

# Diseño e implementación de prototipo de Robot Explorador Inteligente con transmisión de datos y estación receptora

Mario Rodolfo Hernández Valladares, estudiante de Ingeniería Mecatrónica  
Universidad Ricardo Palma, Perú, [mhernandezv2012@gmail.com](mailto:mhernandezv2012@gmail.com)

*Abstract—The purpose of this project was to develop an algorithm for an intelligent explorer robot, based on Artificial Neural Networks using Matlab software and an Arduino module. In addition, the objective of this robot was to collect environmental data through the DHT11 sensor and ultrasonic sensors; thus, this data is sent while the robot moves. Then, the information will be transmitted through the wireless transmission module NRF24L01 to the receiving station, which will be connected to the Matlab software through a computer where the data will be processed, while the outputs obtained will be retransmitted to the robot to move it. The prototype had an efficiency of 88.00% in the obstacle avoidance, with a useful range of 6 meters of transmission in enclosed spaces. Also, the robot is powered by 3 Li-ion batteries, and the receiving station is powered by a USB cable connected to the computer.*

*Keywords—Explorer Robot, artificial neural network, intelligent robot, wireless transmission, Matlab.*

## I. INTRODUCCIÓN

Después de las catástrofes naturales, la prioridad de los equipos de rescate es buscar y salvar a la mayor cantidad de personas, exponiendo sus propias vidas al ingresar a lugares que presentan grandes concentraciones de gases peligrosos, y espacios reducidos sensibles a derrumbes, todo con la finalidad de rescatar vidas humanas. Por ello, tras el terremoto de Kobe (Japón) en 1995 que tuvo como consecuencia el fallecimiento de más de 6000 personas, y con el creciente avance en la robótica, científicos de todo el mundo, como Robin Murphy et al. [1], centraron sus investigaciones en el desarrollo de robots de rescate para ayudar a salvar a la mayor cantidad de víctimas producto de estos fenómenos naturales.

De esta manera, como primer paso, en este trabajo se ha propuesto el desarrollo de un algoritmo basado en inteligencia artificial, con la intención de tomar decisiones propias a partir de la generalización de experiencias anteriores. Particularmente, se ha optado por las Redes Neuronales Artificiales para la evasión de objetos hallados en el camino, en cuanto se realiza la tarea de transmisión de datos recolectados a una estación receptora, como producto de la tarea de censado. Como referencias al desarrollo de este prototipo se cita la investigación realizada por Pandey, A. y Parhi, D. [2], quienes diseñaron un modelo de robot móvil autónomo que buscaba recorrer la menor distancia de punto a punto esquivando objetos. Este diseño está basado en Sistemas Adaptativos de Inferencia Neuro-Difusa (ANFIS) simulado en Matlab, y posteriormente implementado en Arduino UNO. Ellos obtuvieron que la simulación se desplazó, en promedio,

107.2 cm. para llegar a la meta, mientras que el robot real lo hizo en 115.3 cm. en promedio, por lo que concluyeron que la red ANFIS condujo eficientemente el robot de forma segura en los ambientes desordenados propuestos.

Así como también, la investigación realizada en [3], señala el uso de una red neuronal artificial del tipo propagación hacia atrás con 4 capas ocultas: 12, 10, 8 y 2 neuronas respectivamente, y una función de activación tipo TANSIG. Asimismo, implementaron una interfaz gráfica para generar una base de datos, y sirva para el entrenamiento de la red neuronal que permita al robot desplazarse de un punto inicial a un punto final. Además, tal trayectoria se basó en los siguientes parámetros: coordenadas iniciales y finales, número de trayectorias a generar y ángulo de corrección inicial. De esta manera, se demuestra la potencialidad de las redes neuronales en la solución de problemas de robótica móvil, comprobando la efectividad de la misma en la clasificación de superficies al obtener una eficiencia del 97%.

Y, asimismo, la investigación desarrollada en [4], señala que implementaron un robot que busca la construcción de mapas basado en lógica difusa, estimando la posición del mismo con un filtro de Kalman Extendido. Para ello, utilizaron sensores infrarrojos y ultrasónicos. Estos sensores a su vez, se fusionaron y se enviaron a una red neuronal Backpropagation, la cual se encargó de proporcionar una estimación de la medida de las distancias con el fin de ser usadas en las tareas de navegación. De esta forma, al fusionar los datos provenientes de ambos sensores se reduce la incertidumbre en la medida, con la cual consiguieron construir mapas más precisos que empleando los sensores independientemente.

De la misma manera, en [5] se hace mención del uso de un controlador neuronal basado en una red multicapa, con algoritmo de aprendizaje Backpropagation, pero con un número de neuronas diferentes por cada capa. Además, para cada comportamiento autónomo de navegación, se diseñó y entrenó una red neuronal en particular; y, se tuvo en cuenta los vectores de entrada y de salida deseada. No obstante, los cuatro controladores neuronales coincidieron con el mismo número de capas. Y, según [6] opta por utilizar una red neuronal difusa para llevar a cabo las tareas de control.

Por lo tanto, el propósito de este proyecto fue desarrollar un algoritmo de desplazamiento para un robot explorador inteligente, basado en Redes Neuronales utilizando el software Matlab y un módulo Arduino. Este robot a su vez sortea obstáculos y recolecta datos del entorno, mientras se desplaza.

Para ello, se utiliza el sensor de temperatura y humedad DHT11, y sensores ultrasónicos. Y, una vez hecha la recolección, tal información se transmite mediante el módulo de transmisión NRF24L01 hacia una estación receptora. Una vez recibido los datos de los sensores, se envían a través del puerto serial hacia una computadora, para ser procesados en el software Matlab y con la finalidad que se ejecute la Red Neuronal, y luego se muestren los datos de temperatura y humedad en un gráfico. Asimismo, estos servirán de apoyo al equipo de rescate para la toma de precauciones durante el evento. Por último, los datos de salida obtenidos por la Red Neuronal serán enviados de regreso al robot explorador para que pueda continuar con el desplazamiento.

## II. MARCO TEÓRICO

En esta sección se hace una descripción breve de los principales elementos utilizados en el desarrollo de este proyecto.

### A. Redes neuronales artificiales

Las redes neuronales artificiales no se aproximan a la complejidad del cerebro; sin embargo, existen dos similitudes clave entre las redes neuronales biológicas y artificiales. Primero, los componentes básicos de ambas redes son dispositivos computacionales simples (aunque las neuronas artificiales son mucho más simples que las neuronas biológicas) que están altamente interconectadas. En segundo lugar, las conexiones entre neuronas determinan la función de la red [7].

Además, el aprendizaje supervisado en las redes neuronales, implica la modificación de sus pesos sinápticos mediante la aplicación de un conjunto de ejemplos de capacitación etiquetados o ejemplos de tareas. Cada ejemplo consta de una señal de entrada única y la correspondiente respuesta deseada (objetivo), y el entrenamiento se repite para muchos ejemplos en el conjunto, hasta que la red alcanza un estado estable en el que no hay más cambios significativos en los pesos sinápticos [8].

### B. Módulo Arduino Mega

Para controlar al robot explorador inteligente se empleó un Arduino Mega 2560 R3, que es una tarjeta de desarrollo que utiliza el microcontrolador ATmega2560 [9].

### C. Módulo Driver Puente HL298N

Fue utilizado para el control del sentido de giro de los motores, y de la velocidad a través de señales TTL obtenidas de los microcontroladores. Además, también admite entradas de señal PWM para el control de la velocidad [10]. Asimismo, tiene integrado un regulador de voltaje de 5V encargado de alimentar la parte lógica del módulo L298N; por ello, el uso de este regulador se hace a través de un Jumper y se utiliza para alimentar la etapa de control [10]. A continuación, la figura 1 muestra una representación del módulo driver Puente utilizado para el control de giro del motor.

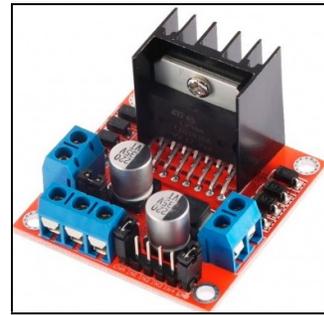


Fig. 1. Módulo Driver Puente H L298N [10].

### D. Módulo Transmisor/Receptor NRF24L01

Para la transmisión y recepción de datos se utilizaron dos módulos NRF24L01. Este módulo es un transceptor RF (transmisor + receptor) que trabaja a una frecuencia comprendida entre 2.4 a 2.53 GHz, y donde es posible elegir entre 125 canales espaciados a razón de 1 MHz. Asimismo, la velocidad de transmisión es configurable entre 250 Kbps, 1Mbps, y 2 Mbps. Además, permite la conexión simultánea hasta con 6 dispositivos [11]. A continuación, la figura 2 muestra una representación del módulo en mención.

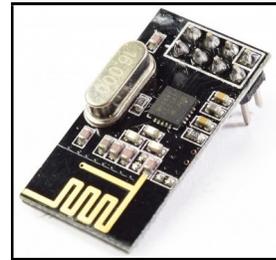


Fig. 2. Módulo Transmisor/Receptor NRF24L01 [11].

Por otro lado, el control del módulo NRF24L01 se realiza a través de un bus SPI, a través de dos versiones. El primero, utilizando una antena integrada en forma de zig-zag para un alcance máximo de 20-30 metros, y la segunda versión de alta potencia que incorpora un amplificador y una antena externa, con un alcance máximo hasta de 700-1000 metros [11].

### E. Módulo Sensor de Temperatura y Humedad

Corresponde al módulo DHT11, que es un sensor digital de temperatura y humedad relativa. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos [12]. A continuación, la figura 3 muestra una fotografía del módulo sensor de temperatura y humedad utilizado en este proyecto.

### F. Módulo de distancia HC-SR04

Para obtener los datos de entrada a la red neuronal se utilizaron 3 módulos HC-SR04, que corresponden a sensores de distancia utilizando el ultrasonido para determinar el alcance de un objeto en un rango de 2 a 450 cm. Estos módulos poseen dos transductores: un emisor y un receptor piezoeléctricos [13]. Ver la figura 4.

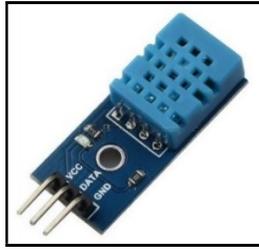


Fig. 3. Módulo Sensor de Temperatura y Humedad DHT11 [12].



Fig. 4. Módulo de sensor de distancia HC-SR04 [13].

De esta manera, para obtener la distancia a la que se encuentra el obstáculo se emplearon las siguientes fórmulas. Entonces, si se tiene (1).

$$Velocidad = \frac{\text{distancia recorrida}}{\text{tiempo}} \quad (1)$$

Y, la velocidad es igual a 340m/s, que es la velocidad del sonido; y, se conoce que el tiempo se encuentra en la unidad de microsegundos, la distancia en centímetros se hallará con (2).

$$\frac{340m}{s} \times \frac{1s}{1000000us} \times \frac{100cm}{1m} = \frac{2d}{t} \quad (2)$$

Despejando se obtiene la expresión matemática (3), que es empleada en el código de programación del robot explorador inteligente.

$$d(cm)_{sensor} = 0.017 \times t(us) \cong \frac{t(us)}{58} \quad (3)$$

#### G. Motorreductores para el desplazamiento

También se utilizaron 4 motorreductores, que son motores DC con una caja reductora integrada, lo cual permite entregar un adecuado torque en un pequeño tamaño y bajo voltaje. La carcasa del motor es de plástico resistente, no tóxico y de color amarillo. La llanta de goma posee un excelente agarre en superficies de cemento, madera y acrílico. Incluso permite soportar cierto nivel de rebote [14]. A continuación, la figura 5 muestra una representación del motorreductor utilizado en este proyecto.

#### H. Fuente de alimentación portátil

Como fuente de alimentación portátil, se utilizaron 3 baterías Li-ion 18650, los cuales son dispositivos diseñados para almacenamiento de energía eléctrica que emplean como electrolito una sal de litio, que consigue los iones necesarios para la reacción electroquímica reversible que tiene lugar entre el cátodo y el ánodo [15].



Fig. 5. Motorreductor con llanta [14].

#### I. Módulo Display LCD 1602

Para mostrar los datos se utilizó un display LCD 1602, que posee 2 filas y 16 columnas de dígitos alfanuméricos, funciona con el controlador interno HD44780 [16].

#### J. Módulo Adaptador LCD a I2C

Se utilizó el módulo adaptador LCD a I2C para manejar el display LCD con 2 pines (SDA y SCL). El Módulo está basado en el controlador I2C PCF8574 que es un expansor de Entradas y Salidas digitales controlado por I2C. Por el diseño del PCB, este módulo se utilizó especialmente para controlar un LCD alfanumérico. La dirección I2C por defecto del módulo puede ser 0x3F o en otros casos 0x27 [17]. A continuación, la figura 6 muestra una representación del módulo adaptador LCD a I2C utilizado en el proyecto.

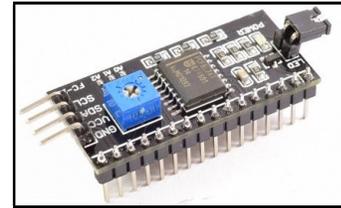


Fig. 6. Módulo adaptador LCD a I2C [17].

### III. IMPLEMENTACIÓN DEL ROBOT EXPLORADOR INTELIGENTE

Para la implementación real del robot explorador inteligente se desarrollaron tres etapas: diseño del circuito electrónico, diseño del prototipo mecánico y diseño del algoritmo basado en una red neuronal.

#### A. Diseño del circuito electrónico

El circuito electrónico del robot explorador inteligente está compuesto por un Arduino Mega, un transceptor NRF24L01, un sensor de temperatura/humedad DHT11, un driver puente H L298N, tres sensores de ultrasonido HC-SR04 y cuatro motorreductores para efectuar el desplazamiento. Por otro lado, el circuito electrónico de la estación receptora está compuesto por un Arduino Mega, un transceptor NRF24L01 y un display LCD.

Asimismo, cada uno de los elementos anteriormente citados fue seleccionado tomando en cuenta sus respectivos datasheet. Así como también, fueron probados antes de ser implementados en el robot explorador y la correspondiente estación receptora. A continuación, la figura 7 muestra una fotografía del montaje de la estación receptora en un

Protoboard, acompañada del módulo LCD para la visualización de la cantidad de veces que los datos fueron recibidos y enviados exitosamente.

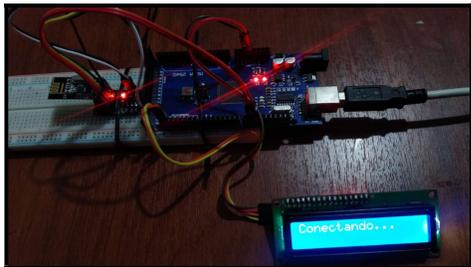


Fig. 7. Fotografía del montaje de la estación receptora y el módulo LCD.

### B. Diseño del prototipo mecánico

El diseño del prototipo mecánico se realizó sobre un chasis genérico de 4 llantas. A dicho chasis, se le agregaron 3 baterías de Li-ion 18650 como fuente de alimentación, y un interruptor para controlar el encendido y apagado del mismo. A continuación, la figura 8 muestra dos fotografías del montaje del robot explorador inteligente, con vista aérea y frontal.

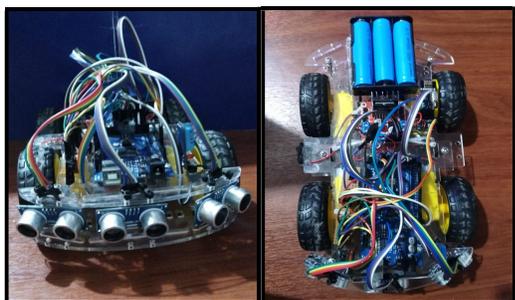


Fig. 8. Fotografías del montaje del robot explorador inteligente.

### C. Diseño del algoritmo de Red Neuronal

Consistió en el diseño de un algoritmo de control inteligente para el robot explorador desarrollado, de tal forma que controle el giro de los motorreductores mediante señales digitales. Dicho algoritmo de control es gestionado por una Red Neuronal Artificial que fue diseñada y entrenada con apoyo del software Matlab y la función “nntool”, la cual permite importar, crear, utilizar y exportar tanto datos como modelos de redes neuronales.

Asimismo, se seleccionaron tres diferentes estructuras de redes neuronales que emplearon el mismo tipo de función de activación, tangente hiperbólica, en todas sus capas, pero con diferentes números de capas y cantidad de neuronas por cada una de ellas; además, los valores de salida se correspondieron a 0 o 1, que representan los dos estados posibles de la salida digital que permitió el control de los motores. Entonces, la primera estructura seleccionada estuvo compuesta por tres capas y con la siguiente cantidad de neuronas: 12, 8 y 4. La segunda estructura seleccionada estuvo compuesta de cuatro capas y con la siguiente cantidad de neuronas: 12, 8, 6 y 4. Y,

la tercera estructura seleccionada estuvo compuesta de cuatro capas y con la siguiente cantidad de neuronas: 5, 6, 7 y 4.

De esta manera, de las tres estructuras propuestas, se eligió la tercera, debido a que otorgó mejores resultados respecto al número de iteraciones, performance, gradiente, y resultado de la simulación en el menor tiempo de entrenamiento, además de ser la única en la que el número de comprobación de validación fue de 0 al finalizar el entrenamiento. Asimismo, la base de datos empleada correspondió a 729 casos diferentes, los cuales fueron representados por vectores de 3 elementos. A su vez, los valores de estos vectores fueron discretizados en 8 niveles dentro de un rango establecido. Asimismo, con la estructura de red neuronal seleccionada se logró obtener una mejor precisión en la toma de decisión para efectuar el desplazamiento, así como una mejor evasión de obstáculos hallados en el camino. La estructura de la Red Neuronal y los parámetros utilizados en el entrenamiento se detallan en la figura 9.

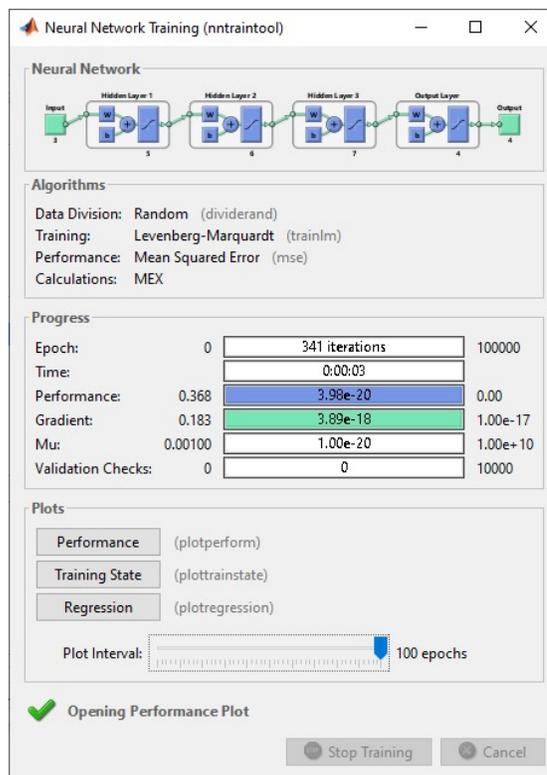


Fig. 9. Resultado del Entrenamiento de la Red Neuronal en Matlab

Los datos de entrada a la red neuronal, conformados por vectores de 3 elementos, correspondieron a las distancias obtenidas por los sensores de ultrasonido HC-SR04. Estas distancias fueron acondicionadas a partir de la expresión matemática (4), con el objetivo de situarlas en el rango de -1 a 1. Donde -1 corresponde a la distancia máxima, y 1 cuando la distancia es 0.

$$entrada_i = \left( \frac{-2}{d_{MAX}} \times d_{sensor\ i} \right) + 1 \quad (4)$$

Donde:

$$d_{MAX} = 50.0 \text{ cm.}$$

$$i = 1, 2, 3$$

Seguidamente, la transferencia de datos obtenidos del sensor de temperatura/humedad DHT11 y los valores de las entradas obtenidas de la expresión matemática (4), se realizó a través del módulo transmisor/receptor NRF24L01 mediante el envío de un solo paquete de datos, y utilizando una carga útil de 20 bytes hacia la estación receptora.

Por otro lado, el código de programación establecido en la estación receptora se encargó de procesar los datos enviados por el robot explorador inteligente, para luego enviarlos a través del puerto serial hacia un computador, en donde fueron procesados con apoyo del software Matlab. Asimismo, estos datos de salida de la Red Neuronal fueron enviados al robot explorador, por medio de la estación receptora, para el control de su desplazamiento y los de temperatura/humedad fueron mostrados en un gráfico que se actualiza en tiempo real utilizando el entorno de programación del propio Matlab. El flujo de datos mencionado se observa en la figura 10. Por otro lado, el LCD de la estación receptora se encargó de mostrar la cantidad de veces que los datos provenientes de los ultrasonidos y del sensor DHT11 fueron recibidos por la estación receptora, así como la cantidad de veces que los datos de salida de la red neuronal fueron enviados exitosamente al explorador, con la finalidad de corroborar que se ha realizado una comunicación exitosa. Todos los códigos de programación fueron enviados a los Arduino Mega mediante la IDE del Arduino.

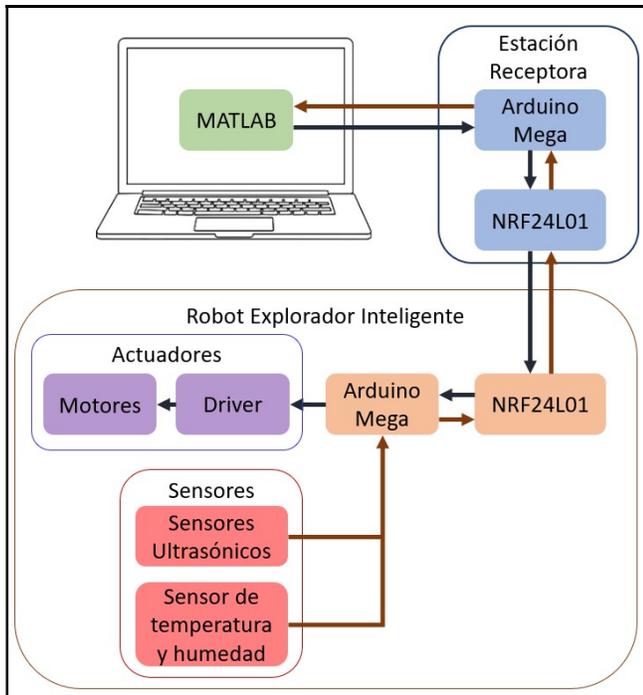


Fig. 10. Diagrama de flujo de datos entre el robot explorador inteligente, la estación receptora y el software Matlab

A continuación, en la figura 11, se muestra una parte del código de programa correspondiente a la comunicación entre módulo LCD y el Arduino.

```

...if (radio.available()){
    radio.read(dato,sizeof(data1));
    lcd.setCursor(0,0);
    lcd.print("Received ");
    i++;
    lcd.print(i);
    ... }
...
while(ok==false){
    ok = radio.write(data2, sizeof(data2)); }
    lcd.setCursor(0,1);
    lcd.print("Sent ");
    j++;
    lcd.print(j);
    ...

```

Fig. 11. Código de programación correspondiente a la comunicación entre el LCD y el módulo Arduino.

#### IV. PRUEBAS Y RESULTADOS

Para corroborar la efectividad de la red neuronal se realizaron distintas pruebas y se establecieron numéricamente en las tablas I y II. En dichas tablas se observa el número de impactos, el número de evasiones y el número de acercamiento a obstáculos respecto al tiempo de ejecución del código de programa desarrollado. Además, el robot explorador fue probado solamente sobre una superficie lisa. Ver las tablas I y II.

TABLA I

NÚMERO DE IMPACTOS, EVASIONES Y OBSTÁCULOS RESPECTO AL TIEMPO DE EJECUCIÓN DEL PROGRAMA

Prueba	Tiempo (min)	Distancia efectiva (m.)	Nº de impactos	Nº de evasiones	Nº de obstáculos
1	5	6	6	61	67
2	7	6	14	74	88
3	10	6	13	90	103
4	8	6	13	81	94
5	7	6	12	76	88
6	9	6	10	87	97
7	6	6	7	71	78
8	5	6	7	64	71
9	7	6	11	73	84
10	10	6	12	93	105
Prom.	7.4	6	10.5	77	87.5

De la Tabla I se observa que se presentaron 10.5 impactos y 77 evasiones, de un total de 87.5 acercamientos a obstáculos en un tiempo total de 7.4 minutos, en promedio. Se observa también que hubo una distancia de comunicación efectiva de 6 metros en promedio, a espacio cerrado.

Igualmente, de la Tabla II, se observa que, de las diez pruebas realizadas, se alcanzó en promedio un 88% de aciertos y un 12% de fallas.

TABLA II  
PORCENTAJE DE ACIERTOS Y FALLAS EN LA EVASIÓN DE OBSTÁCULOS

Pruebas	% Aciertos	% Fallas
1	91.04%	8.96%
2	84.09%	15.91%
3	87.38%	12.62%
4	86.17%	13.83%
5	86.36%	13.64%
6	89.69%	10.31%
7	91.03%	8.97%
8	90.14%	9.86%
9	86.90%	13.10%
10	88.57%	11.43%
Prom.	88.00%	12.00%

Seguidamente, en las figuras 12 y 13, se muestran dos fotografías correspondientes a la estación receptora y al robot explorador inteligente, en funcionamiento. Y, en la figura 14, se muestra el gráfico generado por el software Matlab con los datos de humedad y temperatura enviados por el robot explorador, en tiempo real.

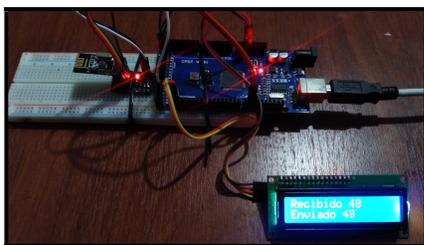


Fig. 12. Estación receptora en funcionamiento.

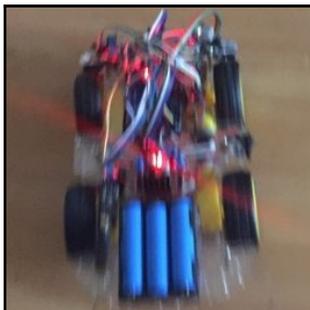


Fig. 13. Robot explorador inteligente en funcionamiento.

## V. CONCLUSIONES

Para el diseño del prototipo de robot explorador no se ha contemplado la situación de terrenos inclinados; por lo cual, se hace necesario aumentar la velocidad de giro de los motores. Además, se debe agregar un giroscopio que también deberá de suministrar la información a la red neuronal, para una toma de acción necesaria. Y de la misma manera, solamente se han utilizado 3 sensores de ultrasonido a diferencia del trabajo de [18] donde emplea 5 sensores, pero condicionado a realizar solamente tres tipos de movimientos básicos. Asimismo, la red

neuronal utilizada en este artículo fue del tipo multicapa a diferencia del trabajo realizado por [19], quien hace uso de una red neuronal convolucional. Y como también, se debe añadir un controlador para las baterías de litio, con la finalidad de evitar que se descarguen más de lo debido y se notifique en la estación receptora.

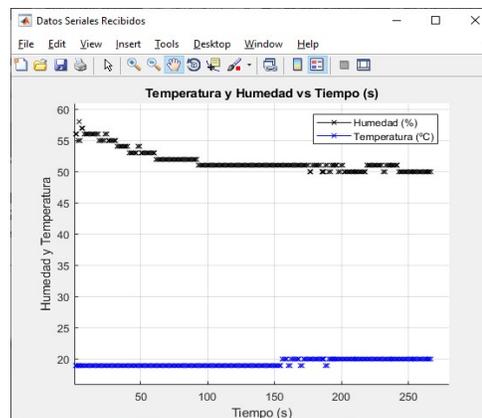


Fig. 14. Gráfico de Temperatura y Humedad Vs. Tiempo, generada en el software Matlab.

Asimismo, se comprobó la efectividad de las redes neuronales en el control de desplazamiento para la evasión de obstáculos; pues, se obtuvo un 88% de evasión de obstáculos sobre una superficie lisa. Así como también, se observó durante las pruebas que los momentos en los cuales el robot explorador colisionó, correspondió al instante cuando la cantidad de datos enviados aumentaba mientras la cantidad de datos recibidos se mantenía fija, es decir, se perdió la conexión con la estación receptora. Esto confirma que la acción de la red neuronal tuvo un desempeño importante, porque en promedio logró evadir obstáculos hasta en un 12%.

Por otro lado, en lo que respecta al tiempo de ejecución del programa algorítmico, fue equivalente a 7.4 minutos en promedio. Asimismo, existió una distancia de comunicación efectiva de hasta 6 metros en promedio, para las diez pruebas realizadas con el robot explorador.

## AGRADECIMIENTOS

Agradezco al Profesor Pedro Huamani Navarrete por su apoyo en la redacción de este artículo, por los alcances dados y su ayuda incondicional a este proyecto; así como también, a los demás profesores de la Universidad Ricardo Palma por contribuir con sus conocimientos científicos.

## REFERENCIAS

- [1] R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset and A. Erkmén, "Search and Rescue Robotics," Springer Handbook of Robotics, pp. 1151-1173, 2008.
- [2] A. Pandey, and D. Parhi. "Multiple Mobile Robots Navigation and Obstacle Avoidance Using Minimum Rule Based ANFIS Network Controller in the Cluttered Environment," International Journal of Advanced Robotics and Automation, vol. 1, no. 1, pp. 1-11, February 2016.

- [3] L. Hernando, M. Bueno y S. Sánchez. “Generación de trayectorias para un robot móvil empleando redes neuronales,” *Scientia Et Technica*, vol. 2, no. 39, pp. 94-99, Septiembre 2008.
- [4] M. Zamora-Izquierdo, L. Tomás-Balibrea, H. Martínez and A. Skarmeta. (2000, January). Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos. [Online]. Available: <https://pdfs.semanticscholar.org/4a46/ea08b4cdcbeacbc57191192680c3e3bdf0ee.pdf>
- [5] J. Vega and P. Huamani, “Neural control to simulate 4 autonomous navigation behaviors in a differential-drive mobile robot,” in 2017 IEEE URUCON, Montevideo, Uruguay, October 2017.
- [6] Ch. Liang-Hsuan and Ch. Cheng-Hsiung, “New approach to intelligent control systems with self-exploring process,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 1, pp. 56-66, February 2003.
- [7] M. Hagan, H. Demuth, M. Beale and O. De Jesús. (2019, September 15). *Neural Network Design* (2nd ed.). Available: <https://hagan.okstate.edu/NNDesign.pdf>
- [8] S. Haykin. (2019, September 15). *Neural Networks and Learning Machines*, (3th ed.). 3th ed. New York: Pearson Prentice Hall, 2009.
- [9] Naylamp Mechatronics. Arduino Mega 2560 R3. (s.f.) [Online]. Available: [https://naylampmechatronics.com/arduino-tarjetas/9-arduino-mega-2560.html?search\\_query=arduino+mega&results=22](https://naylampmechatronics.com/arduino-tarjetas/9-arduino-mega-2560.html?search_query=arduino+mega&results=22)
- [10] Naylamp Mechatronics. Driver Puente H L298N. (s.f.) [Online]. Available: [https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html?search\\_query=motor+12+&results=22](https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html?search_query=motor+12+&results=22)
- [11] L. Llamas. (2019, october). Comunicación inalámbrica a 2.4Ghz con Arduino y NRF24L01 (s.f.). [Online]. Available: <https://www.luisllamas.es/comunicacion-inalambrica-a-2-4ghz-con-arduino-y-nrf24l01/>
- [12] Naylamp Mechatronics. (2019, october). Sensor de temperatura y humedad relativa DHT11 (s.f.) [Online]. Available: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html>
- [13] Naylamp Mechatronics. Sensor Ultrasonido HC-SR04 (s.f.) [Online]. Available: <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- [14] Naylamp Mechatronics. Motor DC con caja reductora 6V 200rpm y llanta goma (s.f.) [Online]. Available: <https://naylampmechatronics.com/motores-dc/20-motor-dc-caja-reductora-y-llanta-goma.html>
- [15] Wikipedia. Batería de ion de litio (s.f.) [Online]. Available: [https://es.wikipedia.org/wiki/Bater%C3%ADa\\_de\\_ion\\_de\\_litio](https://es.wikipedia.org/wiki/Bater%C3%ADa_de_ion_de_litio)
- [16] Naylamp Mechatronics. Display Alfanumérico LCD 1602 (s.f.) [Online]. Available: <https://naylampmechatronics.com/displays/110-display-lcd1602-azul-backlight.html>
- [17] Naylamp Mechatronics. Módulo adaptador LCD a I2C (s.f.) [Online]. Available: <https://naylampmechatronics.com/displays/60-modulo-adaptador-lcd-a-i2c.html>
- [18] M. Marcos, R. Guzmán and R. Alaiz, “Autoguiado de Robots Móviles mediante Redes Neuronales,” in XXV Jornadas de Automática, Ciudad Real, España, Setiembre 2004. <https://intranet.ceautomatica.es/old/actividades/jornadas/XXV/documentos/56-oclideaiz.pdf>
- [19] F. Lotfi, V. Ajallooeian and H. Taghirad, “Robust Object Tracking Based on Recurrent Neural Networks,” in 6th RSI International Conference on Robotics and Mechatronics (IcRoM), Tehran, Iran, pp. 507-511, October 2018.