

# Robot con detección de obstáculos usando una línea láser y procesamiento de imágenes

J. I. Vega Luna<sup>1</sup>, J. F. Cosme Aceves<sup>1</sup>, G. Salgado Guzmán<sup>1</sup>, V. N. Tapia Vargas<sup>1</sup>, F. J. Sánchez Rangel<sup>1</sup>

<sup>1</sup>Área de Sistemas Digitales, Departamento de Electrónica, Universidad Autónoma Metropolitana-Azcapotzalco, Av. San Pablo 180, Col. Reynosa, C.P. 02200, Ciudad de México, México

[\\*vji@azc.uam.mx](mailto:*vji@azc.uam.mx)

**Área de participación:** Ingeniería Eléctrica y Electrónica

## Resumen

Se presenta la construcción de un robot cuyo objetivo fue recorrer el perímetro interno de un centro de datos para capturar imágenes del piso. El robot cuenta con un sistema de navegación basado en una tarjeta Arty Z7-10, una cámara de video, un emisor de línea láser y dos sensores ultrasónicos. El funcionamiento está basado en el método de detección de obstáculos en 2D a partir de la línea láser en 1D usando los cambios de gradiente de la línea láser para detectar obstáculos. La programación se implantó en Python y bibliotecas de OpenCV. Las imágenes serán suministradas a otro módulo del sistema, para detectar agua y evitar inundaciones. Las pruebas realizadas indicaron que el tiempo de procesamiento fue 200 ns. A partir de los resultados logrados, el robot puede usarse en otro tipo de aplicaciones de tiempo real.

**Palabras clave:** Navegación, OpenCV, Python, robot.

## Abstract

*This paper presents the construction of a robot whose objective was to walk the internal perimeter of a data center to capture images of the floor. The robot has a navigation system based on an Arty Z7-10 card, a video camera, a laser line emitter and two ultrasonic sensors. The operation is based on the 2D obstacle detection method from the 1D laser line using the gradient changes of the laser line to detect obstacles. The programming was implemented in Python and OpenCV libraries. The images will be supplied to another module of the system, to detect water and prevent flooding. Tests carried out indicated that the processing time was 200 ns. Based on the results achieved, the robot can be used in other types of real-time applications.*

**Key words:** Navigation, OpenCV, Python, robot.

## Introducción

La operación de un centro de datos debe ser continua para proporcionar de manera segura y eficiente el acceso a equipos e información de los clientes y usuarios. Existen diferentes aspectos a considerar en la seguridad física de estos recursos. Una amenaza en la operación de un centro de datos son las inundaciones. Éstas pueden ser causadas por diferentes eventos como por ejemplo fenómenos ambientales, accidentes, sabotajes y daños o fugas en las tuberías de los sistemas de enfriamiento y ventanas defectuosas. La presencia de agua puede causar que los equipos alojados en el centro de datos detengan su operación o inclusive se dañen o pierdan la información almacenada en ellos. Comúnmente, los centros de datos no pueden enfrentar una inundación, ya que son instalaciones en crecimiento continuo y carecen de medidas pro-activas basadas en los últimos avances tecnológicos que permitan reaccionar oportunamente en caso de contingencias. En algunos de ellos se encuentran instalados sistemas con sensores de agua en ciertos lugares del mismo. Es difícil, o a veces imposible, cubrir toda el área del piso del centro de datos usando estos sistemas. Una alternativa para la detección de agua es usar un robot móvil que se desplace en el centro de datos y capture imágenes del piso para procesarlas, detectar agua y evitar inundaciones.

Los robots móviles surgen en la década de los 90. Este tipo de robots tiene la capacidad de movimiento sobre entornos no estructurados, de los que posee un conocimiento incierto, mediante la interpretación de la información suministrada a través de sensores y del estado o ubicación del vehículo. Se enfrentan con significativas incertidumbres en la posición e identificación de objetos. La incertidumbre para trasladarse de un punto a otro puede representar una actividad de riesgo para el robot [1]. A raíz de esto, los robots móviles no siguen rutas al destino final con el mismo nivel de precisión que otro tipo de robots como por ejemplo los industriales. Los índices de operación, medidos por la incertidumbre y precisión requeridas de los robots móviles, en relación a los robots industriales fijos, se deben a la existencia de un conjunto diferente de prioridades. Las prioridades en los robots móviles están orientadas a las áreas de sensado y raciocinio [2].

Uno de los principales problemas a resolver en un robot móvil es generar rutas para guiar su movimiento de acuerdo a la información proveniente del sistema de sensores externos. Esto permitirá desplazarse entre dos puntos del ambiente de trabajo de manera segura y sin colisiones. El diseño de rutas exige contar con sistemas de control de trayectorias en diversos niveles jerárquicos, de manera tal que el procesamiento de la información proveniente de los sensores asegure la mayor autonomía posible [3].

Este trabajo se realizó por requerimiento de una empresa operadora de centros de datos. Se solicitó realizar un robot autónomo móvil de bajo costo que realice el recorrido en el perímetro interior del centro de datos, así como capturar y almacenar periódicamente fotografías del piso. Debe considerarse que el piso falso del centro de datos no es uniforme, ya que existen escalones y rampas con desnivel en algunas áreas como por ejemplo en las puertas de acceso.

Con base en lo solicitado, la solución propuesta fue desarrollar un robot compacto de dos ruedas equipado con una cámara de video, un emisor de línea láser, dos sensores ultrasónicos y un módulo de navegación. El módulo de navegación usa el método de detección de obstáculos basado en los cambios de gradiente de la línea láser de las imágenes capturadas. El robot es controlado por una tarjeta Arty Z7-10, la cual fue seleccionada de un conjunto de tarjetas de desarrollo entre las cuales se encuentran la Arduino, TI LaunchPad y Pyboard por las siguientes razones: 1) La tarjeta Arty Z7-10 cuenta con los recursos de hardware necesarios para conectar una cámara de video para la captura de las imágenes; 2) El método de detección de obstáculos puede realizarse de forma eficiente utilizando el FPGA que integra la Arty Z7-10; 3) La programación del sistema puede llevarse a cabo en Python. Desde Python se puede usar una gran cantidad de bibliotecas de funciones y herramientas de desarrollo disponibles en la comunidad de software libre de código abierto de fácil instalación, configuración y uso; y 3) Escalabilidad, la tarjeta Arty Z7-10 permitirá en el futuro adicionar funciones al sistema para el módulo de detección de inundaciones.

Durante los últimos años se ha desarrollado una cantidad grande de investigaciones y trabajos de robótica, con características y funciones diversas que incorporan tecnologías recientes para resolver necesidades en muchos campos de la vida humana y procesos industriales. Particularmente, el desarrollo de robots de dos ruedas se ha enfocado en tres líneas: detección de obstáculos, algoritmos de navegación y aplicaciones. La detección de objetos y personas se realiza usando dos técnicas diferentes. La primera utiliza dispositivos electrónicos que incluyen sensores, cámaras de video o cámaras térmicas que activan un actuador o registran el evento. Algunos sensores comúnmente utilizados son los: pasivos infrarrojos (PIR-Passive Infrared), ultrasónicos, láser, emisores de señales de radio y los dispositivos de rango y detección de luz (LIDAR-Light Detection and Ranging). Una desventaja de este tipo de sensores es que no pueden detectar fácilmente objetos y superficies reflejantes ni desniveles en el piso. La segunda técnica combina el uso de un dispositivo, como los indicados anteriormente, y un algoritmo o método de visión por computadora [4].

Los robots que usan técnicas basadas en cámaras de video obtienen imágenes del entorno que pueden usarse para reconocer colores, texturas, movimiento y detectar obstáculos. No obstante, uno de los problemas que se presentan al usar las imágenes es que el algoritmo de detección

usado puede ser engañado por patrones específicos, ilusiones ópticas, escenas homogéneas o cambios en el entorno, como por ejemplo cambios de iluminación y movimientos [5].

Se han propuesto varios métodos para extraer información de la profundidad de una escena usando cámaras. Una de las técnicas más usadas por su facilidad de implantar y porque no requiere gran cantidad de recursos computacionales es la triangulación láser. Esta técnica de visión utiliza un haz de luz o línea láser y una cámara dirigidos a un objeto o superficie. Considerando la diferencia de ángulos entre el haz de laser que incide en objeto y el que incide en el sensor de la cámara se puede obtener la profundidad usando relaciones trigonométricas. La distancia vertical entre el láser y el objeto producirá un cambio proporcional en la posición en el sensor. Una variación de esta técnica es usar el gradiente de la luz láser reflejada. Esto es, cuando la línea láser impacta en un objeto o una protuberancia existe un cambio en el gradiente y posición de la línea, lo cual permite detectar objetos y desniveles en el piso y, si es necesario, la distancia al punto de impacto. Se trata de distinguir en la imagen la línea láser usando el gradiente y posición de la línea. Esta fue la técnica usada en este trabajo, para la cual es conveniente usar una línea láser de color rojo con el fin de generar gradientes altos o cambios grandes de intensidad de luz.

La navegación de robots y vehículos guiados automáticamente (AGV-Automatic Guided Vehicles) en ambientes con obstáculos es aún un reto. Las investigaciones realizadas los últimos años en lo referente al desarrollo de algoritmos y métodos para detección y evasión de obstáculos se han concentrado en el desarrollo de técnicas para resolver el mapeo, localización y planeación de rutas en la navegación [6]. Como resultado, se han realizado diversos trabajos cuyo resultado ha sido la creación de robots utilizados en diferentes aplicaciones cuyos sistemas se basan en las siguientes tecnologías: ultrasonido para guiar el robot en aplicaciones clínicas que usan un catéter venoso central [7]; endoscopia para navegar en el tracto humano gastrointestinal y detectar y destruir anomalías y tejidos malignos [8]; redes de sensores inalámbricos con medidores de rango para micro robots voladores usados la Internet de las Cosas Industriales (Industrial Internet of things-IIoT) [9]; algoritmos de navegación utilizando paradigmas de control cinemático [10]; grupos o clusters de robots para navegación cooperativa en ambientes desconocidos y transporte de objetos usando lógica difusa [11]; localización y navegación en interiores basado en redes de petri y tecnología de identificación por radio frecuencia (RFID) [12]; dinámica no lineal aplicada en irregularidades del suelo en la agricultura [13]; planeación de movimiento para navegación punto a punto en superficies irregulares de robots esféricos [14]; computación en la nube y procesamiento de imágenes estereoscópicas para navegación remota asistida [15]; navegación basada en imágenes obtenidas con fluoroscopia para cirugías de espina dorsal [16]; navegación basada en aprendizaje con redes neuronales [17]; ajuste de flujo óptico y visión por computadora para la estimación de distancia en robots voladores usados en ambientes interiores y exteriores [18]; redes de sensores para determinar la localización y trazar mapas de la topología de la ruta a seguir [19], navegación asistida por GPS y uso de ondas de radio de VHF [20]; reconocimiento de imágenes y redes neurales profundas [21] y navegación inercial submarina basada en giroscopios y la velocidad Doppler [22].

Las ventajas y aportaciones de este trabajo son las siguientes: 1) El uso del robot no es intrusivo, ya que no se modificó el piso ni la infraestructura del centro de datos; 2) No existe actualmente una solución comercial para detección de inundaciones para un centro de datos. Una vez procesadas las imágenes, por otro módulo que no es parte de este trabajo, se podrá detectar agua en el piso y evitar inundaciones. Esto resuelve una necesidad real, dando oportunidad para tomar medidas adecuadas antes de una emergencia y 3) La tecnología usada es de reciente creación, las herramientas software usadas son de código abierto lo cual redujo la complejidad y costo del sistema.

## **Metodología**

El robot construido está compuesto por dos módulos: el de navegación y el control de movimiento.

### **El módulo de navegación**

La superficie del centro de datos es de 30 x 30 metros y el robot se diseñó para desplazarse en el perímetro interno en sentido de las manecillas del reloj como se indica en la Figura 1.

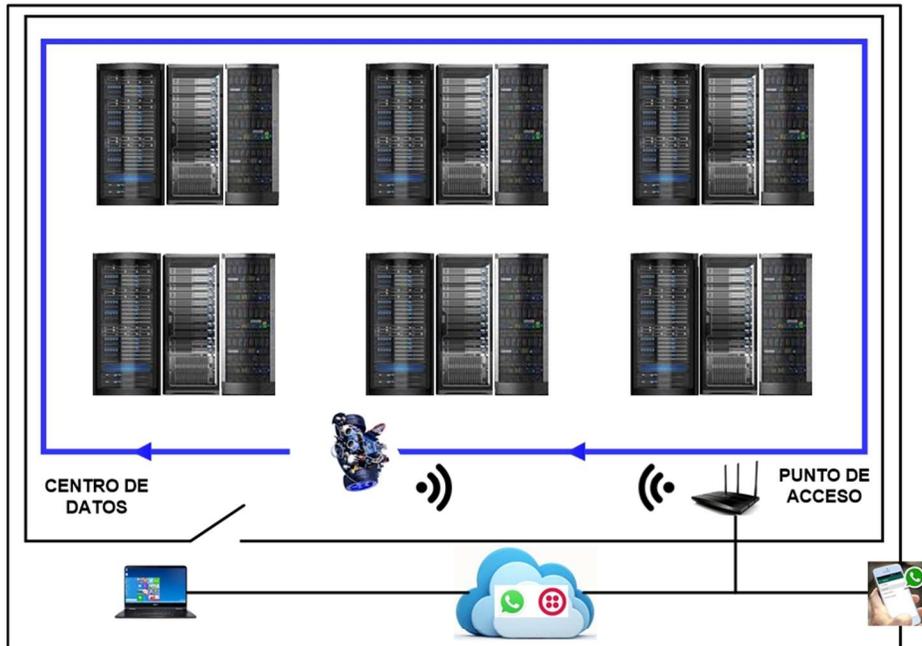


Figura 1. Desplazamiento de robot en el perímetro interno.

En la Figura 2 se indican los componentes del robot. La cámara y el emisor de línea láser se instalaron al frente y los sensores ultrasónicos a los costados del robot. La tarjeta Arty Z7-10 está basada en SoC FPGA XC7Z010-1CLG400C el cual integra un procesador de dos núcleos ARM Cortex-A9. Esta tarjeta cuenta con los siguientes recursos: memoria DDR3 de 512 MB, memoria flash de 16 MB, puerto Gigabit Ethernet, tres puertos USB, un puerto HDMI, push-buttons y leds.

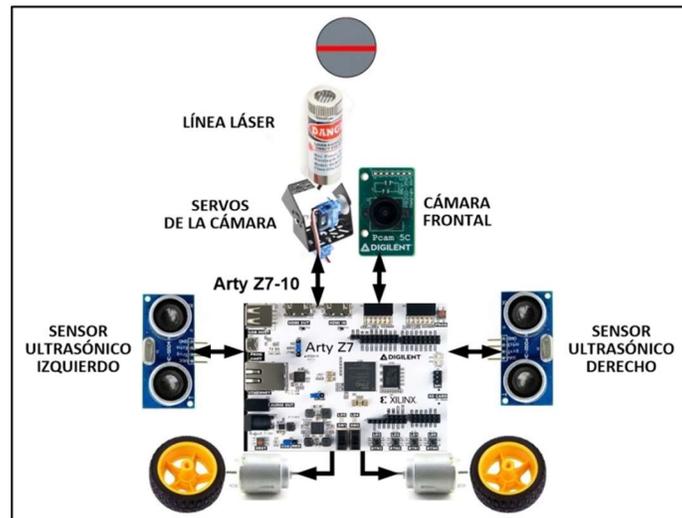


Figura 2. Componentes del robot.

La programación de la Arty Z7-10 se encarga de realizar las siguientes tareas: A) Capturar periódicamente las imágenes del entorno y B) Controlar el movimiento del robot para mantener la ruta de navegación al interior del centro de datos. La programación de la Arty Z7-10 se realizó en Python y funciones de la biblioteca de acceso libre OpenCV.

La cámara de video usada es el dispositivo Logitech C920. Esta cámara de tamaño compacto tiene resolución máxima de 1920x1080 pixeles y 30 cuadros por segundo, enfoque automático y proporciona imágenes en formato H.264 y M-JPEG a través de la salida USB 2.0. La cámara se conectó a un puerto USB de la tarjeta Arty Z7-10 y se instaló en una estructura de movimiento tipo sombrero Pimoroni Pan-Tilt. La estructura integra dos servomotores para el movimiento horizontal panorámico en el eje X, o pan, y para el movimiento vertical en el eje Y, inclinación o tilt. En la estructura Pimoroni se fijó el módulo con diodo láser de línea roja horizontal clase IIa de 5 mW y 650 nm. La alimentación del láser es de 3 a 5 VDC, consume 40 mA máximo y su alcance es 1 metro.

Los servos se mueven en el rango de 180 grados, de -90° a 90°, se alimentan con 5 VDC, trabajan a una frecuencia de 50 Hz, el ancho del pulso de control es de 1 a 2 ms y se mueven a una velocidad de 230 ms cada 60 grados. El control de movimiento del servomotor se realizó usando la biblioteca de funciones de código libre *pantiltthat* y mediante la función *pantiltthat.pan()* se indica el ángulo o posición en grados. El servomotor se mueve constantemente a lo largo del eje X de forma horizontal, capturando imágenes del entorno cada 30°, cada 115 ms.

El algoritmo usado por el módulo de navegación se basa en la detección de obstáculos a los costados y al frente. Para el primer caso, el robot usa los sensores ultrasónicos, o de ultrasonidos. Los sensores ultrasónicos utilizados en este módulo son los dispositivos HC-SR04. Estos sensores ofrecen buena relación costo/beneficio, ya que son económicos, compactos, fácil de usar y su alcance es de 3 centímetros a 3 metros, con una resolución de 0.3 centímetros. Cada sensor se conectó a dos terminales GPIO de la tarjeta Arty Z7-10. La entrada de disparo del sensor se conectó a una terminal de salida GPIO para generar un pulso de 10  $\mu$ s y activarlo. La salida del sensor se conectó a una terminal de entrada GPIO para determinar la longitud del pulso de retorno y la distancia del objeto. En caso de no detectar objetos, la longitud del pulso de eco es 36 ms.

La detección de obstáculos al frente y desniveles en el piso se realiza determinando si existen cambios en el gradiente de la línea proyectada por el emisor láser en las imágenes frontales capturadas. Inicialmente se determinó el valor del gradiente máximo de la línea laser en condiciones de reposo, sin obstáculos y con el piso uniforme. Este valor del gradiente en reposo es usado como referencia en la operación del módulo y se obtuvo usando las siguientes fases y funciones de OpenCV: 1) Convertir la imagen a formato YUV; 2) Obtener el componente Y; 3) Aplicación de un filtro Gaussiano para difuminar; 4) Aplicación de un filtro Sobel; y 5) Determinar el gradiente máximo. El algoritmo usa el formato YUV, comúnmente usado en procesamiento de imágenes a color, ya que utiliza tres componentes para representar la imagen: la Y indica la luminosidad, intensidad o información de blanco y negro, y la U y V, que representan la crominancia o información del color. Una forma para determinar cambios en la imagen filtrada es usar derivativas, ya que un cambio alto en el gradiente indica un cambio mayor en la imagen. Para realizar esto, es necesario ubicar los pixeles cuyo gradiente es más mayor que el de sus vecinos. El operador Sobel determina el gradiente de una imagen basado en los cambios de intensidad, o derivativas, horizontal y vertical. Este operador genera una imagen resaltando los bordes detectados y un fondo negro.

Una vez que inicia el recorrido el robot, el programa captura imágenes cada 500 ms. Cada imagen es procesada a través de las cinco fases anteriores. En la última etapa se obtiene el gradiente vertical. Si existe un cambio entre el valor de referencia del gradiente de la línea laser y el obtenido en esta etapa, se asume que se ha detectado un obstáculo o desnivel del piso, el robot se detiene y decide el siguiente movimiento. El algoritmo de módulo de navegación toma como base que el robot se mueve manteniendo a su izquierda la pared y consiste de cinco funciones: *adelante*, *atrás*, *paro*, *giro\_derecho* y *giro\_izquierdo*. El robot navega a una velocidad de 10 cm/s y se estableció un límite mínimo de 15 cm al que puede estar alejado de una pared u objeto y un límite máximo de 25 cm al que puede estar alejado de la pared izquierda. Es decir, la distancia al obstáculo no debe ser igual o menor a 15 cm. Si el robot detecta un obstáculo al frente a una distancia igual o menor al límite mínimo gira a la derecha 90 grados. El algoritmo activa cada segundo los sensores laterales

para determinar si existe un obstáculo y la distancia al mismo. Si se detecta un obstáculo en los costados a una distancia igual o menor al límite mínimo el robot gira 180 grados. Si se detecta un obstáculo a la izquierda a una distancia menor al límite mínimo, se asume que el robot tiende a moverse y acercarse a la izquierda, entrando a un ciclo en el que invoca las funciones *giro\_derecho*, adelante y *giro\_izquierdo* hasta corregir la ruta. Si la distancia es igual o mayor al límite máximo se asume que el robot tiende a moverse a la derecha y entra a un ciclo en el que invoca las funciones *giro\_izquierdo*, adelante y *giro\_derecho* para corregir la ruta. Si el robot no se encuentra en ninguno de los casos anteriores avanza hacia adelante.

### El módulo de control de movimiento

La programación de las cinco funciones de movimiento del robot controla la activación de dos moto-reductores de 6 V y 600 rpm usados para girar las ruedas. Los motores se controlan mediante un Puente H conectado a cuatro terminales de salida GPIO de la tarjeta Arty Z7-10. En la Tabla 1 se indican los niveles lógicos utilizados en las terminales de salida para activar los motores según el movimiento a realizar. En la función *adelante*, los valores de ambos motores, se mantienen constantes. Para corregir la ruta, se detiene un motor y se hace avanzar el opuesto para retomar la ruta correcta en función de los sensores ultrasónicos. Para *giro\_derecho*, se hace avanzar el motor izquierdo y retroceder el derecho durante 100 microsegundos. Para *giro\_izquierdo*, se hace avanzar el motor derecho y retroceder el izquierdo durante 100 microsegundos. Para giro de 180° en la función *atrás*, se repite el proceso anterior durante 200 microsegundos.

**Tabla 1. Niveles lógicos de las terminales de salida GPIO que controlan las ruedas del robot.**

Función	Nivel de las terminales para el motor derecho		Nivel de las terminales para el motor izquierdo	
adelante	0	1	0	1
atrás	1	0	1	0
paro	0	0	0	0
giro_derecho	0	1	1	0
giro_izquierdo	1	0	0	1

### Resultados y discusión

Se realizaron dos grupos de pruebas. El primer conjunto tuvo como objetivo determinar las distancias máxima y mínima a que el robot puede detectar un objeto o desnivel del piso. En algunas de estas pruebas se colocaron diversos objetos frente al robot y en otras se ubicó al robot cercano a desniveles. A pesar de que nominalmente el láser tiene un alcance de un metro, las pruebas indicaron que el emisor debe colocarse con una inclinación de 10° hacia el piso para detectar obstáculos situados a una distancia máxima de 85 centímetros. Con una inclinación menor disminuye el alcance del láser y difícilmente la cámara detecta la línea laser reflejada. La distancia mínima que se logró fue 2 centímetros. Este último valor se debe a dos razones: 1) El tiempo de procesamiento del algoritmo de detección son 200 ms y 2) El robot se mueve hacia adelante a una velocidad de 10 centímetros por segundo, de manera que al terminar la ejecución el algoritmo, el robot avanza 2 centímetros y volverá a ejecutarse el algoritmo antes de avanzar. Esto implica que existe un compromiso entre la velocidad del robot y la distancia mínima al objeto. La distancia es inversamente proporcional a la velocidad. Considerando lo anterior y que el robot captura una imagen antes de ejecutar el algoritmo, la captura de imágenes es cada 200 ms y el algoritmo usa 5 de los 30 cuadros por segundo que proporciona la cámara.

El segundo grupo de pruebas tuvo como objetivo determinar la distancia máxima de detección con los sensores ultrasónicos. El alcance nominal de los sensores es 3 metros, sin embargo las pruebas realizadas mostraron que el alcance real es 2.45 metros. Este resultado es aceptable tomando en cuenta que el robot se mueve a una distancia máxima de 15 centímetros de la pared

izquierda. Pudo haberse usado el método de detección basado en analizar la segmentación de la línea laser al impactar en un obstáculo, sin embargo, la implantación del método de cambio de gradiente resultó relativamente sencilla de implantar y eficiente.

## Trabajo a futuro

Existen dos líneas de trabajo a realizar en un futuro inmediato. La primera es realizar la programación para procesar y reconocer las imágenes capturadas y detectar la presencia de agua con el objetivo de evitar inundaciones en el centro de datos. Comúnmente, los algoritmos de reconocimiento demandan mayor cantidad de recursos computacionales que los de detección de obstáculos. La implantación del algoritmo a usar se llevará a cabo usando el FPGA que integra la tarjeta Arty Z7-10. La segunda línea es incorporar un mecanismo de control remoto de navegación, mediante el cual pueda indicarse manual o automáticamente la ruta a seguir para el robot pueda moverse en los pasillos del centro de datos o en cualquier lugar.

## Conclusiones

Se obtuvo un robot de dos ruedas que se desplaza en el perímetro interior de un centro de datos capturando imágenes cada 200 ms las cuales servirán para detectar agua en el piso y evitar inundaciones. Con base en los resultados logrados, se considera que el tiempo de procesamiento del algoritmo de detección de objetos es aceptable ya que se estima que el algoritmo de detección y reconocimiento de agua o herramienta, como por ejemplo TensorFlow, consume 250 ms. Esto significa que cada 450 ms el sistema podrá indicar si ha detectado agua y mover el robot a una distancia más cercana al punto donde se encuentra el agua. El robot avanza a una velocidad de 10 centímetros por segundo. Si es necesario que me mueva más rápidamente, deben realizarse cambios mínimos en la programación contemplando que entre mayor sea la velocidad disminuye la distancia a la que se puede detectar un obstáculo.

## Referencias

- [1] M. Ferro, A. Paolillo, A. Cherubini and M. Vendittelli, "Vision-Based Navigation of Omnidirectional Mobile Robots", *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2691-2698, 2019.
- [2] X. Truong and T. D. Ngo, "Toward Socially Aware Robot Navigation in Dynamic and Crowded Environments: A Proactive Social Motion Model", *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1743-1760, 2017.
- [3] A. G. C. Gonzalez, M. V. S. Alves, G. S. Viana, L. K. Carvalho and J. C. Basilio, "Supervisory Control-Based Navigation Architecture: A New Framework for Autonomous Robots in Industry 4.0 Environments", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1732-1743, 2018.
- [4] D. D. Uyeh et al., "Evolutionary Greenhouse Layout Optimization for Rapid and Safe Robot Navigation", *IEEE Access*, vol. 7, pp. 88472-88480, 2019.
- [5] C. Luo, S. X. Yang, X. Li and M. Q. Meng, "Neural-Dynamics-Driven Complete Area Coverage Navigation Through Cooperation of Multiple Mobile Robots", *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 750-760, 2017.
- [6] P. J. Costa, N. Moreira, D. Campos, J. Gonçalves, J. Lima and P. L. Costa, "Localization and Navigation of an Omnidirectional Mobile Robot: The Robot@Factory Case Study", *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 11, no. 1, pp. 1-9, Feb. 2016.
- [7] D. Wang et al., "Efficient multi-parameter calibration method for CVC assist robot with servo-navigation system", *Journal of Engineering*, vol. 2019, no. 14, pp. 536-542, 2019.
- [8] Y. Guo, S. Zhang, A. Ritter and H. Man, "A Case Study on a Capsule Robot in the Gastrointestinal Tract to Teach Robot Programming and Navigation", *IEEE Transactions on Education*, vol. 57, no. 2, pp. 112-121, 2014.

- [9] H. Li and A. V. Savkin, "Wireless Sensor Network Based Navigation of Micro Flying Robots in the Industrial Internet of Things", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3524-3533, 2018.
- [10] A. S. Matveev, M. C. Hoy and A. V. Savkin, "Extremum Seeking Navigation Without Derivative Estimation of a Mobile Robot in a Dynamic Environmental Field", *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1084-1091, 2016.
- [11] C. Juang, M. Lai and W. Zeng, "Evolutionary Fuzzy Control and Navigation for Two Wheeled Robots Cooperatively Carrying an Object in Unknown Environments", *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1731-1743, 2015.
- [12] F. A. X. Da Mota, M. X. Rocha, J. J. P. C. Rodrigues, V. H. C. De Albuquerque and A. R. De Alexandria, "Localization and Navigation for Autonomous Mobile Robots Using Petri Nets in Indoor Environments", *IEEE Access*, vol. 6, pp. 31665-31676, 2018.
- [13] X. Gao et al., "Review of Wheeled Mobile Robots' Navigation Problems and Application Prospects in Agriculture", *IEEE Access*, vol. 6, pp. 49248-49268, 2018.
- [14] V. Dwaracherla, S. Thakar, L. Vachhani, A. Gupta, A. Yadav and S. Modi, "Motion Planning for Point-to-Point Navigation of Spherical Robot Using Position Feedback", *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2416-2426, 2019.
- [15] J. Salmerón-García, P. Íñigo-Blasco, F. Díaz-del-Río and D. Cagigas-Muñiz, "A Tradeoff Analysis of a Cloud-Based Robot Navigation Assistant Using Stereo Image Processing", *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 444-454, 2015.
- [16] S. Li, Z. Du and H. Yu, "A Robot-Assisted Spine Surgery System Based on Intraoperative 2D Fluoroscopy Navigation", *IEEE Access*, vol. 8, pp. 51786-51802, 2020.
- [17] E. A. Antonelo and B. Schrauwen, "On Learning Navigation Behaviors for Small Mobile Robots With Reservoir Computing Architectures", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 763-780, 2015.
- [18] A. Aminzadeh and M. A. A. Atashgah, "Implementation and performance evaluation of optical flow navigation system under specific conditions for a flying robot", *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 11, pp. 20-28, 2018.
- [19] A. Gunathillake, H. Huang and A. V. Savkin, "Sensor-Network-Based Navigation of a Mobile Robot for Extremum Seeking Using a Topology Map", *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3962-3972, 2019.
- [20] D. Reyes, G. Millan, R. Osorio-Corparan and G. Lefranc, "Mobile Robot Navigation Assisted by GPS", *IEEE Latin America Transactions*, vol. 13, no. 6, pp. 1915-1920, 2015.
- [21] J. Bai, S. Lian, Z. Liu, K. Wang and D. Liu, "Deep Learning Based Robot for Automatically Picking Up Garbage on the Grass", *IEEE Transactions on Consumer Electronics*, vol. 64, no. 3, pp. 382-389, 2018.
- [22] D. Wang, X. Xu, Y. Yao, Y. Zhu and J. Tong, "A Hybrid Approach Based on Improved AR Model and MAA for INS/DVL Integrated Navigation Systems", *IEEE Access*, vol. 7, pp. 82794-82808, 2019.