

**DESIGN AND IMPLEMENTATION OF AN AUTONOMOUS VEHICLE
BARRIERS EVASER USING NEURAL NETWORKS IN A DIGITAL SIGNAL
PROCESSOR DSP56F801**

**DISEÑO E IMPLEMENTACIÓN DE UN VEHÍCULO AUTÓNOMO EVASOR DE
OBSTÁCULOS EMPLEANDO REDES NEURONALES EN UN PROCESADOR
DIGITAL DE SEÑALES DSP56F801**

PhD. Oscar Eduardo Gualdrón Guerrero, Ing. Carlos Alberto Gamarra Sierra

Universidad de Pamplona

Ciudadela Universitaria. Pamplona, Norte de Santander, Colombia.

Tel.: 57-7-5685303, Fax: 57-7-5685303 Ext. 164

E-mail: oscar.gualdron@unipamplona.edu.co; carlos_gamarra@hotmail.com

Abstract: This paper deals with the design and implementation of an autonomous vehicle capable of avoiding obstacles in real time controlled by a neural network, we will use sensors to learn about their environment and take action to avoid collision, the control was implemented in a Digital Signal Processor (DSP56F801) from Motorola.

Resumen: Este trabajo trata sobre el diseño y la aplicación de un vehículo autónomo capaz de evitar obstáculos en tiempo real controlado por una red neuronal artificial, se utilizaron sensores para aprender sobre su entorno y tomar medidas para evitar la colisión, el control fue implementado en un Procesador de Digital de Señales el DSP56F801 de Motorola.

Keywords: DSP, Microbot, Artificial Neural Networks.

1. INTRODUCCION

Los microbots han tenido un auge en los últimos años por la reducción de costos de sus componentes y la facilidad de adquirirlos, hoy en día son utilizados por las industrias, las universidades y cualquier persona con un interés en ellos.

La ciencia aplicada que surge de la combinación de la tecnología de maquinas-herramientas y la informática o bien llamada robótica, amplía su rendimiento integrándola a una forma de computación nueva como son las redes neuronales, que son redes interconectadas masivamente en paralelo con elementos simples, las cuales intentan interactuar con los objetos del mundo real

del mismo modo que lo hace el sistema nervioso biológico.

Al ser implantadas las redes neuronales artificiales en un procesador digital de señales para aplicaciones de microbótica, brinda la oportunidad de un procesamiento rápido con una precisión elevada, tolerancia a fallos con un bajo costo.

Para la elaboración de un microbot evasor de obstáculos se parte de la adquisición de datos del entorno por medio de sensores para realizar operaciones propias de la red neuronal en el DSP, obteniendo una respuesta que evita la colisión.

2. DISEÑO DEL MICROBOT

El Robot móvil, es capaz de detectar la presencia de obstáculos a su alrededor y evaluar por medio de redes neuronales la forma de evitarlos.

Para la elaboración de este microbot se tuvo en cuenta la funcionalidad que este debía poseer, una estructura fuerte, debió ser económico, fácil de modificar.

Después de analizar las diferentes Estructuras se escogió por su tamaño reducido y simplicidad la estructura mostrada en la figura 1, otro factor que se tuvo en cuenta fue la experiencia al trabajar previamente con esta clase de estructura mecánica.

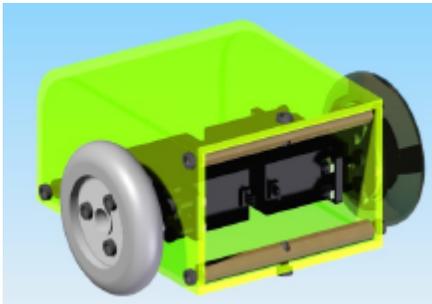


Fig. 1. Vista microbot

2.1. Sistema de tracción

La tracción diferencial es la más utilizada por su simplicidad, permitiendo realizar giros y avanzar solo controlando el sentido de las ruedas.

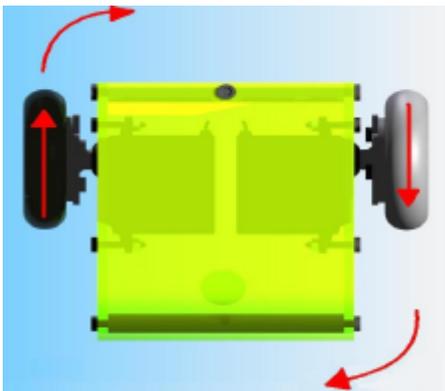


Fig. 2. Ejemplo tracción diferencial. Propiedad autor de este trabajo

Servomotor

Los servomotores al ser trucados para que giren 360 grados con el circuito de control interno son una opción eficiente para el funcionamiento del vehículo objeto de este trabajo, ahorrando el circuito de cambio de giro.



Fig. 3. Servomotor Futaba s3003

2.2. Acople del DSP56F801 al servo motor

Dado que el nivel de tensión máxima que emplea el DSP56F801 es de 3,3V y no está en el rango de operación de los servomotores la cual es de 5V, se hace necesario realizar un acondicionamiento de esta señal para garantizar niveles de tensión compatible.

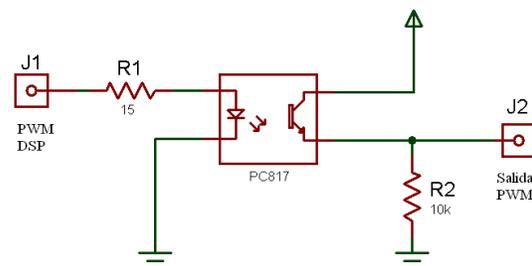


Fig. 4. Esquema circuito acople DSP56F801 al Servomotor

Para el cálculo de la resistencia R1, se aplica la siguiente fórmula, donde V_{DSP} es el voltaje máximo del DSP, V_D , Voltaje de trabajo del diodo led interno del opto acoplador; y I_D la corriente máxima de trabajo del diodo del opto acoplador.

$$R_1 = \frac{V_{DSP} - V_D}{I_D} \quad (1)$$

$$R_1 = \frac{3.3V - 3V}{20mA} = 15 \Omega \quad (2)$$

2.3. Fuente de potencia

La fuente de poder será la encargada de suministrar la alimentación al robot móvil para realizar la función de diseño, hay que tener en cuenta el consumo del vehículo para garantizar que en un futuro la energía que tenemos sea suficiente.

Se calcula una corriente de 401 miliamperios. Sin considerar que los motores al estar sometidos a una carga requieren mayor energía, para esto se le agrega un 25% más a la corriente calculada. Obteniendo un total de 501,25 miliamperios como máximo.

Acondicionamiento de la fuente de poder

El vehículo requiere una alimentación de 5 Voltios, tanto para los servomotores como para los sensores, para ello se conecto dos baterías de 3.6 V en serie.

Para garantizar la estabilidad y nivel de voltaje del suministro de energía se empleo un regulador LM7805, conectándolo según las recomendaciones del fabricante.

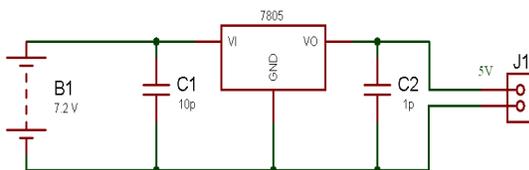


Fig. 5. Circuito acondicionador de tensión

Este circuito regulará el voltaje de entrada, garantizando una tensión de 5V aproximadamente, con una corriente máxima de 1 A. los condensadores son utilizados como eliminadores de señales parasitas.

2.4. Sensor de proximidad

El vehículo al ser un evasor de obstáculos debe conocer su entorno, para ello se seleccionó el sensor de proximidad de la familia Sharp IS471F por su inmunidad a la luz visible, facilidad de calibración, tamaño y su grado de sensibilidad.

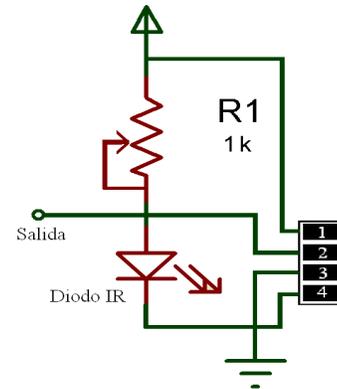


Fig. 6. Esquema muestra el funcionamiento del sensor

2.5. Control del vehículo

El dispositivo encargado del procesamiento de las señales, donde se implementará la red neuronal es el DSP56F801 de Motorola, un poderoso procesador digital de señales con una frecuencia de núcleo de 80 megahertz.

2.6 Estado del vehículo

Con esto la estructura física del vehículo se encuentra terminada, solo resta la implementación de software para poder así realizar las diferentes calibraciones y pruebas del funcionamiento.



Fig. 7. Vehículo Neurobot

3. DISEÑO DEL SOFTWARE DE CONTROL

El sistema de control es el encargado del procesamiento de las señales que afecte al microbot neurobot para tomar las acciones correspondientes. Para esta aplicación se emplea una red neuronal por su poca utilización en robótica móvil.

3.1. Selección de la red neuronal

La red neuronal de Retro-propagación es la adecuada para llevar a cabo el proceso de control del microbot, debido a su relativa facilidad de implantación, y a sus características como clasificador.

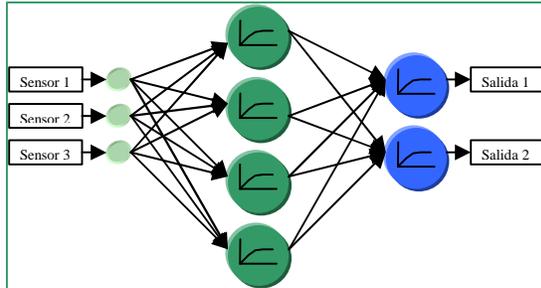


Fig. 8. Estructura red neuronal

3.2. Entrenamiento de la red neuronal

El entrenamiento de la red neuronal es realizado por Matlab®. El cual arroja los pesos sinápticos necesarios para la integración de está al DSP56F801.

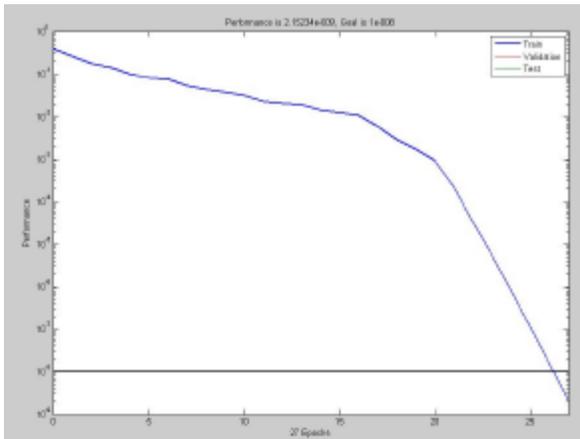


Fig. 9 Red neuronal con 4 neuronas

Se aprecia en la fig. 9 que la red neuronal se entrenó satisfactoriamente con un error significativamente bajo.

3.3. Integración al DSP56F801

Para la integración con el DSP56F801 se emplean:

- Los pesos sinápticos de la capa de entrada a la capa oculta.
- Los pesos de la capa oculta a la capa de salida.

- Los *Bias*, que son los pesos extras.
- Así como conocer qué tipo de función de activación fue empleada. En la capa oculta *Tansig* y en de salidas *logsig*.

Para obtener los pesos de la capa de entrada a la capa oculta se utiliza el comando “red.IW”, para los pesos de la capa oculta a la capa de salida “Red.LW”, y para los *bias* se obtiene con “Red.b”. Los pesos y las *bias* son guardados en la memoria del DSP56F801, para ser aplicadas ha las formulas de verificación de la red MLP las cuales son las siguientes:

$$n^o_j = \sum_{i=1}^q W^o_{ji} P_i + b^o_j \quad (3)$$

W^o_{ji} : Peso que une la capa de entrada con la capa oculta

P_i : Componente del vector de entrada.

b^o_j : Ganancia de la capa oculta, pesos extra de la red.

Esta fórmula multiplica cada entrada con su respectivo peso y luego le suma la bias correspondiente a cada neurona.

Cada una de las neuronas de la capa oculta posee una salida dada por:

$$a_i = \text{tagsig}\left(\sum_j w_{ji} P_i - b^o_j\right) = \frac{e^{n^o_j} + e^{-n^o_j}}{e^{n^o_j} - e^{-n^o_j}} \quad (4)$$

Donde *tansig* es la función de activación de la capa oculta.

La salida de cada neurona es la entrada de los pesos de la capa oculta a la capa de salida:

$$s_i = \left(\sum_j w_{ij} a_j + b^1_i\right) \quad (5)$$

w_{ij} : son los pesos que conectan la capa oculta con la capa de salida.

b^1_i : son los pesos complementarios de la capa de salida.

La salida total de la red es n_k :

$$n_k = \log \text{sig}\left(\sum_j s_j w_{kj} + b^o_j\right) = \frac{1}{1 + e^{-s_j}} \quad (6)$$

Los pesos que conectan a la capa de entrada con la capa oculta fueron guardados en una matriz $W1$, con cuatro filas y cinco columnas; Las variables de

entradas son adquiridas y guardadas en el vector P; los pesos que conectan la capa oculta con la capa de salida son guardados en la matriz W2 de tres filas y cuatro columnas; la *bias* que corresponde a la capa oculta fue guardada en un vector de cinco posiciones b1 y por último la *bias* de la capa de salida en un vector de tres posiciones b2.

La salida de esta red decide la acción a seguir para que pueda evitar los obstáculos que se presenten. Utilizando los servomotores como actuadores que posicionaran al vehículo en una posición segura.

Gracias a la exactitud del DSP56F801 al trabajar con puntos flotantes la salida de la red es equivalente a la procesada por MatLab. Siendo un éxito la integración de la red neuronal al DSP56F801.

4. CONCLUSIONES

Las limitaciones de memoria del DSP56F801 sólo permitieron una red neuronal de cuatro neuronas impidieron realizar aplicaciones extras más complejas. La red neuronal es capaz de dar una respuesta a casos que no se tuvieron en cuenta en el entrenamiento, siendo un sistema de control confiable y tolerante a fallos.

La función de un sistema puede ser cambiada con sólo la sustitución de los pesos de la red neuronal, mejorando su flexibilidad.

El control del vehículo fue sencillo debido al sistema de tracción diferencial que se empleó, solo fue necesario cambiar el sentido de giro de las ruedas para girar o avanzar.

Los sensores que se emplearon satisficieron las necesidades de este trabajo de grado. Por su precisión, facilidad de calibración, y tamaño, logrando un diseño confiable para la implementación y manejo de las redes neuronales de retropropagación.

El empleo de redes neuronales resulta atractivo para el mejoramiento de sistemas que empleen otras lógicas de control, mejorando considerablemente su desempeño. Teniendo en cuenta el nivel de hardware que se utilizó para lograr una aproximación considerable. Para determinar el número de neuronas a utilizar se utilizó el criterio del autor, ya que no existe ninguna regla que lo determine.

5. RECOMENDACIONES

Se sugiere para investigaciones futuras realizar un entrenamiento online de la red neuronal. Para aplicaciones más avanzadas se recomienda quitar la carga computacional del DSP56F801 y entregársela a un procesador con mayores recursos.

Se recomienda el uso de técnicas de odometría, junto con sensores como giroscopios, brújula electrónica, encoder y otros que permitan garantizar la posición del microbot en el espacio.

Se recomienda el uso de más sensores de proximidad para tener un mejor conocimiento del entorno siempre y cuando se libere al microbot de la carga computacional de la red neuronal.

REFERENCIAS

- Kartalopoulos, S.V. Biological neural networks. In Understanding Neural Networks an Fuzzy Logic Basic Concepts and Applications, The Institute of Electrical and Electronic Engineers, Inc., New York, 1996.
- Vivas, R. L. Pasantía: Estudio de métodos de inteligencia artificial y desarrollo de códigos y aplicativos en Matlab para el modelamiento de sistemas basados en árboles de decisión, lógica difusa y redes neuronales. Universidad Francisco de Paula Santander. Cúcuta, Colombia, 2005. P.25
- Patiño, H.D. Apuntes Curso Introducción a las redes Neuronales Artificiales en Ingeniería (Rama Estudiantil IEEE de la UNSJ), San Juan, Argentina, 2000.
- Guzzo, M.A. Garguilo, E. Patiño, H.D. Diseño y desarrollo de un Neuro-chip basado en FPGAs, Universidad Nacional de San Juan, San Juan, Argentina. P-24
- Bond and L. Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, Readings in Distributed Artificial Intelligence, pages 3–36. Morgan Kaufmann Publishers: San Mateo, CA, 1988.
- Asada M., Stone P, Kitano H., Drogoul A., The RoboCup Physical Agent Challenge: Goals and Protocols for Phase I. Kitano Hiroaki (Eds). (1998). RoboCup .97: Robot Soccer World Cup I. Lectures Notes in Computer Science; Vol 1935: Lectures Notes in Artificial Intelligence. Berlin: Springer-Verlag.
- Moctezuma Eugenio, Huitzil, Cesar. Estudio sobre la implementación de redes neuronales

- artificiales usando Xilinx System Generator.
Puebla, México. P.I. Benemérita
Universidad Autónoma de Puebla.
- Gualdrón G., O. *Redes neuronales, Apuntes de Clases*, Universidad de Pamplona, Colombia, 2008.
- Gutiérrez, J. *Introducción a las redes neuronales*, Universidad de Cantabria, 2000.
- Larrañaga, P.; Iñaki, I. y Abdelmalik, M. *Redes neuronales*, Universidad Carlos III, Madrid, España, 2007.
- Matich, D. *Redes Neuronales: Conceptos Básicos y aplicaciones*, 2001.

INTERNET

- <http://www.microrrobotica.es>, (Biblia CT6811)
- <http://www.eupmt.es/cra/>
- <http://www.jeuazarru.com/html/microbots.html>
- <http://es.wikipedia.org/wiki/Servomotor>
- <http://www.alldatasheet.com>. (DSP56F801)