

## DESARROLLO Y APLICACIÓN DE ROBOT AUTÓNOMO MÓVIL

\*Luis Carlos Alberto Espinosa Alvarado

\*Carlos Castorena Castillo

\*\*Gerardo Nery Lara Coronado

\*\*Daniel Rodríguez Medrano

\*\* Docentes de la carrera Ingeniería en Mecatrónica en la Universidad Emiliano Zapata. Contacto: luis.espinosa@unez.edu.mx, carlos.castorena@unez.edu.mx

\*\*\*Estudiantes de la Ingeniería en Mecatrónica por parte de la Universidad Emiliano Zapata. Han realizado su práctica mecatrónica en la misma institución.

Recibido: 15 de abril de 2023.

Aceptado: 15 de mayo de 2023.

### Resumen

Trabajar y desarrollar este proyecto involucró explorar nuevos métodos de desarrollo de programación, se trabajó arduamente en cada una de las etapas. El objetivo principal fue que un auto se moviera por sí mismo y lograra detectar todo lo que estaba dentro de su rango de visión. El resultado final del proyecto fue positivo.

El desarrollo de un vehículo autónomo es de suma importancia hoy en día; ha llegado a ser indispensable para la industria 4.0, y puede desarrollar actividades y tareas precisas sin necesidad de una intervención directa humana, además de la libertad de movimientos y llegar a coordinar acciones con otros robots autónomos mediante redes neuronales, algoritmos genéticos, entre otros métodos de programación de Inteligencia Artificial.

**Palabras clave:** Auto. Visión. Neuronal. IA.

### Abstract

Working and developing this project involved exploring new methods of programming development, so it implied hard work at each stage. The main objective was that the car moved by itself

and managed to detect everything that was within its range of vision. The work gave good results.

The development of an autonomous car is very important in these days; it has become essential for industry 4.0, and it can develop precise activities and tasks without the need for direct human intervention as well as freed in its movements, and coordinate actions with other autonomous robots through neural networks, genetic algorithms among other Artificial Intelligence programming methods.

**Keywords:** Car. Vision. Neural. AI.

En la actualidad los robots autónomos móviles son realmente una necesidad en la industria para la automatización de diferentes procesos. Estos robots cuentan con una amplia eficiencia y precisión sobre los procesos de producción. Además, los robots móviles pueden laborar en entornos de alto riesgo para los humanos, reduciendo accidentes o lesiones.

En general estos robots móviles ofrecen muchos beneficios que pueden apoyar a mejorar las operaciones y resultados finales de cada proceso

a realizar, además de que algunos de ellos cuentan con Inteligencia Artificial para realizar las tareas más complejas, como la manipulación de objetos y recolección de datos, permitiéndoles una alta velocidad y precisión, sin la necesidad de la supervisión humana.

En ese sentido, la Inteligencia Artificial es una herramienta muy importante para la mejora de productividad y la eficiencia de muchos procesos como la automatización de las tareas repetitivas y de la evolución que involucra la toma de decisiones, siendo muy útil para las diferentes aplicaciones, como el control de sistemas en las industrias, educación y medicina.

### Robot autónomo móvil

La máquina que se puede reprogramar para realizar diferentes habilidades y movimientos proyectados para la realización de una secuencia de tareas, tiene similitud con el humano por su habilidad de desarrollar el trabajo.

En ese orden de ideas, un vehículo autónomo móvil es un robot capaz de moverse y realizar tareas sin la supervisión de un operador humano. Estos robots cuentan con muchas aplicaciones en las diferentes ramas de la industria.

Hay que tener en cuenta que los vehículos autónomos móviles son cada vez más comunes en la actualidad, y esto se debe a los avances de la robótica, la cual ha permitido que sean capaces de moverse y realizar tareas con una alta precisión y velocidad. Aunado a esto, los robots pueden trabajar en espacios peligrosos donde los trabajadores humanos no pueden llegar o donde corren el riesgo de un accidente, evitándose pérdidas temporales, económicas y humanas.

### Inteligencia Artificial

La Inteligencia Artificial es una rama de la informática que permite realizar tareas o actividades que normalmente requiere la inteligencia

humana, es diseñada para realizar tareas específicas como la toma de decisiones, el aprendizaje autónomo, el reconocimiento de patrones y el análisis de datos.

### Red neuronal

Por su parte, la red neuronal es una tecnología de aprendizaje que permite a los sistemas aprender de los datos que se le proporcionan, y se basa en la forma que funciona un cerebro humano; Dicha red con facilidad puede llegar a conclusiones nunca antes alcanzadas sobre ciertos datos, esto es permitido gracias a los pesos asignados en cada neurona, la cual se ajusta una y otra vez para encontrar los patrones óptimos en los datos, como se presenta a continuación en la figura 1

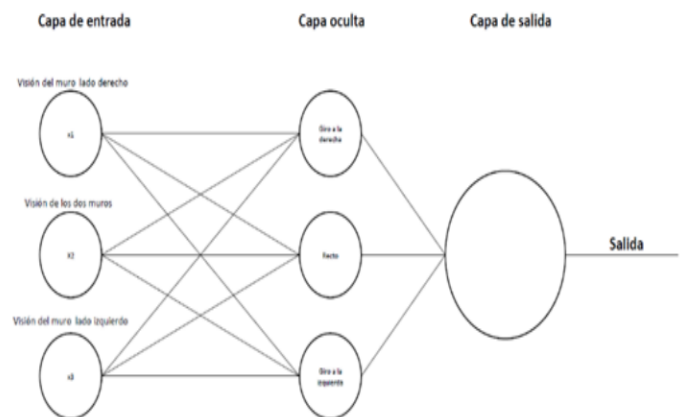


Figura 1. *Arquitectura de la red neuronal.* Fuente: elaboración propia.

### Entrenamiento de la red neuronal

En un primer momento, el robot realiza un recorrido manual. En este caso se controla con un mando a distancia, esto para recolectar imágenes de la pista para su entrenamiento de la toma de decisiones dando un giro de izquierda, derecha o simplemente de manera recta.

Para esta red neuronal se encuentran tres factores que influyen en la toma de decisiones, los cuales son la visualización de los muros de contención de lado derecho e izquierdo.

Por mencionar un ejemplo, cuando el carro detecte solamente el muro de contención de lado derecho, su giro tiene que ser hacia el lado izquierdo. Cuando el robot solo proyecte una visualización del muro de contención de lado izquierdo su funcionalidad es girar al sentido contrario, pero cuando el robot visualiza los dos muros de contención, debe realizar una trayectoria recta, como se muestra en la siguiente figura 2.

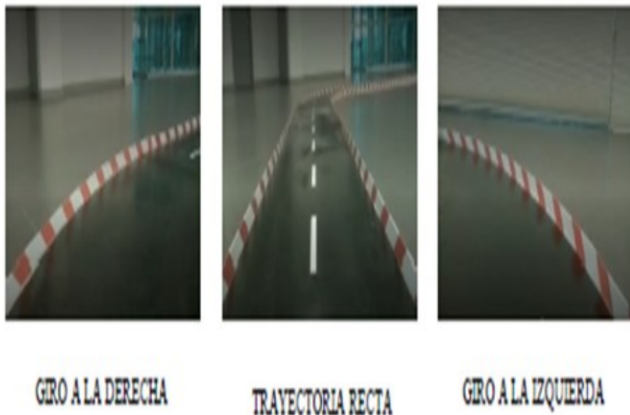


Figura 2. Visualización del robot. Fuente: elaboración propia.

Con la recolección de datos, contenida en las imágenes grabadas mediante el recorrido, el robot puede procesar su respuesta a través de la toma de decisiones que crea la red neuronal; así, el modelo recibe la información y aprende a relacionar las nuevas referencias de entrada con el comportamiento que se desea obtener.

### Hardware

Para el funcionamiento de este robot se utilizaron diferentes tipos de hardware, los cuales se encuentran: motor-reductor, puente H modelo L298, sensor de inclinación, sensor ultrasónico, módulo regulador de voltaje, batería de litio 18650, cámara y procesador.

Se toma la decisión de adquirir cuatro motores reductores que son piezas claves en el robot, ya que estos permiten moverse con precisión, además de reducir la velocidad del motor y le permi-

te un mayor desempeño. Los motores de 500 rpm ayudan al desplazamiento suave y con un mayor control.

Para el control de estos motores se le conecta el puente H modelo L298 que permite controlar al vehículo moviéndose a la velocidad necesaria sin tener que controlar cada motor por separado; además, ayuda a que los movimientos sean más precisos.

El sensor de inclinación se instala para la detección de cambios en la superficie de la pista, y cuando éste se activa, manda una señal a los motores, aumentando su velocidad; de igual manera, cuando éste se encuentre en una pendiente ascendente de  $20^\circ$  y se active en una pendiente de  $20^\circ$  descendente, los motores reducen su velocidad para evitar algún descarrilamiento.

La función que realiza el sensor ultrasónico modelo HC-SR04 es medir la distancia que está entre el robot y el muro de contención, y ayuda a corregir la trayectoria cuando esté a 7 cm del muro. Como fuente de alimentación, se utilizaron dos paquetes de baterías 18650, ya que aportan 7.4 volts, además de que entregan 3800 mAh (miliamperios hora) de potencia con una mayor duración de baterías, siendo unas relativamente pequeñas, ideales para este prototipo, ya que tienen un tamaño pequeño y peso ligero, alta capacidad, y son muy portátiles.

Para el procesamiento, control y ejecución se utilizó una Raspberry Pi, ya que esta tarjeta es de bajo consumo de energía y de alta velocidad de procesamiento; asimismo, almacena datos, tiene conectividad inalámbrica, y cuando se le añade la cámara de Raspberry Pi, el robot puede usarla para obtener la información visual del entorno donde se encuentra explorando.

### Programación

En esta etapa se realizó la programación en lenguaje *Python*, que es utilizado en las aplica-

ciones web, el desarrollo de software, la ciencia de datos y el *Machine Learning* (ML). Es un lenguaje de alto nivel, el cual contiene implícitas algunas estructuras de datos, como listas, diccionarios y conjuntos, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible. Para el vehículo autónomo, se desarrolló como base la siguiente estructura.

**Paso 1.** Utilización de *datetime* para manejar correctamente las fechas, *tensorflow* para crear la red neuronal y, finalmente, se realizó un gráfico sencillo de la función de pérdida mediante *matplotlib*. En otras circunstancias, se utilizaría otra paquetería para el gráfico, pero se optó por este paquete considerando el modelo en cuestión.

**Paso 2.** Se utilizó una función para cargar los datos de un archivo CSV (siglas en inglés de *valores separados por comas*) en un marco de datos de pandas.

**Paso 3.** Se separaron las características de los datos en  $X$  del valor que se quiere predecir  $Y$ , se pretendió el dato de la última columna del archivo CSV.

**Paso 4.** Se dividieron los datos de  $(X, Y)$  en dos conjuntos, de entrenamiento y de prueba, el tamaño de ésta se establece en la función (*test\_size*) que para el caso es de 0.2 (20%). Y la función (*random state*) aseguró que los conjuntos de entrenamiento y prueba fueran generados de forma aleatoria.

**Paso 5.** Se realizó una transformación de los datos de entrenamiento y de prueba escalando los valores una distribución normal centrada en 0 y con una desviación estándar de 1.

**Paso 6.** Se agregó capacidad a la red neuronal para construir un modelo de aprendizaje

**Paso 7.** Se añadió la primera capa de la red neuronal de tipo densa, la cual es una capa de entrada con 3 nodos de entrada conectadas a 3

nodos de la capa oculta, usando una inicialización uniforme para los pesos y una función de activación para la capa oculta.

**Paso 8.** Se agregó una segunda capa oculta al modelo con 3 neuronas, una inicialización uniforme y una función de activación relu.

**Paso 9.** Se añadió una capa de salida del modelo, la cual tiene una unidad, una inicialización del kernel uniforme y una función de activación sigmoide.

**Paso 10.** Compilación de la red neuronal para poder utilizarla; se configuró el optimizador, la función de pérdida y las métricas que se iban a usar para el progreso de la red neuronal.

**Paso 11.** Se entrenó la red neuronal con los datos almacenados en  $X_{train}$  t  $Y_{train}$ , esto se realizó en los conjuntos de 10 datos y se repitió durante 100 ciclos de entrenamiento.

**Paso 12.** Se evaluar y realizó la predicción con los datos de prueba guardados en  $(X_{test})$ , y se comparó con los resultados reales almacenados en  $(Y_{test})$ , convirtiendo la predicción en binario, o sea, verdadero o falso, utilizando un umbral de 0.5.

**Paso 13.** Se utilizó el modelo entrenado para hacer las predicciones en conjunto de los datos de prueba, se utilizó el modelo entrenado y luego se evaluó la predicción para resolver si era mayor que 0.5, finalizando con una matriz booleana.

```
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import Adam

def build_model():
    model = Sequential([
        Dense(64, activation='relu'),
        Dense(64, activation='relu'),
        Dense(64, activation='relu'),
        Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    return model

def train_model():
    model = build_model()
    model.fit(x_train, y_train, validation_data=(x_val, y_val), epochs=100, batch_size=32)

def predict_model():
    model = build_model()
    model.predict(x_test)
```

Figura 3. Red neuronal utilizada. Fuente: elaboración propia.

### Resultados

Diariamente se realizaron distintas pruebas en diferentes pistas con diferentes tipos de suelo para lograr que el robot autónomo móvil identificara cada uno de ellos, obteniendo efectos muy favorables en cuanto a visión y desarrollo. De esta forma, en la pista obtuvo un desempeño de una manera ágil y con velocidad estable, girando en las curvas de una forma autónoma debido a que tenía una amplia visión de todos los objetos dentro de su campo y resolución de la pista.

El robot autónomo móvil realizado y entrenado mediante la red neuronal, fue capaz de realizar tareas complejas con un buen nivel de preci-

sión, incluso interpretó la prueba y se mantuvo constante en su desempeño durante la competencia. Con estas evidencias, se demuestra que el robot autónomo es una herramienta muy útil para diversas aplicaciones, cada vez más complejas, ya que, por su autonomía, pudo predecir los giros en función de los obstáculos, completando el recorrido de la pista de competición.

### Conclusión

La inteligencia artificial es el gran paso que la humanidad necesitaba dar para disminuir accidentes viales, ya que esta innovación puede llegar incluso a salvar vidas humanas que en ciertos contextos podrían perderse. En la actualidad, ya se encuentran diversos vehículos autónomos aprobados para su utilización en múltiples países; algunos de los más populares los ofertan marcas como *Mercedes Benz*, *Tesla*, entre otros.

### Bibliografía

- Van Rossum, G., y Drake, F.L. (Eds). (2010). *The Python Library Reference. Teoría de redes Limitada.*
- Bolton, W. (2013). *Mecatrónica. Sistema de control electrónico en la ingeniería mecánica y eléctrica. Un enfoque multidisciplinario. Alfaomega.*