.
4. Se verificó que la red era capaz de generalizar, permitiendo simular trayectorias con distintas ubicaciones de orígenes y metas, y en entornos variados y desconocidos, es decir no presentados durante el entrenamiento.
5. Se realizaron las simplificaciones oportunas para poder implementar el Perceptrón en el microbot PICBOT 3, y se comprobó que el comportamiento, una vez hechas las aproximaciones, era robusto

No obstante no se trata de una línea de trabajo agotada, existiendo varias líneas de trabajo futuro entre las que pueden mencionar:

1. Ampliar las acciones de control que realiza el robot, para que pueda efectuar giros diferentes a 45º y ejecutar así movimientos más precisos.
2. Considerar que los sensores no sólo detectan la presencia o ausencia de un objeto sino que también miden la distancia a la que se encuentra el obstáculo, determinando así la cercanía o alejamiento de un objeto y no sólo su presencia y pudiendo optimizar la ruta calculada.
3. Añadir sensores en la parte trasera del robot con el fin de que pueda navegar marcha atrás sin peligro de colisión.

7 REFERENCIAS

[1] Brooks,R. "*A Robust Layered Control System for a Mobile Robot",* IEEE Journal or Robotics and Automation, Vol. RA-2,Nº1,pp 14-23.

[2] Chronis,G. Skubic, M.;*"Training a Neural Network for Navigation Behaviors"*. Dept. or Computer Engineering and Computer Science. University of Missouri-Columbia.

[3] Marcos Provecho, M.C. *"Autoguiado de Robots Móviles mediante Redes Neuronales"*. PFC . Escuela de Ingenierias Industrial e Informática. Universidad de León.

[4] MathWoks "Neural network Toolbox".

[5] Mitchell, R.J. , Keating,D.A.; "*Neural Network Control of Simple Mobile Robot"*. Department of Cybernetics. The University of Reading. Reading, UK.

[6] Nilsson,N. *"Inteligencia Artificial: una nueva síntesis"* Ed. McGraw-Hill. España 2001. ISBN 84-481-2824-9

[7] Ollero Baturone, A.; *"Robótica Manipuladores y robots móviles"*, Ed. Marcombo. 2001. ISBN: 84-481-0815-9.

[8] Yan Zhou, Wilkins, D., Cook Robert; *"Neural Network Control for A Fire- Fighting Robot",* Unyversity of Mississippi. Dept. of Computer and Information Science.