

Evación de obstáculos en tiempo real para robots móviles mediante redes neuronales

Real time avoidance for mobile robots with neural networks

Alexander Cerón*
Facultad de Ingeniería
Universidad Militar Nueva Granada.

Resumen- En el presente artículo se realiza una revisión acerca de la planeación de trayectorias en robots móviles y específicamente la aplicación de una red neuronal que toma las decisiones con respecto a la trayectoria de un robot móvil en la evación de obstáculos hasta llegar a un objetivo. El resultado de estos conceptos se muestra en una aplicación desarrollada en C++.

Este trabajo puede ser tomado como base para realizar juegos en los que intervienen objetos móviles que evitan colisiones o para implementar sistemas de detección de obstáculos para robots.

Palabras Claves- Robots móviles, planeación de trayectorias, redes neuronales, evación de obstáculos, computación grafica.

Abstract- This paper reviews the theory of robot motion planning and elaborates on an application in which neural networks are used to take decisions regarding the orientation of a robot in its search for a destination target with obstacle avoidance. The results of these concepts are implemented in an application developed with C++. This work can be used to develop games which mobile objects, that avoid obstacles, are involved or to implement obstacle detection systems for robots.

Keywords- Mobile robots, motion planning, neural networks, obstacle avoidance and computer graphics.

INTRODUCCIÓN

Un robot móvil es un dispositivo mecánico que esta provisto de sensores y actuadores, este

debe tener la habilidad de tomar decisiones mediante un sistema de control. Las decisiones a tomar están dadas de acuerdo a la información capturada del ambiente o entorno en el cual se encuentra el robot y estas tienen el objetivo de realizar una tarea determinada. Es prioritario definir dicha tarea y como debe ser realizada por el robot [8].

La planeación de trayectorias es uno de los problemas mas complicados en el desarrollo de robots autónomos ya que se debe tener el control del movimiento y la planeación de tareas.

Existen métodos que permiten hacer una evación rápida de obstáculos como el campo de fuerza virtual y el histograma de vector de campo [1][2] y otros que tienen en cuenta aspectos de la geometría del objeto como en [9] y [10].

Este trabajo esta enfocado en dotar a un robot móvil con la capacidad de tomar decisiones acerca de su propio movimiento, para ir desde una posición inicial a un punto final o meta, en un ambiente en el cual se encuentran obstáculos, los cuales se detectan mediante sensores de proximidad.

El comportamiento del robot es simulado mediante una aplicación de computador. El robot, simulado mediante la aplicación, utiliza una red neuronal para la toma de decisiones para realizar evación de obstáculos en planeación local de trayectorias. El objetivo es implementar

* Docente tiempo completo Universidad Militar Nueva Granada
Ingeniero de Sistemas de la Universidad Nacional de Colombia sede Bogota.
Magíster en Ingeniería Electrónica y de Computadores Universidad de los Andes. aceronc@umng.edu.co

estos conceptos en un robot móvil como el desarrollado en [3].

I. DEFINICIONES BÁSICAS

Se define W como un espacio Euclidiano, llamado espacio de trabajo, representado como R^N , con $N = 2$ en este caso.

Sea A un objeto rígido simple (el robot móvil) con geometría aproximadamente conocida, que se mueve en el espacio W sin restricciones cinemáticas en el movimiento de A (es decir A es un objeto de libre movimiento).

Sean B_1, \dots, B_q objetos rígidos distribuidos en W que actúan como obstáculos, y que tienen geometrías aproximadamente conocidas.

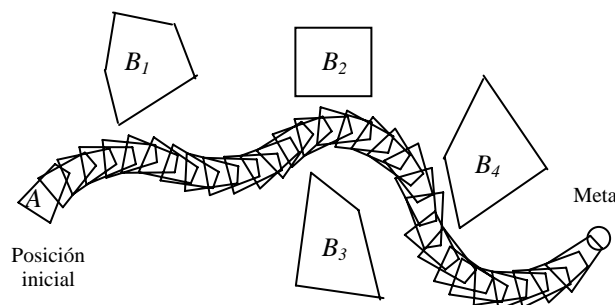


Figura. 1. Problema básico de planeación de trayectorias

El problema de planeación de trayectorias consiste en dada una posición inicial y orientación de A en W , generar un camino especificando una secuencia continua de posiciones de A evitando el contacto con los B_i , hasta llegar a una posición final o meta con una orientación deseada ver figura 1.

Para la solución del problema de planeación de trayectorias del robot A al evitar los obstáculos B_i , en el espacio de trabajo W existen dos enfoques, el de la planeación local y el de la planeación global.

- El problema básico de planeación global de trayectorias es generar una trayectoria, especificando una secuencia continua de posiciones y orientaciones de A evitando el contacto con los B_i , comenzando en una

posición y orientación inicial de A y terminando en la posición de la meta, reportando falla si dicha trayectoria no existe [8].

- El problema de la planeación local es un poco diferente ya que no se conoce de antemano la ubicación de los obstáculos, por lo tanto estos van a ser encontrados en tiempo real de acuerdo a la información que proviene de los sensores.

El problema se vuelve más complejo cuando existe interacción entre varios robots móviles tanto para planeación local como global y en especial cuando se realiza la planificación en un espacio tridimensional [6].

En este caso la información que capturan los sensores permite tomar la decisión con respecto al ángulo de rotación de móvil. Gracias a esta información es posible generar un mapeo del ambiente en el cual se encuentra el robot móvil.

II. SISTEMA DE DETECCIÓN

Muchos tipos de sistemas inteligentes, no únicamente robots móviles, dependen de las tecnologías de sensado dado que la manera de actuar de un robot móvil puede ser simplificada grandemente si el sistema de detección es suficientemente sofisticado [4]. Existen sensores visuales como las cámaras de video y algunos tipos de sensores no visuales como el sonar, el radar, los sensores inerciales y los de activación por presión [5].

El robot móvil a simular está provisto con tres sensores ultrasónicos ubicados en la parte delantera como se ve en la figura 2. Estos sensores se activan de acuerdo a la proximidad de un obstáculo, generando un patrón diferente de acuerdo al ángulo de choque.

En la aplicación se utilizó una descomposición de celdas del espacio de trabajo para implementar el espacio en el cual se van a colocar los obstáculos y se va a mover el robot.

En este caso se asume que la posición es conocida ya que en la práctica existen métodos para detectar la posición de un robot como el uso

de GPS, brújulas electrónicas o triangulación mediante sensores infrarrojos.

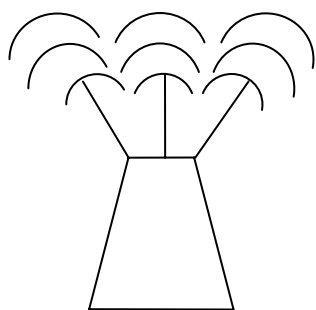


Figura. 2. Sistema de detección

III. RED NEURONAL APLICADA

Se utilizó una red neuronal de tres entradas y una salida, con función de activación sigmoideal en la capa intermedia y lineal en la de salida, ver figura 3 [7].

En la red neuronal, se interconectan varias unidades de procesamiento en capas, las neuronas de cada capa no se interconectan entre sí. Sin embargo, cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa, es decir, cada neurona transmitirá su señal de salida a cada neurona de la capa siguiente [7].

Las entradas a la red son la información de sensores ubicados a la izquierda, en el centro y a la derecha, y dichos sensores se activan de acuerdo a la proximidad de un obstáculo en dicha orientación.

La red va a tener una salida de acuerdo a esta entrada lo cual va a permitir corregir la posición y así evadir el obstáculo, ver figura 3.

Los datos de entrada son las distintas posibilidades en los sensores, ver tabla 1.

	Datos de entrada	Datos de salida	
1	000	0.0	No hay obstáculo
2	001	0.1	
3	010	0.2	
4	011	0.3	
5	100	-0.1	
6	101	-0.2	

7	110	-0.3	
8	111	-0.4	Máximo giro

Tabla 1. Patrones de entrenamiento

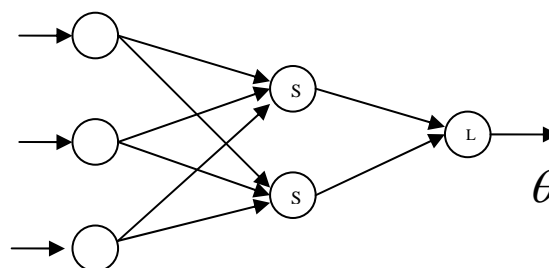


Figura. 3. Estructura de la red neuronal

La red neuronal se entreno convencionalmente en Matlab ® con los siguientes parámetros:

```
P=entrada'; %Patrones de entradas
T=salida'; %Patrones de salida
rango=minmax(P);
%N es el número de entradas
%rango es una matriz de Nx2 con los valores
%máximos y mínimos de las entradas
```

```
C1 = 2; % neuronas en la capa oculta
C2 = 1; % neuronas en la capa de salida
net = newff(rango, [C1 C2], {'tansig' 'purelin'},
'trainlm', 'learnngdm', 'mse');
[net,tr] = train(net, datosentrada, datossalida);
```

En los patrones del 2 al 7 la salida es un ángulo que se va aplicar a la orientación del móvil, si el obstáculo está a la izquierda entonces gira a la derecha con un giro progresivo determinado y si el obstáculo está a la derecha lo contrario de acuerdo al patrón que se obtenga en los sensores.

Los pesos de las conexiones obtenidas en el entrenamiento fueron introducidos en el programa de C++ como una red con la topología mencionada anteriormente. La función de activación que se implementó fue la sigmoideal ver figura 4, en el siguiente código se puede observar la implementación:

```
double tansig(double x)
{
    return 2/(1+exp(-2*x))-1;
}
```

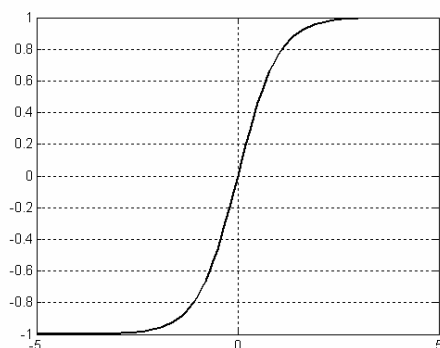


Figura. 4. Función de activación sigmoideal

IV. MOVIMIENTO DEL ROBOT

El robot se encuentra en un sistema de referencia del plano cartesiano en el cual se quiere cambiar su orientación a medida que se busca la meta.

Lo primero que se debe tener en cuenta es la traslación la cual se calcula mediante la siguiente formula:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Para la rotación del robot se aplica una matriz de transformación donde la nueva posición va a estar dada por una combinación lineal de funciones trigonométricas ver figura 5.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Figura. 5. Rotación de ejes para el robot

V. SOFTWARE DESARROLLADO

Como desarrollo práctico de este proyecto, se realizó un programa en lenguaje C++ en el cual se muestra el problema de la planeación local de

trayectorias mediante una simulación de un robot móvil, el cual es capaz de evadir obstáculos utilizando la red neuronal anteriormente nombrada, el programa está compuesto por los siguientes módulos:

1. Visualización del móvil.

Esta parte se encarga de mostrar el móvil en la ubicación y ángulo de orientación preciso. También se encarga de colocar los obstáculos.

2. Sistema para la evasión de trayectorias.

- Manejo de coordenadas y posicionamiento interno de los obstáculos.
- Comportamiento de seguimiento.
El móvil seguirá una trayectoria fija hasta encontrar la meta en esto intervendrá un temporizador.
- Comportamiento de evasión del obstáculo.

El móvil permanentemente está sensando el ambiente y el resultado de ese sensado se convierten en las entradas a la red neuronal.

3. Red neuronal

Dado la entrada actual de los sensores, se obtiene una salida la cual se convierte en un ángulo específico para modificar la orientación del móvil. La salida de la red neuronal se calcula mediante una función de activación sigmoideal y es aplicada al móvil con el objeto de afectar su trayectoria en caso de encontrarse con un obstáculo.

4. Opciones del programa

- Colocar obstáculos (menú)
- Colocar punto de partida (botón izquierdo del mouse)
- Colocar punto de llegada (botón derecho del mouse)

VI. RESULTADOS

El algoritmo presenta un buen desempeño, el móvil logra evadir los obstáculos y llegar a la posición final gracias a la red neuronal. Cuando la posición final se coloca dentro de un obstáculo, el móvil se queda orbitando alrededor del obstáculo lo cual es normal, pero cuando la meta se coloca muy cerca al móvil este se queda

orbitando sobre la meta, esto se puede corregir implementando restricciones simples que tengan en cuenta la proximidad y el ángulo recorrido en un tiempo determinado.

En las figuras 6 y 7 se muestran distintas posibilidades con las que cuenta el programa desarrollado.

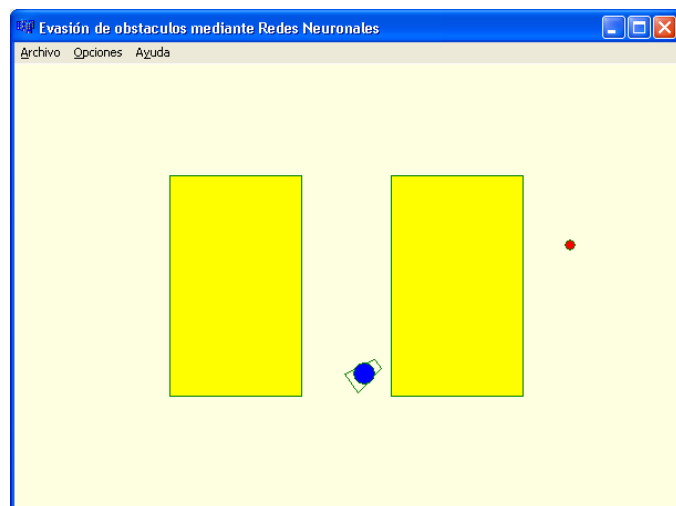


Figura 6. Ventana del programa (dos obstáculos)

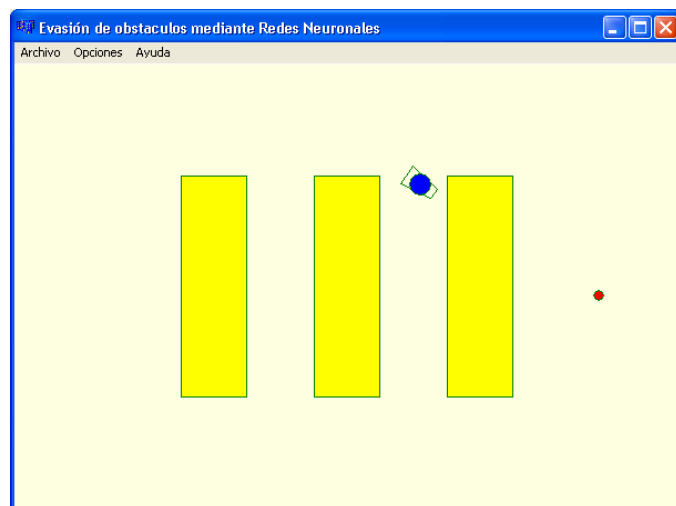


Figura 7. Ventana del programa (tres obstáculos)

VII. CONCLUSIONES

Las redes neuronales son una buena herramienta que puede ser aplicable a la planeación de trayectorias de tipo local dado que tienen la facilidad de trabajar con información imprecisa

como una señal proveniente de un sensor ultrasónico. Otra alternativa para tratar el problema es combinar las redes neuronales con lógica difusa lo cual se puede explorar en trabajos posteriores.

En un trabajo futuro se puede explorar la posibilidad de generación de trayectorias conociendo la ubicación de los obstáculos es decir planeación global. También se pueden incorporar otros móviles para que interactúen en un mismo ambiente.

Finalmente se espera en un futuro implementar este método en un robot móvil físico como el que se encuentra en [3], en este caso la información proveniente de los sensores físicos puede ser interpretada por una red más compleja que maneje una mayor cantidad de entradas incluyendo información proveniente de la cámara.

REFERENCIAS

- [1] Borenstein J. Real-time Obstacle Avoidance for Fast Mobile Robots. IEEE. Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept./Oct. 1989, pp. 1179-1187.
- [2] Borenstein J. The vector field histogram - Fast obstacle avoidance for mobile robots. IEEE Journal of Robotics and Automation Vol 7, No 3, June 1991, pp. 278-288.
- [3] Cerón A. Desarrollo de un Robot Móvil teleoperado con vision remota. IEEE Colombian Workshop on Robotics and Automation 2005. IEEE-Colombia. ISBN: 958-695-182-0.
- [4] Dudek G. Computational principles of mobile robotics, Cambridge University Press, 2000.
- [5] Everet H. R. Sensors for Mobile Robots. Theory and Application, Ak. Peters, 1995.
- [6] Fitch R. 3D Rectilinear Motion Planning with Minimum Bend Paths. International Conference on Intelligent Robots and Systems, 2001.
- [7] Haykin S. Neural Networks, a comprehensive foundation. Macmillan College Publishing Company, New York, 1994.
- [8] Latombe Jean-Claude, Robot Motion Planning, Kluwer Academic Publishers, 1991.
- [9] Lengyel J. Real-time robot motion planning using rasterizing computer graphics hardware, SIGGRAPH '90 Proceedings, 1990.

- [10] Song G. Randomized Motion Planning for Car-like Robots with C-PRM, Conference on Intelligent Robots and Systems, 2001.