Dyna
ISSN: 0012-7353
dyna@unalmed.edu.co
Universidad Nacional de Colombia
Colombia

Ramírez, Gustavo
Método de aprendizaje simple para navegación de minirobots móviles rodantes
Dyna, vol. 70, núm. 138, julio, 2003, pp. 59-66
Universidad Nacional de Colombia
Medellín, Colombia

MÉTODO DE APRENDIZAJE SIMPLE PARA NAVEGACIÓN DE MINIROBOTS MÓVILES RODANTES

GUSTAVO RAMÍREZ

Escuela de Ingeniería Eléctrica y Mecánica, Facultad de Minas, Universidad Nacional de Colombia, Medellín.

Recibido para revisar 23 de agosto de 2002, aceptado 22 de Enero de 2003, versión final 30 de Enero de 2003.

RESUMEN: El presente artículo propone un algoritmo de aprendizaje para que un mini-robot móvil pueda navegar en un ambiente desconocido. Se justifica su validez y se demuestra que se puede asimilar a una red neuronal asociativa. También se presenta una metodología formal para entrenar la red neuronal. Se descubre luego que la red planteada es topológicamente semejante a un arreglo lógico programable, que podría llegar a ser implementado sobre un microchip con tecnología F.P.G.A. Finalmente, por medio de dos fases propuestas para la navegación autónoma: entrenamiento y navegación, se logra que el minirobot se desplace en el entorno con un mínimo de colisiones.

PALABRAS CLAVES: Robótica Móvil, Navegación autónoma, Redes Neuronales Artificiales, Aprendizaje de Máquina.

ABSTRACT: A learning algorithm for the navigation of a minirobots in an unknown environment is proposed. The algorithm validity is justified and its implementation as a associative neural network is demonstrated. A formal methodology for training such a neuronal network is exposed. The topological similarity beetwen the network and a programmable logical array is described looking for its implementation on a microchip using F.P.G.A technology. Finally, by means of a two steps proposal: navigation and training, it is possible to attain that the mini-robot moves with minimum collisions.

KEYWORDS: Mobil Robots, Autonomous Navigation, Artificial Neural Networks, Machine Learning.

1. INTRODUCCIÓN

Un mini-robot móvil autónomo rodante, es un móvil con características cinemáticas limitadas, empleado para resolver tareas de navegación en ambientes desconocidos. La autonomía radica en varios aspectos: no requeriere supervisión de un operador para su desempeño y su fuente de energía es local. Se emplean actualmente en diversos campos y su aspecto más interesante radica en la forma como se comporta expresando cierto grado de inteligencia que le permite realizar tareas útiles para humanos, usando una estructura cibernética relativamente simple y económica.

Las situaciones socio-políticas actuales buscan en esta tecnología alternativas que permitan reducir costos causados por situaciones que puedan llegar a atentar contra el patrimonio y bienestar de los seres humanos, como es el caso de los actos de guerra y terrorismo. En manufactura asistida, son candidatos perfectos para transportar materia prima y herramientas, así como en sistemas de inspección y monitoreo. Estos se programan para que puedan navegar en un ambiente desconocido. La metodología para lograr esto es la incorporación de algoritmos formales o heurísticos.

Con miras a resolver varios problemas a los que se enfrenta una criatura cuando navega en un entorno real. En ciertas ocasiones esas tareas pueden ser llevadas acabo por agentes inteligentes en tiempo real.

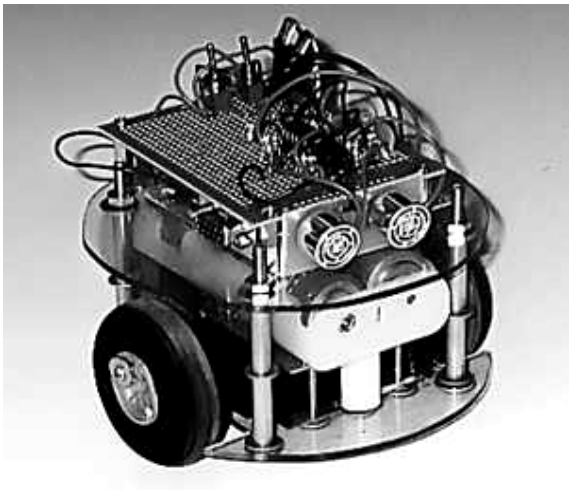

Figura 1. Mini-robot móvil de tracción diferencial

- Clasificación de patrones.
- Navegación y Planificación de la trayectoria.
- Sistema de evasión de obstáculos.
- Administración de la energía.
- Agentes asociados a la misión en particular.

Para lograr esto, los diseñadores colocan sobre la estructura mecánica sistemas computacionales, que ejecutan algoritmos de navegación autónoma con aprendizaje. Este artículo describe y justifica un método muy simple que puede llegar a ser útil en la implementación de un algoritmo anticolisión para un robot, basado en una arquitectura electrónica soportada sobre un microcontrolador de pequeña escala. La aplicación puede ser de interés en el diseño de robots móviles y juguetes interactivos.

2. EL PROBLEMA DE LA NAVEGACIÓN AUTÓNOMA

Los robots móviles autónomos son estructuras mecatrónicas montadas sobre una plataforma que permite que este pueda navegar por un ambiente de topografía muy específica. Los robots a los que nos referiremos en este articulo son robots simples rodantes de locomoción tipo diferencial o sincrónica. Para lograr que esta criatura navegue en un ambiente desconocido, debe ejecutar las tareas más intuitivas que una criatura que comparte el entorno con un humano debe resolver. Los humanos en sus complejísimos sistemas de navegación no tienen un mapa o sistema detallado, en términos de un sistema formal de coordenadas en el que navegan; simplemente bajo una observación del entorno, van seleccionando en línea la decisión que deben tomar para activar su mecanismo biomecánico, con el fin de aportar la maniobra que le permita continuar su desplazamiento normal al objetivo propuesto (Zalzala, 1996) .

Estos procesos mentales no siempre están fundamentados sobre la experiencia que se tenga del mismo. Por ejemplo, el acto de desplazarse por un corredor independiente de sus características generalizadas, se ejecuta casi siempre en términos de un desempeño seguro, dicho de otra forma, caminamos por el centro de la ruta salvo se imponga una restricción. Cuando tratamos de aprender a montar en bicicleta siempre preservamos nuestro bienestar, hecho que condiciona la forma como se lleva a cabo el aprendizaje, que en robótica móvil se conoce como *información a priori***.**

Se propone que durante la fase de aprendizaje y navegación del robot móvil, se incorporen dos agentes (Anzai, 1992): un *módulo de precisión* y un *módulo lógico***,** cuyas funciones son reducir el número colisiones o errores, para así acelerar el aprendizaje y dotarlo de cierta información previa.

Para apropiar estos trucos de navegación se entrenan durante una etapa de su vida útil, donde aprenden a habitar y compartir el entorno con humanos y otros objetos. En esa fase de formación, aprenden a reconocer situaciones y a aplicar estrategias que probadas, resultaron ser exitosas. De esta observación intuitiva se va a plantear un algoritmo muy simple de aprendizaje para permitir que un robot móvil rodante aprenda trucos muy sencillos.

Inicialmente considere una situación muy ideal donde un robot debe atravesar un corredor como muestra la Figura 3.C. Esta lo obliga a cambiar la trayectoria inicial de navegación a la entrada. Se puede observar que la criatura debe determinar si puede navegar o no (sensor de colisión) y comparar un patrón observado, con

algún patrón almacenado previamente en memoria y bajo un algoritmo de decisión, aplicar la estrategia que permita continuar su marcha normal.

Este algoritmo se divide en dos etapas: la primera fase es donde el robot aprende a identificar escenarios de su entorno y estrategias que debe aplicar a su estructura mecatrónica para no colisionar y crear una base de conocimiento o población de datos, y en la segunda etapa con el algoritmo programado sobre el microcontrolador, el robot mide el entorno y toma una decisión.

Es inadmisible pretender que un robot, así como un humano, aprenda todas las situaciones que se le presenten a su sistema de sensórica y anticolisión para poder navegar. Esto a nivel de memoria en el robot, recargaría el sistema con demasiada información que se puede considerar redundante. Para lograr un desempeño aceptable sin aprender todos los casos posibles de forma detallada, los humanos recuren a un proceso mental llamado *generalización*, que les permite resolver un caso particular cuando este se asemeja a un caso general que intuitivamente reúne muchos casos y se evalúa las propiedades del algún caso percibido en tiempo real y se compara con el caso generalizado y luego se suministra una estrategia de navegación.

Esta propiedad de las criaturas biológicas que resuelven este problema bajo el acto de generalizar, se ha definido como un *comportamiento inteligente***.** En los procesos inteligentes la criatura toma sus propias decisiones sin recurrir a una base de datos o a la autorización de un operador.

Cuando estos algoritmos simplistas se resuelven empleando máquinas de computo digital se llaman algoritmos de inteligencia artificial. Esta es la justificación del porque suele imponerse inteligencia artificial a un robot; la respuesta es que las técnicas de inteligencia artificial permiten a un sistema generalizar. Por ello los programadores y diseñadores de robots pueden echar mano de cualquier tipo de algoritmo de inteligencia artificial que se adapte mejor a las necesidades y a la arquitectura de su hardware. Continuando con el ejemplo del corredor, se van a analizar las percepciones simples y las estrategias intuitivamente posibles. El sistema posee un *agente autónomo* que le permita al robot medir la eficacia de la información, que logrará descubrir una estrategia de navegación efectiva y no comprometa la operación del robot.

Para resolver esto, se plantearán dos *agentes autónomos* en forma de módulos de programación; el primero sería un *módulo de precisión* y el segundo seria un *modulo de lógica*; Ambos operan juntos y se colaboran entre sí. La función del *módulo de precisión* es medir el entorno y retirarle la mayor cantidad de ruido que eventualmente pueda entorpecer el procesamiento. Algunos diseñadores emplean técnicas heurísticas o procedimientos más formales como cálculos auto-regresivos y estocásticos o filtros de Kalman.

Ahora, considerese un mini robot móvil simple de dos sensores ultrasónicos en su parte frontal y un par de motores montados en una topología de tracción tipo diferencial.

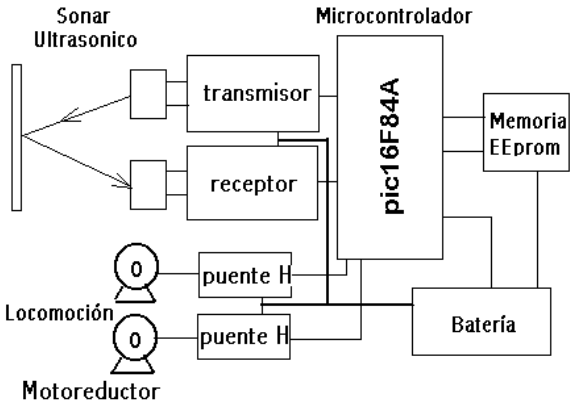

Figura 2. Arquitectura del robot propuesto.

La Figura 2 muestra la arquitectura simple de un robot con sistema anticolisión, basado en sonar ultrasónico, adicionalmente tiene una memoria serial tipo EEprom de característica no volátil, que sirve para almacenar patrones y pesos de la red neuronal y dos circuitos tipo puente H para accionar los dos motores de tracción através del microcontrolador PIC16F84A.

Se ha graficado el escenario que pretende atravesar. Simplificando aun más este ejemplo se considera que el sensor ultrasónico anticolisión es de dos estados; señal activa considerada como un uno lógico [1] se presenta en caso de posible colisión y señal lógica de cero[0] se da cuando no hay colisión posible en el momento de la

lectura. La situación se ha graficado en la Figura 3. En ella se puede observar el robot y el entorno así como una tabla de verdad de su comportamiento lógico como máquina de estado finito.

3. EVASIÓN DE OBSTÁCULOS EN UN ROBOT MÓVIL EMPLEANDO UNA RED NEURONAL ASOCIATIVA.

El ejemplo adaptado a una estructura del robot, que se prueba con una red neuronal, puede ser de utilidad para lograr que un robot móvil navegue evadiendo obstáculos en una situación muy simple. Suponga dos sensores anti-colisión con salida binaria (todo o nada) y una estructura mecánica de tracción diferencial que requieren dos señales binarias por motor; para girar en los dos sentidos y parar.

Además, se desea que el robot pueda evadir obstáculos cuando navega en un ambiente desconocido y tratando de mantener un rumbo hacia adelante, siempre que pueda. Intuitivamente se puede definir una serie de reglas simples que pueden evitar la colisión del robot y dependen del número de sensores sobre el robot. Con dos sensores a cada lado del robot, se pueden caracterizar cuatro patrones de entrada así:

- No colisiona.
- Se aproxima una colisión por el lado derecho.
- Se aproxima una colisión por el lado izquierdo.
- Se aproxima una colisión por ambos lados.

Para cada patrón de entrada se plantea una regla que relacione entradas y salidas al sistema:

1. Si no colisiona, entonces avance adelante.
2. Si se aproxima una colisión por el lado derecho, entonces gire por el lado izquierdo.
3. Si se aproxima una colisión por el lado izquierdo, entonces girar por el lado derecho.
4. Si se aproxima una colisión por ambos lados, entonces retroceda.

Estas cuatro reglas se representan como cuatro pares de patrones entrada y salida, que se almacenan sobre una red neuronal tipo perceptrón. Si se expresan como vectores binarios, tenemos el siguiente conjunto de pares entrada-salida que cumplen con las cuatro reglas simples anteriores:

X =[0 0]; Y =[1 0 1 0] no choca → adelante
X =[0 1]; Y =[0 1 1 0] choca derecha → giro izquierdo
X =[1 0]; Y =[1 0 0 1] choca izquierda → giro derecho
X =[1 1]; Y =[0 1 0 1] choca en dos lados → atrás

El primer termino de X, corresponde al estado del sensor izquierdo y el segundo, al estado del sensor derecho. Los dos primeros términos de Y corresponden al estado del motor izquierdo y los dos últimos, al estado del motor derecho.

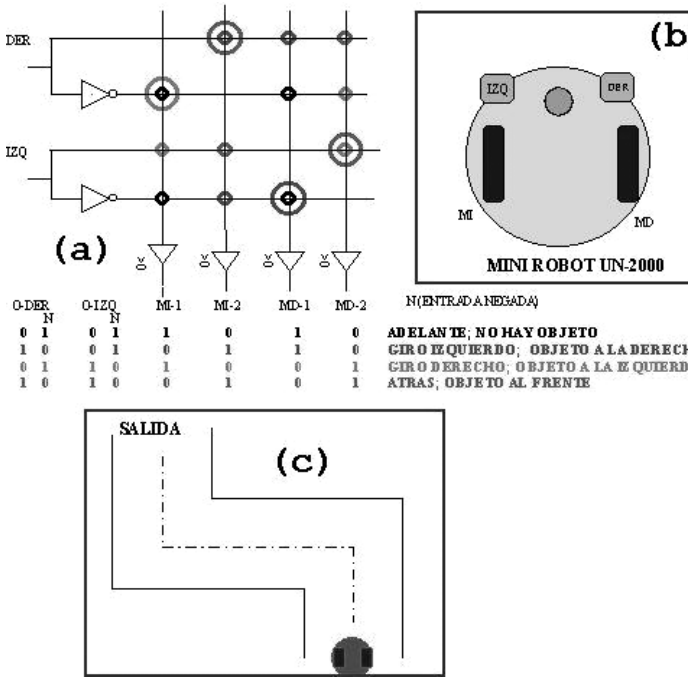

Figura 3. Red tipo perceptrón propuesta(a), Estructura del robot(b), Laberinto de prueba(c)

Este perceptrón se puede entrenar de forma grafica, marcando puntos donde la señal de entrada binaria aplicada por la columna debe activar alguna salida que se aplica por la fila en la parte inferior (par entrada-salida). En algunos casos se marca en el mismo punto y aparece concéntrica la marca. Esto significa refuerzo de conexión como se propone en (Delgado, 1998).

Con la información propuesta para entrenar este comportamiento, la teoría de las redes neuronales impide caracterizar los cuatro casos anteriores, ya que la clásica desigualdad siguiente no se cumple según (Ramírez, 2002):

$$PatronesMaximos < \left[\min\left(Dim\ X, Dim\ Y\right)\right]$$

Bajo este criterio sólo se puede almacenar con la información propuesta, un solo caso. Para superar este inconveniente se aumenta la dimensión del vector de entrada sin agregarle más sensores al robot; tomando simplemente la parte complementaria de la información de entrada en el vector Xi. Dicho de otra forma, por cada valor de entrada se adiciona un término complemento y se obtiene así un par de vectores de dimensión idéntica que al ser asociados sobre un perceptrón, producen una matriz cuadrada que asocia patrones normales y negados. Ahora se aplica el método descrito en (Delgado, 1998). Al terminar de asociar las cuatro reglas en cuestión, resulta la topología de la Figura 3.a.

Se investiga ahora por un método formal que sea útil para entrenar una red de dos capas con patrones de entrenamiento en procedimiento supervisado, esto con el fin de ser embebido en la memoria y sistema operativo implementado sobre un popular microcontrolador barato de ocho bits tipo PIC16F84A. El algoritmo buscado debe adaptarse a la siguiente topología:

- Dos capas.
- Pesos sobre una matriz cuadrada.
- Memorice patrones normales y negados.
- Bidirecional.

El procedimiento seleccionado apunta a una red neuronales asociativas tipo B.A.M. En este tipo de redes los elementos de ambas capas están interconectados entre si. Las unidades pueden o no tener conexiones de realimentación consigo mismas. Sin perder la generalidad de las redes neuronales, en estas arquitecturas tipo B.A.M. hay pesos asociados a las conexiones entre elementos del proceso. A diferencia de muchas otras arquitecturas, estos pesos se pueden determinar por anticipado si es posible identificar todos los vectores de entrenamiento.

Se puede tomar el modelo de un asociador lineal para construir la matriz de pesos. Dados L pares de vectores que constituyen el conjunto de ejemplares que se desean almacenar, se construye la matriz:

$$w = y_1 x_t^t + y_2 x_2^t + \ldots\ldots\ldots + y_L x_L^t$$

Esta ecuación da pesos de conexiones procedentes de la capa X y con destino la capa Y. Se transforma la B.A.M. en una memoria auto-asociativa, construyendo la matriz de pesos en la forma siguiente:

$$w = x_1 x_t^t + x_2 x_2^t + \ldots\ldots\ldots + x_l x_L^t$$

En este caso la matriz de pesos es cuadrada y simétrica. Una vez que se ha construido la matriz de pesos, la B.A.M. se puede emplear para recordar información. De hecho si la información tiene ruido o está incompleta la B.A.M. sería capaz de completar la información. Para recordar información empleando esta red, se llevan a cabo los pasos siguientes:

1. Se aplican el par de vectores Xo, Yo a los elementos del proceso de la B.A.M.

2. Se propaga la información de la capa X a la capa Y, y se actualizan los valores de las unidades de la capa.

3. Se vuelve a propagar esta información y se actualiza hasta la capa X, y se actualizan las unidades que se encuentren allí.

4. Se repiten los pasos 2,3 hasta que no hayan cambios en las unidades de ambas capas.

Este algoritmo da a la B.A.M. la naturaleza bidirecional.

La siguiente ecuación (*Freeman and Skapura*,1991), se usa para calcular una matriz que asocia entradas con salidas de acuerdo a la siguiente expresión:

$$M = \sum X_I^T * Y_I$$

siendo,
M, la matriz de asociación,
X, el vector fila de entradas,
Y, el vector columna de salida

Los elementos de la matriz representan los pesos de las conexiones de la red. Se realiza el cálculo de la matriz M con los valores de los cuatro patrones que se deben cumplir en las cuatro reglas de navegación anteriores. Se toman los valores normales y complementarios para construir un nuevo vector columna o vector ampliado y para el vector fila, se toman los valores originales propuestos para la salida deseada en la red. Los vectores de entrada siguientes se obtienen colocando en las filas 2 y 4 los términos negados de las filas 1 y 3. Luego efectuamos el cálculo de las submatrices *Mi*, para cada caso.

$$X_1 = \begin{bmatrix}0\\1\\0\\1\end{bmatrix};\quad Y_1 = \begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix};\ M_1 = \begin{bmatrix}0 & 0 & 0 & 0\\1 & 0 & 1 & 0\\0 & 0 & 0 & 0\\1 & 0 & 1 & 0\end{bmatrix}$$

Cuando no colisiona avanza adelante

$$X_2 = \begin{bmatrix}1\\0\\0\\1\end{bmatrix};\quad Y_2 = \begin{bmatrix}0 & 1 & 1 & 0\end{bmatrix};\ M_2 = \begin{bmatrix}0 & 1 & 1 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 1 & 1 & 0\end{bmatrix}$$

Cuando colisiona izquierda gira por derecha

$$X_3 = \begin{bmatrix}0\\1\\1\\0\end{bmatrix};\quad Y_3 = \begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix};\ M_3 = \begin{bmatrix}0 & 0 & 0 & 0\\1 & 0 & 0 & 1\\1 & 0 & 0 & 1\\0 & 0 & 0 & 0\end{bmatrix}$$

Cuando colisiona a ambos lados retrocede

$$X^4 = \begin{bmatrix}1\\0\\1\\0\end{bmatrix};\quad Y^4 = \begin{bmatrix}0 & 1 & 0 & 1\end{bmatrix};\ M^4 = \begin{bmatrix}0 & 1 & 0 & 1\\0 & 0 & 0 & 0\\0 & 1 & 0 & 1\\0 & 0 & 0 & 0\end{bmatrix}$$

y el resultado de sumar las cuatro matrices parciales es:

$$M = \begin{bmatrix}0 & 2 & 1 & 1\\2 & 0 & 1 & 1\\1 & 1 & 0 & 2\\1 & 1 & 2 & 0\end{bmatrix}$$

Esta matriz asocia los cuatro comportamientos simples. Dicho en otras palabras, representa el conocimiento o experiencia que permiten que el robot no colisione. Se hace la prueba de la matriz de asociación M con los cuatro patrones de entrada y se obtiene lo siguiente:

$$\begin{bmatrix}0 & 1 & 0 & 1\end{bmatrix} * \begin{bmatrix}0 & 2 & 1 & 1\\2 & 0 & 1 & 1\\1 & 1 & 0 & 2\\1 & 1 & 2 & 0\end{bmatrix} = \begin{bmatrix}3 & 1 & 3 & 1\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix} * \begin{bmatrix}0 & 2 & 1 & 1\\2 & 0 & 1 & 1\\1 & 1 & 0 & 2\\1 & 1 & 2 & 0\end{bmatrix} = \begin{bmatrix}1 & 3 & 3 & 1\end{bmatrix}$$

$$\begin{bmatrix}0 & 1 & 1 & 0\end{bmatrix} * \begin{bmatrix}0 & 2 & 1 & 1\\2 & 0 & 1 & 1\\1 & 1 & 0 & 2\\1 & 1 & 2 & 0\end{bmatrix} = \begin{bmatrix}3 & 1 & 1 & 3\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix} * \begin{bmatrix}0 & 2 & 1 & 1\\2 & 0 & 1 & 1\\1 & 1 & 0 & 2\\1 & 1 & 2 & 0\end{bmatrix} = \begin{bmatrix}1 & 3 & 1 & 3\end{bmatrix}$$

Usando un limitador duro para detectar números mayores que el peso de la conexión (dos) como función de activación, se cumple y se concluye que reconoce los cuatro patrones almacenados. Si el lector compara la topología de esta matriz con la topología de la red neuronal perceptrón de la Figura 3, aprecia una semejanza muy interesante**:** *Los elementos ceros indican no-conexión y el número sobre cada elemento de la matriz da el peso de la conexión en la red (grado de refuerzo).*

Con este razonamiento se ha probado que al tratar de obligar a la red binaria tipo perceptrón, que asocie patrones y sus complementos, se logra que se comporte como una red tipo B.A.M. Característica esta, típica de una red neuronal asociativa según (*Freeman and Skapura*, 1991). Además, al ser particularmente simétrica y cuadrada, consolidan la transformación a una red

tipo B.A.M. entrenada. De esta forma se indica, que si se entrena una B.A.M. con los patrones normales y complementarios en una estructura matricial cuadrada, se puede garantizar el aprendizaje sobre la red, si se dispone de un conjunto finito de patrones de entrenamiento deseables y conocidos. Así mismo, esta topología de red perceptrón con patrones normales y negados es idéntica a un arreglo lógico programable o F.P.G.A. Se concluyen que se puede llevar la red entrenada a una solución embebida (neuro-chip).

Hasta ahora, usando valores de dos estados para las señales de los sensores anticolisión del robot, se está en capacidad de entrenar una red asociativa muy simple. Si se adiciona un agente, que por medio de algún método, pueda extraer esta experiencia de la interacción del robot con el entorno sin colisionar y calcule la matriz, durante o al final del proceso de búsqueda de información (aprendizaje), se ha logrado producir un modesto comportamiento de navegación autónoma inteligente.

La pregunta que sigue es, *¿Cómo lograr que un robot móvil aprenda a evitar colisiones en un ambiente desconocido en tiempo real?***.** La respuesta: Requiere que se le proporcione de un agente que le permita seleccionar una población de parejas de patrones entrada a salida, que producen un desempeño con un mínimo de colisiones.

4. NAVEGACION EN AMBIENTES DESCONOCIDOS

En este modo, el robot no tiene una ruta definida, simplemente se desplaza por el recinto, tratando de no colisionar y mientras esto no suceda, su misión se limita a dirigirse hacia adelante siempre y cuando no se encuentre con un obstáculo. Se propone en este trabajo un algoritmo que se compone de dos etapas: primero el robot de forma autónoma extrae información del medio que le permitirá obtener un conjunto de estrategias útiles para poder cumplir la misión de navegación en un ambiente desconocido (entrenamiento). Luego se inicia la fase de aprendizaje donde se crea para el robot la red neuronal propuesta, que él empleará para que le ayude a navegar.

En esta última etapa (Anzai, 1992), el robot prueba soluciones hasta que le permitan un desempeño aceptable (pocas colisiones) definido por una *función objetivo*. Este algoritmo de entrenamiento para navegación en ambientes desconocidos está dotado de un *módulo lógico* y *un módulo de precisión sobre el microcontrolador*. El módulo lógico le permite al robot por medio de una simple *regla de inferencia heurística*, determinar si la solución propuesta tiene posibilidades de éxito. El módulo de precisión vigila que no choque innecesariamente para poder entrenar. El robot capta del medio un conjunto de patrones (entrada-salida) que aparte de estar limitados en cantidad por restricciones en el microcontrolador, deberán ser distintos entre sí. El entrenamiento concluye cuando se logren tomar del entorno la cantidad de patrones definidos en el programa para el robot.

De forma continua el robot vigila si se presenta un estado de colisión, en esta condición, debe detenerse, devolverse hacia atrás en línea recta, hasta que la colisión esté evacuada. Esta maniobra de reversa, garantiza que la nueva posición permitia aplicar una corrección. Una vez que esto suceda, se detiene y mide las distancias a su alrededor. Luego la almacena dentro de la memoria interna del microprocesador. Lo que sigue es analizar esa información para poder crear una estrategia que permitan continuar hacia adelante sin colisionar. Esta tarea se conoce como la *Integración sensorial* (*Dudek*, 2001)**.** La metodología empleada para tomar la población de muestras que debe entrenar el sistema, se basa en generar movimientos de giro derecho e izquierdo para obtener una posición del robot donde se pueda detectar un patrón de señales de distancia como se muestra en la Figura 4.

Se asemeja a un corredor imaginario o un lóbulo que se proyecta al frente del robot y vigilando que las lecturas de los sonares cumplan con esta distribución. Así el robot se puede formar una imagen rudimentaria y abstracta del recinto para poder pasar por él, sea este amplio o estrecho, sin colisionar. Estas operaciones que orientan la toma de datos constituyen lo que se denominó un *módulo lógico* para el

entrenamiento. Las operaciones que siguen son ejecutadas por medio del módulo de precisión adoptado para el robot.

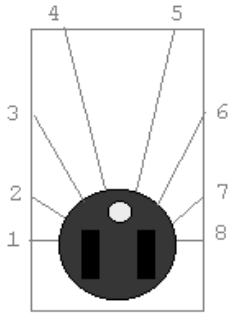

Figura 4. Patrón de ultrasonidos entrenando.

Continuando con la explicación del algoritmo, se puede decir, que luego de decidir si el patrón nuevo cumple con las restricciones impuestas en el módulo anterior, el robot deberá contabilizar la cantidad de giros y su sentido, que son necesarios para obtener esta nueva posisión de buenas posibilidades. Esta condición cumplida puede desencadenar un proceso que decide si la información se almacena o no. Lo que sigue ahora será analizar los registros de memoria del robot para determinar si es experiencia nueva o vieja, operación muy útil para evitar que el robot se llene inútilmente de reglas redundantes, que hacen un mal uso de la memoria. Además el objetivo de la inteligencia artificial es tratar de generalizar.

Otra pregunta que el programa debe hacerle al sistema, es sobre la cantidad de reglas diferentes que la red propuesta puede almacenar, esto además ayuda a limitar el tiempo de entrenamiento. Cumplida la cantidad establecida, el programa para. Está ahora preparado para iniciar su fase de navegación en ambientes desconocidos, que consiste en leer en línea los sensores, calcular la salidas multiplicando el vector de entrada (ampliado) por la matriz de asociación Mi y calcular el accionamiento en el motor del robot.

5. CONCLUSIONES

La inteligencia propuesta no es la más eficiente pero permite soluciones rápidas en aplicaciones simples. De una forma ingeniosa se convierte un perceptrón en una red auto asociativa. Este desarrollo significa un aporte interesante, ya que facilita la implementación de la red. Para facilitar aun más la implementación, se emplearon señales discretas en el sensado de obstáculos para el robot.

Es de primordial importancia que la máquina generalice una vez adquiera cierto grado de aprendizaje. Esto se logra mediante técnicas de inteligencia artificial o aprendizaje basado en reglas. Las reglas pueden crecer mucho, si no se logra incorporar un agente autónomo que permita generalizarlas.

REFERENCIAS

Anzai, Y. *Pattern and Machine Learning*. Academic Press Inc., Boston , 1992.

Delgado, A. *Inteligencia Artificial y Mini robots.* Ecoe Ediciones, Santa Fe de Bogotá, 1998.

Dudek, G and jenkin, M. *Computational Principles of Mobile Robotics*. Cambridge University Press, New York, 2001.

Freeman, J and Skapura, D. *Redes Neuronales, Algoritmos, Aplicaciones y Técnicas de Programación*. Addison-Wesley, Madrid, 1991.

Ramirez, G. Mini-Robots Inteligentes, [Tesis de Maestría], Santa Fe de Bogota: Universidad Nacional de Colombia, 2002.

Zalzala A. and Morris A, *Neural Networks for Robotic Control*. Ellis Horwood, London 1996.