



Ingenius. Revista de Ciencia y Tecnología  
ISSN: 1390-650X  
ISSN: 1390-860X  
revistaingenius@ups.edu.ec  
Universidad Politécnica Salesiana  
Ecuador

## OPERACIÓN REMOTA DE UN ROBOT MÓVIL USANDO UN TELÉFONO INTELIGENTE

---

Flores-Vázquez., Carlos; Rojas de Silva G, Fco. Abiud; Trejo Ramírez, Karla A.

OPERACIÓN REMOTA DE UN ROBOT MÓVIL USANDO UN TELÉFONO INTELIGENTE

Ingenius. Revista de Ciencia y Tecnología, núm. 17, 2017

Universidad Politécnica Salesiana, Ecuador

**Disponible en:** <https://www.redalyc.org/articulo.oa?id=505554801004>

**DOI:** <https://doi.org/10.17163/ings.n17.2017.04>



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 3.0 Internacional.

OPERACIÓN REMOTA DE UN ROBOT MÓVIL USANDO UN TELÉFONO INTELIGENTE

REMOTE OPERATION OF A MOBILE ROBOT USING A SMARTPHONE

*Carlos Flores-Vázquez.*

*Universidad Católica de Cuenca, Ecuador*

*cfloresv@ucacue.edu.ec*

 <http://orcid.org/0000-0002-5159-9469>

DOI: <https://doi.org/10.17163/ings.n17.2017.04>

Redalyc: <https://www.redalyc.org/articulo.oa?id=505554801004>

*Fco. Abiud Rojas de Silva G*

*Universidad Politécnica de Cataluña, España*

*Karla A. Trejo Ramírez*

*Universidad Politécnica de Cataluña, España*

Recepción: 30 Septiembre 2016

Aprobación: 16 Diciembre 2016

## RESUMEN:

En este artículo se presenta un acercamiento al mando a distancia de un robot móvil que emplea un teléfono inteligente para comandarlo. La idea principal es recolectar los datos generados por el acelerómetro incluido en el teléfono inteligente. Los datos son los resultados de mover el teléfono en la dirección de los ejes Y y Z. Tales datos serán usados para entrenar dos redes neuronales que definirán la dirección del movimiento del robot móvil. Las salidas obtenidas de las redes neuronales serán procesadas para calcular y trazar la trayectoria, que es determinada por el modelo cinemático para un robot móvil tipo triciclo.

**PALABRAS CLAVE:** sistemas remotos, redes neuronales, robot móvil, teléfono inteligente, acelerómetro.

## ABSTRACT:

This paper presents an approach to remote control a robot using smartphone. The main idea is to collect data generated by the accelerometer sensor included in the smartphone. The data are the results of moving the smartphone in direction of the axis Y and Z. Such data will be used for training two neural networks that will define the direction of the movement of the mobile robot. The outputs obtained from the neural networks will be processed to compute and plot the trajectory, which is determined by the kinematic model for a tricycle mobile robot.

**KEYWORDS:** Remote System, Neural Network, Mobile Robot, Smartphone, Accelerometer.

## FORMA SUGERIDA DE CITACIÓN:

Flores, C.; Rojas, F.; Trejo, K. (2017). «Operación remota de un robot móvil usando un teléfono inteligente». INGENIUS. N°. 17, (Enero-Junio). pp. 29-35. ISSN: 1390-650X.

## 1. INTRODUCCIÓN

La operación remota ha sido desarrollada durante muchos años y ha conducido a diversos usos prácticos de interacción del usuario con el entorno. El concepto de sistema remoto será abordado como un sistema para manipular y controlar un robot que requiere un operador humano a distancia. Esto implica la disponibilidad de interfaz de hombre-máquina [1].

Su función principal es brindar al operador la capacidad de realizar tareas complejas en entornos peligrosos y menos estructurados, por esta razón es necesario aplicar una concepción del sistema que permita la interacción con el entorno. La Figura 1 muestra una generalización de este sistema [2].

Debido a la reducción de costos en la tecnología de dispositivos móviles, se consideró conveniente aprovechar su *hardware* para desarrollar esta propuesta. En el presente artículo se presenta una aplicación de un sistema remoto para un robot móvil empleando un teléfono inteligente como mando a distancia.

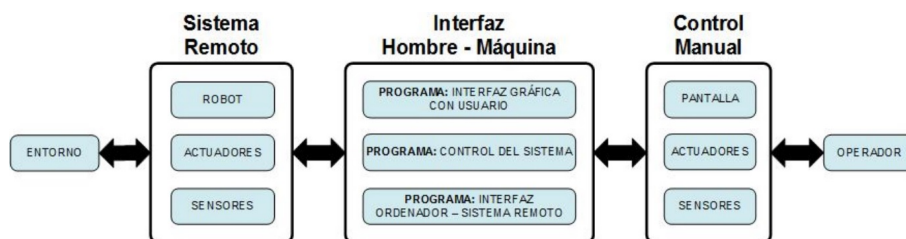


FIGURA 1  
Sistema remoto. [1]

## 2. ESTADO DEL ARTE

Durante las dos décadas pasadas diversos sistemas remotos han sido desarrollados usando varios medios de transmisión de datos, algunos de ellos están basados en protocolos de Internet. Empleando un navegador web, un operador a distancia es capaz de controlar un robot móvil, un mapa es generado y visualizado en pantalla para la navegación, simulando el entorno real [3]. Esta clase de sistema por lo general se usa para apoyar operaciones subterráneas, situaciones de rescate en desastres, entre otras. Sin embargo, el empleo de Internet puede ser una restricción en ciertas áreas donde no se encuentra disponible; adicionalmente, los retardos en la transmisión son la principal desventaja de esta clase de sistemas.

Tomando esto en consideración, otros sistemas han sido desarrollados para mejorar esta desventaja que usa estructuras complejas de control para compensar los retardos de tiempo distancia [3]. Como alternativa para lugares donde no exista servicio de Internet, Kimitsuka, Susuki y Sawai han desarrollado un sistema remoto aprovechando una red inalámbrica de sensores (WSN), [4]. Sistemas recientes de teleoperación emplean interfaces de usuario convencionales como teclados, palancas de mando, controles para juegos, etc. [2]. Actualmente, muchas investigaciones trabajan con dispositivos comerciales, teléfonos inteligentes como mando a distancia para actividades de control, visualización y vigilancia. [5].

Un número importante de investigaciones en el desarrollo de sistemas remotos usan dispositivos de interfaz humana (HID) como el ratón, el teclado, palancas de mando o pantallas táctiles, para mejorar la experiencia de interacción, considerando la destreza y ergonomía de la anatomía humana. Recientemente dispositivos de sinestesia (Haptic), el procesamiento de videos y dispositivos de reconocimiento gestual se hacen más comunes, cambiando la interacción con ordenadores, teléfonos móviles y videojuegos. [2].

Los últimos teléfonos inteligentes ofrecen mayores capacidades multimedia, sensores (acelerómetros), comunicación (*Bluetooth*, *Wifi*, *GPRS/UMTS*), ubicación (*GPS*) y facilidad de programación por lo cual son actualmente los mejores candidatos para servir de interfaz entre las personas y su entorno [6].

Las capacidades tecnológicas de los teléfonos inteligentes han generado el interés para investigar fenómenos físicos como caídas [7] o el caminar humano [8,9]. La Figura 2 ilustra la secuencia para obtener señales de verificación e identificación del caminar usando el acelerómetro [10]. Esta estructura fue usada para la obtención de señales en nuestra propuesta.

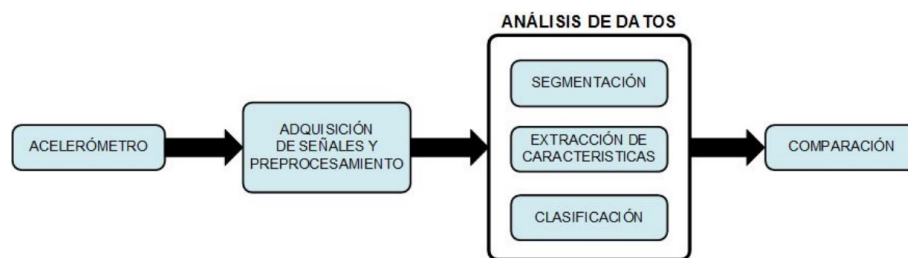


FIGURA 2

Procesamiento de señal para la identificación y verificación del caminar humano.

### 3. METODOLOGÍA PROPUESTA

Esta propuesta presenta el diseño e implementación de un sistema remoto para un robot móvil, controlando su trayectoria mediante un teléfono inteligente: La selección del teléfono inteligente como mando a distancia se debe a su disponibilidad, la alta aceptación social, el manejo intuitivo y el costo. Las órdenes de giro y desplazamiento del robot móvil se realizan mediante el movimiento del dispositivo.

Las señales obtenidas por los sensores del teléfono inteligente, serán procesadas usando un algoritmo de reconocimiento de patrones, la trayectoria resultante se puede visualizar en el ordenador. Este algoritmo será responsable de clasificar los movimientos que el robot móvil debería realizar: aceleración (+), aceleración (-), giro a la derecha, giro a la izquierda, según los movimientos correspondientes hechos con el teléfono inteligente. Por esta razón, hemos puesto en práctica dos redes neuronales. Para desarrollar la generación de trayectorias del sistema remoto se procedió de la siguiente forma:

1. Selección de una aplicación móvil (App) para adquisición de datos
2. Adquisición de datos y registro de estos.
3. Creación de una red neuronal.
4. Entrenamiento y simulación de la red neuronal.

El primer paso para desarrollar este proyecto fue definir la App que permita extraer los datos del acelerómetro. Después del análisis de varias App como *Wireless IMU*, *SensorUdp*, *iSeismo*, entre otras, que transmiten las mediciones de los sensores internos del teléfono inteligente. La selección fue *MlconnectApp*, que permite capturar los datos generados en el teléfono por el programa *Matlab* instalado en un ordenador [11], realizando una adquisición fácil y directa de los datos tratados. Las transmisiones usan protocolo *UDP* que cumple con las necesidades de velocidad para este sistema.

#### 3.1. Adquisición de datos

Para la adquisición de las señales en los sensores es necesario definir ejes de referencia en la estructura física del teléfono, como se observa en la Figura 3. El eje Z es asignado a los movimientos de aceleración (+Z aceleración (+), -Z aceleración (-)); el eje Y corresponde a los giros (+Y giro a la izquierda, -Y giro a la derecha).



FIGURA 3  
Ejes asignados en el teléfono inteligente

El equipamiento tecnológico usado consiste de un teléfono inteligente *Sony Xperia* tipo, con sistema *Android*, en él se instaló la *App MLConnect* versión libre que obtiene las señales de sensor de aceleración. Los datos son capturados en tiempo real y convertidos en una matriz. Esta matriz almacenada en *Matlab* usa datos en el rango desde 10 a -10 con cuatro decimales de precisión.

Eje Z: Tres registros de datos fueron generados con 1000 datos de entrada a los cuales se asignaron los objetivos correspondientes a los movimientos descritos considerando su rango de operación como se observa en la Tabla 1.

Los movimientos posibles en el eje Z se detallan a continuación:

- Inactivo: El estado donde el teléfono se encuentra en una posición neutral, lo cual implica una postura natural del operador para sostener el teléfono.
- Aceleración (+): Implica la orden de realizar el movimiento para desplazamiento frontal del robot.
- Aceleración (-): Posee dos interpretaciones, la primera si el robot se encuentra en movimiento frenar, la segunda orden si el robot se encuentra estático la dirección para desplazamiento del robot es en retroceso.

Eje Y: De similar forma al eje Z, tres registros de datos fueron generados con 1000 datos de entrada a los cuales se asignaron los objetivos correspondientes a los movimientos descritos considerando su rango de operación como se observa en la Tabla 2.

Los movimientos posibles en el eje Y se detallan a continuación:

- Inactivo: Permite mantener la dirección recta, se realiza cuando no existe una inclinación evidente del teléfono a un costado.
- Giro a la derecha: Es ejecutado cuando el teléfono se encuentra notablemente inclinado al costado derecho del operador. La inclinación del teléfono se realiza en forma natural.

MOVIMIENTO	Eje Z				
	Rango de datos de ingreso	Objetivos			
Inactivo	2 a 6	0	1	0	
Aceleración (+)	6 a 10	1	0	0	
Aceleración (-)	-10 a 2	0	0	1	

TABLA 1  
Objetivos y rango de operación del eje Z.

MOVIMIENTO	Eje Y				
	Rango de datos de ingreso	Objetivos			
Inactivo	-2 a 2	0	1	0	
Aceleración (+)	-2 a -10	0	0	1	
Aceleración (-)	2 a 10	1	0	0	

TABLA 2  
Objetivos y rango de operación del eje Y.

- a) Giro a la izquierda: Es ejecutado cuando el teléfono se encuentra notablemente inclinado al costado izquierdo del operador.

Como se evidencia en la Tabla 1, los rangos de los datos de ingreso para el eje Z no son simétricos debido al hecho que cuando el usuario sostiene el teléfono de modo vertical, este no se posiciona perpendicular con respecto al plano Y.

Para realizar esta adquisición de datos, se modificó el código original del programa sobre *Matlab*, para obtener los datos de cada eje de forma independiente con el fin de procesarlos posteriormente. En este punto, dos redes neuronales fueron implantadas: la primera solo para reconocer los datos de desplazamiento (con tres neuronas sobre la capa oculta) y la segunda para los datos correspondientes a movimientos de giro (dos neuronas sobre su capa oculta), el tipo de red implementada se detalla más adelante.

Como se evidencia en las Tablas 1 y 2, tendremos un conjunto de datos de entrada para cada red y tres objetivos de salida.

### 3.2. Generación de la red neuronal

Para el diseño e implementación de las redes neuronales un análisis de varios tipos de redes existentes fue realizado con el fin de seleccionar las mejores características para este proyecto. Por consiguiente, de este análisis, se define la arquitectura para las redes neuronales a ser aplicadas.

CARACTERÍSTICAS FUNCIÓN	RED NEURONAL 1 DESPLAZAMIENTO	RED NEURONAL 2 GIROS
Número de ingresos	1	1
Número de capas ocultas	1	1
Número de salidas	3	3
Número de neuronas	3	2
Función de transferencia de las capas ocultas	Logsig	Logsig
Función de transferencia de las capas de salida	Purelin	Tansig

TABLA 3  
Resumen de características de las redes neuronales implementadas.

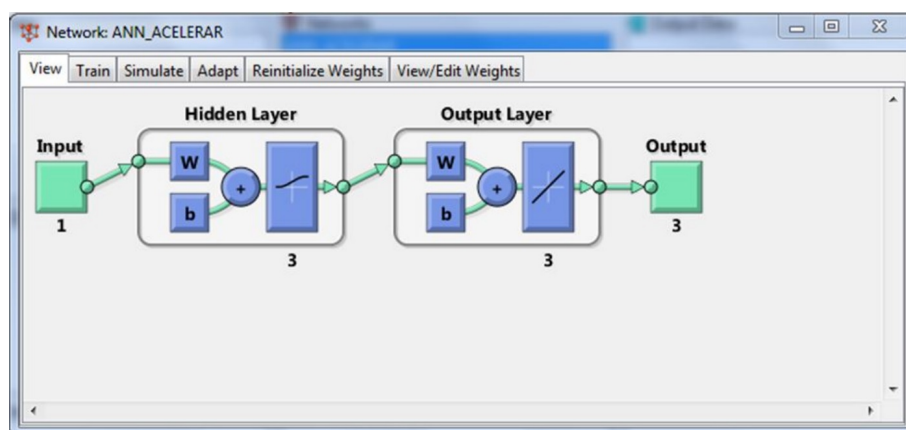


FIGURA 4  
Red neuronal para el desplazamiento.

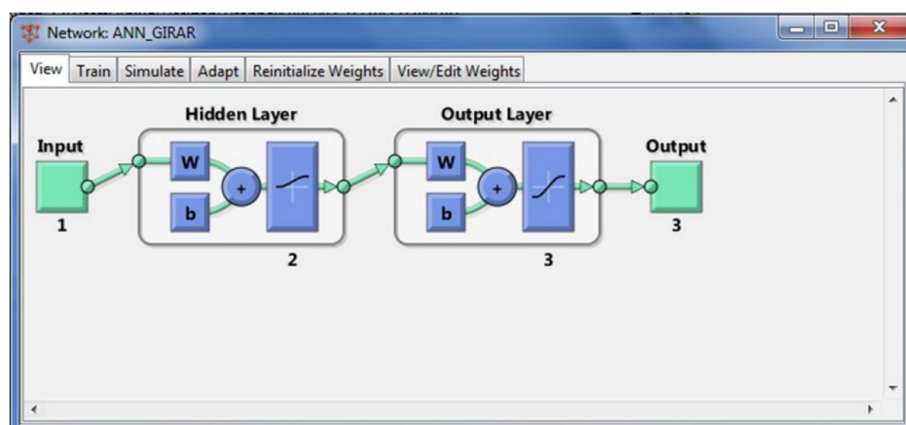


FIGURA 5  
Red neuronal para los giros.

1. Se escogió una red supervisada (*Supervised Networks*) la cual puede ser entrenada para producir salidas predefinidas por el usuario y es adecuada para clasificación de datos que contienen ruido.
2. De los diferentes tipos de redes supervisadas se seleccionó la red Feedforward y de entre su clasificación la red Backpropagation. [12].
3. Para la función de entrenamiento de la red se escogió *TRAINLM Levenberg –Marquardt* por ser generalmente la más rápida con la selección previa de red *Backpropagation*.
4. La función de aprendizaje *LEARNGDM* que emplea gradiente descendente, para la ejecución se aplica el error cuadrático medio (MSE).



5. Las funciones de transferencia deben ser continuas. Por esta razón sobre la base de [13] se escogió para la capa oculta Logsig como función de transferencia y Pureline y Tansig para las capas de salida.

Basado en las características descritas, dos redes neuronales fueron diseñadas para controlar el desplazamiento (eje Z) y los giros (eje Y) del robot móvil usando el Neural Toolbox de Matlab. Los resultados son mostrados en la Tabla 3 y las Figuras 4 y 5.

De los 6000 datos adquiridos con sus respectivas etiquetas, se decide utilizar solo 600 para realizar el entrenamiento de las redes neuronales y 120 de ellos sin etiquetar para la simulación. Las Figuras 6, 7, 8, 9 muestran las gráficas de los datos usados. Se necesitó un intervalo de tiempo igual a 300 segundos para la adquisición de datos en la función de desplazamiento, en cuanto a la función de giros el intervalo de tiempo fue menor es decir 60 segundos. Esta diferencia de tiempo se debe a la variación de los datos cuando se realizaba la posición correspondiente al desplazamiento con el teléfono inteligente.

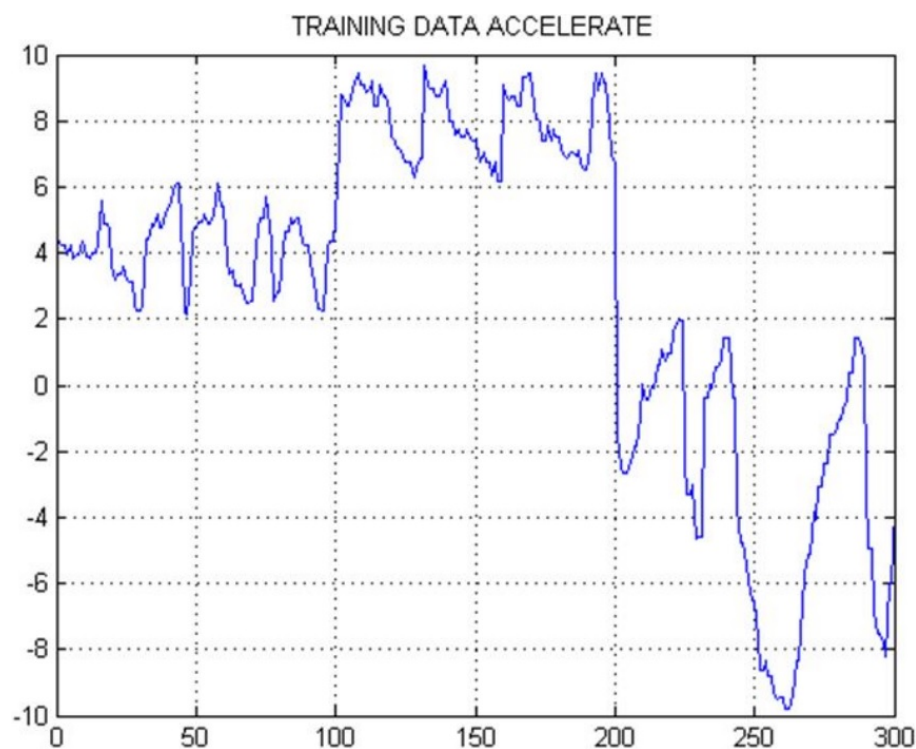


FIGURA 6

Datos para entrenamiento de la red neuronal de desplazamiento. Eje horizontal corresponde al tiempo de captura en segundos, el eje vertical corresponde al valor de los datos del acelerómetro.



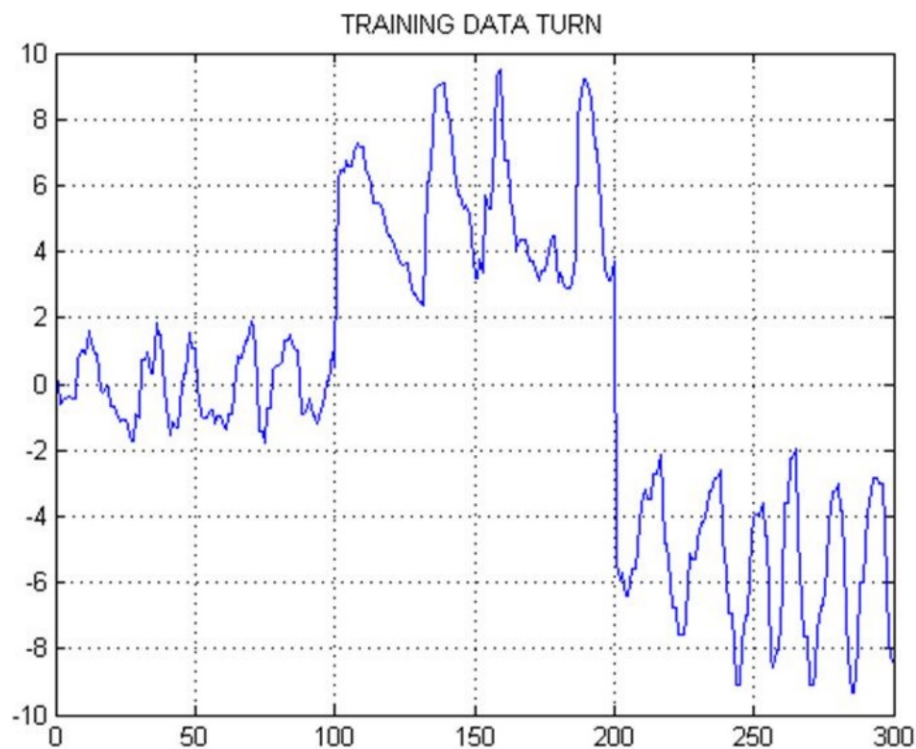


FIGURA 7

Datos para entrenamiento de la red neuronal de giros. Eje horizontal corresponde al tiempo de captura en segundos, el eje vertical corresponde al valor de los datos del acelerómetro.

### 3.3. Implementación

Una vez que las redes fueron entrenadas, simuladas para verificar que los datos eran suficientes y la operación de las redes era apropiada, se incluyó en el programa de adquisición de datos las redes neuronales de desplazamiento y giros.

Para aplicar las redes neuronales creadas, antes de controlar la trayectoria de un robot móvil las actividades siguientes fueron realizadas:

1. Inclusión de las redes neuronales en el programa de adquisición de datos sobre *Matlab*.
2. Establecer una ecuación para calcular la trayectoria de un robot móvil.
3. Pruebas y resultados.

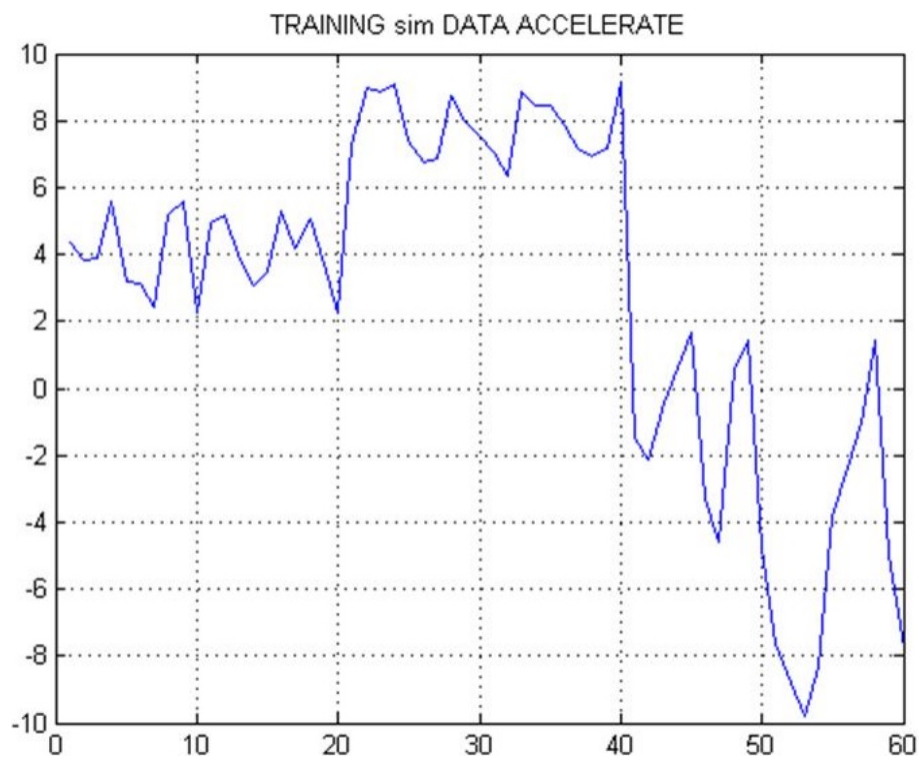


FIGURA 8

Datos para simulación de la red neuronal de desplazamiento. Eje horizontal corresponde al tiempo de captura en segundos, el eje vertical corresponde al valor de los datos del acelerómetro.

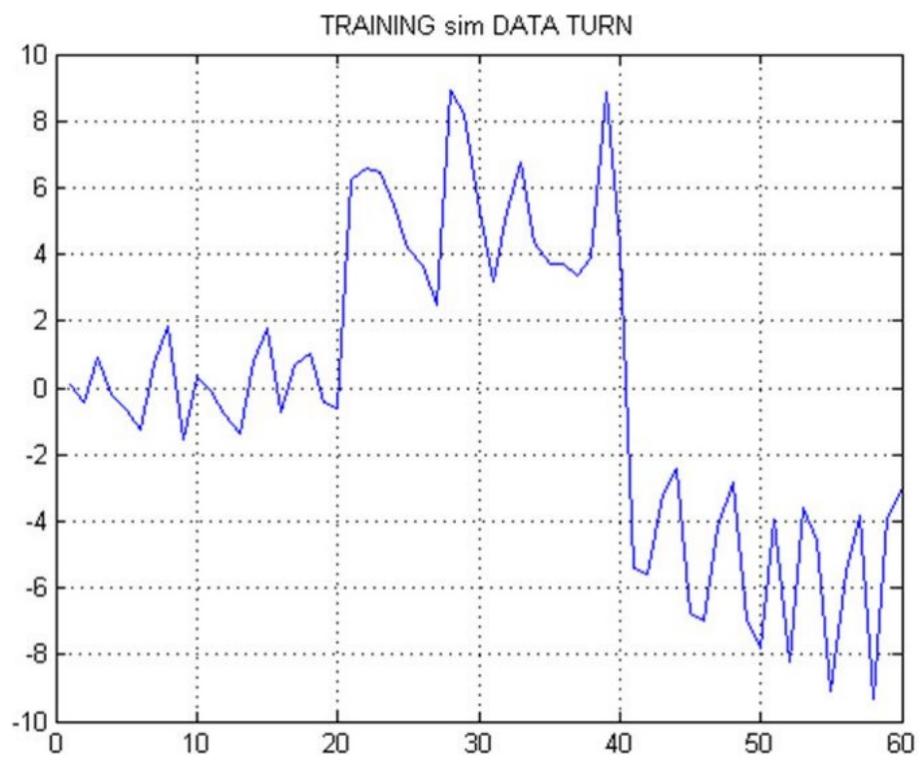


FIGURA 9

Datos para simulación de la red neuronal de giros. Eje horizontal corresponde al tiempo de captura en segundos, el eje vertical corresponde al valor de los datos del acelerómetro.

Para la inclusión de las redes neuronales, era necesario modificar una vez más el código original de *MLCONNECT* para adquisición de datos. Las modificaciones realizadas se describen a continuación:

1. Declaración de dos matrices de datos para almacenaje de las salidas de redes neuronales (desplazamiento y giros).
2. Declaración de dos variables, la primera corresponde a la aceleración (rad/s<sup>2</sup>) y la segunda es el ángulo de giro (rad).
3. Inclusión del código de redes neuronales al programa.
4. Cálculo de aceleración y ángulo de giro basada en los datos de salida de las redes neuronales.
5. Asignación de límites para aceleración y ángulo de giro, según la configuración física del robot móvil.
6. Cálculo y trazado de la trayectoria (FCAC).

La función de FCAC fue creada para calcular y trazar trayectorias basándose en el modelo de cinemática avanzado para un robot de móvil tipo triciclo. En esta función se especifica la correspondencia de nuestra aceleración y ángulo de giro:

$$aceleración = \frac{d\omega}{dt} \quad (1)$$

$$ángulo\ de\ giro = \gamma \quad (2)$$

La Figura 10 muestra el esquema representativo del robot móvil y sus parámetros calculados para trazar las trayectorias. El detalle completo de la interpretación física y la derivación de las fórmulas fue propuesto por el Dr. Antonio Benito Martínez Velasco.

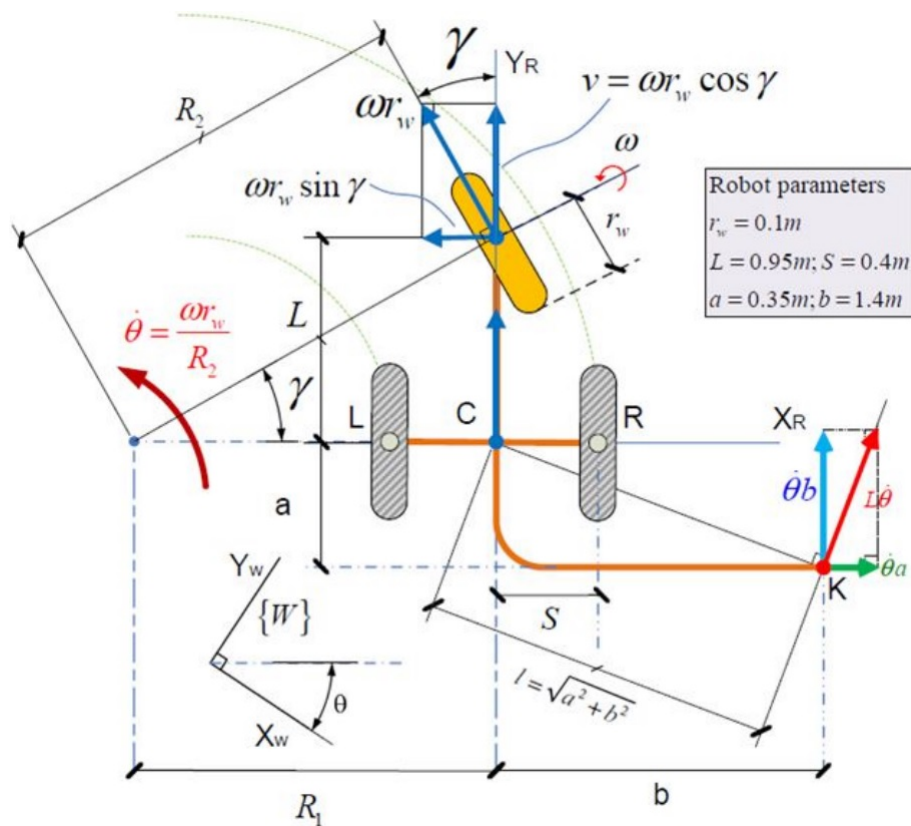


FIGURA 10  
Modelo cinemático directo de un robot móvil tipo triciclo.

#### 4. RESULTADOS

El funcionamiento de cada red neuronal puede ser analizado considerando su matriz de confusión Figura 11. Como se evidencia en la matriz de confusión del desplazamiento, todos los elementos fueron clasificados sobre la clase correcta, obteniendo el 100 % de funcionamiento. Además, el funcionamiento de la red neuronal para los giros es el 99,7 % debido a un elemento de la clase 3 que fue erróneamente clasificado en la clase 2.

Dos tipos de pruebas fueron realizadas con el robot real, las primeras para ajustar algunos parámetros como el tiempo de desplazamiento y los ángulos de giros; mientras que, las segundas consistieron en la evaluación del comportamiento del robot móvil ante las órdenes enviadas por los movimientos del teléfono inteligente. Para evidenciar las pruebas mencionadas, se graficó las trayectorias 30 veces cada uno de los movimientos posibles (desplazamiento y giros), es decir, 30 veces para aceleración (+), 30 veces para aceleración (-), 30 veces giro a la izquierda, 30 veces giro a la derecha, obteniendo el comportamiento esperado en el 100 % de casos Figura 12.

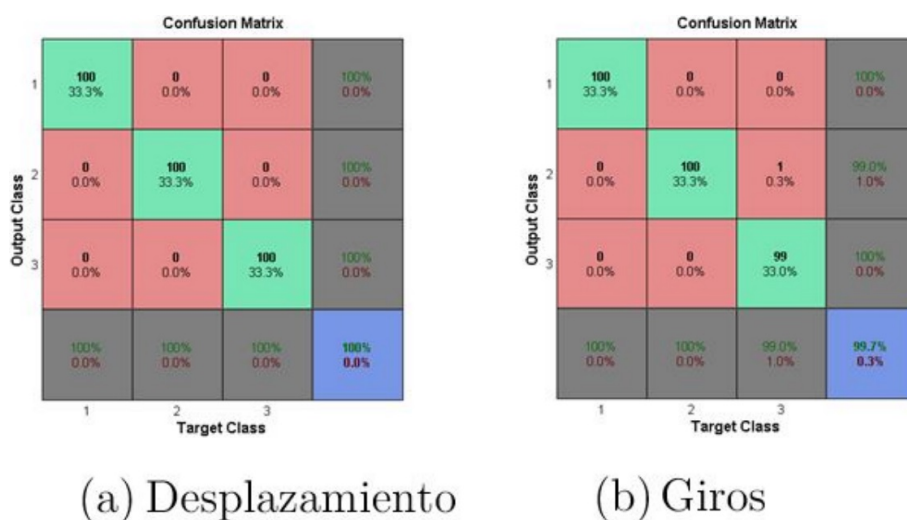


FIGURA 11  
Matrices de confusión de las redes neuronales.

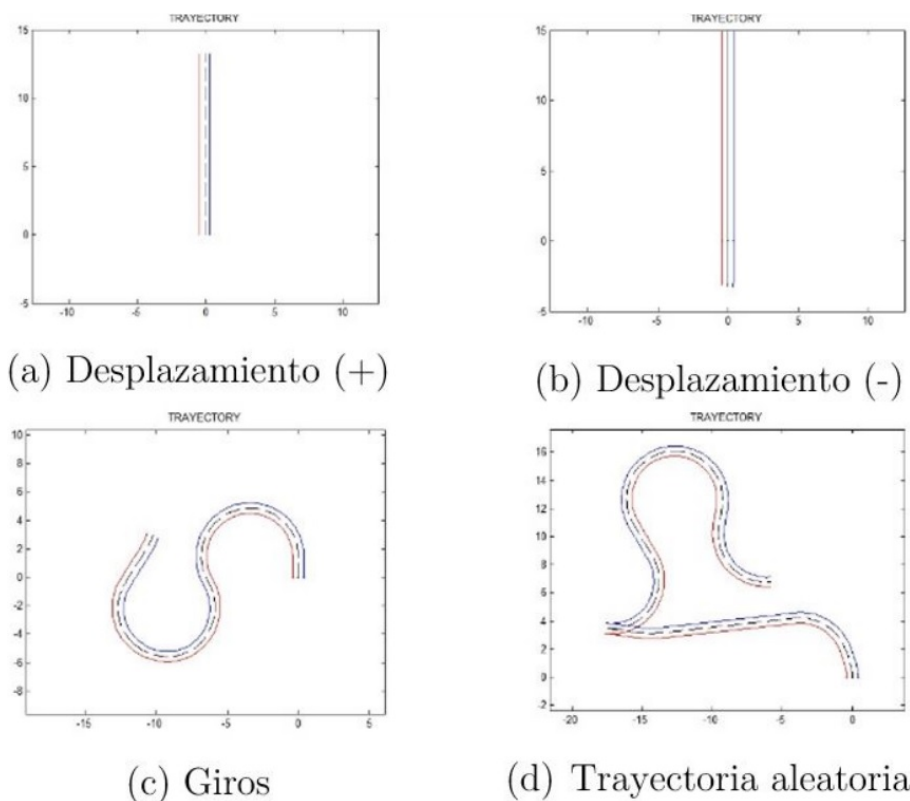


FIGURA 12  
Resultados de las pruebas con el robot real.

## 5. CONCLUSIONES

En este artículo se presentó una propuesta de sistemas remotos para robótica móvil usando la tecnología disponible de bajo costo como teléfonos inteligentes, considerando que la mayor parte de estos dispositivos cuenta con sensores que permiten una medición de parámetros físicos diferentes como la aceleración, la posición, ubicación, entre otros. Adicionalmente, el uso de dos redes neuronales permite clasificar los datos

recibidos para controlar y trazar la trayectoria de un robot móvil usando como mando a distancia los movimientos de un teléfono inteligente.

El manejo del robot mediante el uso del teléfono inteligente permite movimientos intuitivos, lo cual proporciona al usuario facilidad de manejo.

La implementación de una red neuronal evita usar equipo electrónico adicional para el acondicionamiento de señales como filtros y otros circuitos auxiliares. Por la simplicidad de las redes propuestas se podría realizar la implementación de estas en sistemas embebidos como *Raspberry Pi*, Arduino, entre otros como parte del robot móvil.

## AGRADECIMIENTOS

Los autores deseamos agradecer a Cristian M. Schmid por permitir usar su aplicación y modificar el código para el ordenador.

## REFERENCIAS

- [1] J. Cui, S. Tosunoglu, R. Roberts, C. Moore, and D. W. Repperger, "A review of teleoperation system control," Proceedings of the Florida Conference on Recent Advances in Robotics (FCRAR), pp. 1–12, 2003.
- [2] P. Batsomboon, S. Tosunoglu, and D. W. Repperger, "Development of a mechatronic system: a telesensation system for training and teleoperation," Chapter, Recent Advances in Mechatronics, Springer-Verlag, New York, pp. 304–321, 2000.
- [3] E. Slawiński, V. A. Mut, and J. F. Postigo, "Teleoperation of mobile robots with time-varying delay," IEEE Transactions on Robotics, vol. 23, no. 5, pp. 1071–1082, 2007.
- [4] Y. Kimitsuka, K. Sawai, and T. Suzuki, "Development of mobile robot teleoperation system utilizing robot sensor network," Proceedings of INSS 2008 - 5th International Conference on Networked Sensing Systems, vol. 23, no. 4, p. 250, 2008.
- [5] Uribe, S. Alves, J. M. Rosario, H. F. Filho, and B. Pérez-Gutiérrez, "Mobile robotic teleoperation using gesture-based human interfaces," 2011 IEEE 9th Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, LARC 2011 - Conference Proceedings, 2011.
- [6] H. Surmann, D. Holz, S. Blumenthal, T. Linder, P. Molitor, and V. Tretyakov, "Teleoperated Visual Inspection and Surveillance with Unmanned Ground and Aerial Vehicles," International Journal of Online Engineering (iJOE), vol. 4, pp. 1–13, 2008.
- [7] P. Vogt and J. Kuhn, "Analyzing free fall with a smartphone acceleration sensor," The Physics Teacher, vol. 50, no. 3, pp. 182–183, 2012.
- [8] Samà, C. Angulo, D. Pardo, A. Català, and J. Cabestany, "Analyzing human gait and posture by combining feature selection and kernel methods," Neurocomputing, vol. 74, no. 16, pp. 2665–2674, 2011.
- [9] G. Hache, E. D. Lemaire, and N. Baddour, "Mobility change-of-state detection using a smartphonebased approach," 2010 IEEE International Workshop on Medical Measurements and Applications, MeMeA 2010 - Proceedings, pp. 43–46, 2010.
- [10] M. Derawi, "Accelerometer-Based Gait Analysis, A survey Mohammad Omar Derawi Accelerometer Based Gait Analysis," Norwegian Information Security Conference, no. Mv, pp. 33–45, 2010.
- [11] C. Schmid. (2012) Mlconnect. [Online]. Available: <http://mlconnect.chschmid.com/index.php/> Main Page, 2012.
- [12] H. Demuth, Neural Network Toolbox™ 6 User's Guide, 2000, vol. 9, no. 4.
- [13] J. C. Moctezuma, "Neural network toolbox de matlab," in Ciencias computacionales, septiembre del 2006.