

Plataforma para testeo de algoritmos de escape implementados en hardware

Franco Dinter, M. Celeste Cebedio, Luciana De Micco y Lucas A. Rabioglio
ICYTE, Depto. de Electrónica y Computación, Facultad de Ingeniería - UNMDP
Mar del Plata, 7600, Argentina
francopiasensa@gmail.com, {celestecebedio,lucas.rabioglio,ldemicco}@fi.mdp.edu.ar

Resumen—En este trabajo se presenta la implementación de un robot diseñado para resolver laberintos, junto con un laberinto de prueba adecuado para evaluar diversos algoritmos de escape implementados en una FPGA. Este entorno experimental permite probar y comparar la eficiencia de distintos enfoques de navegación y optimización en tiempo real.

Palabras clave—FPGA, Laberinto, Robot, Algoritmo de Escape.

I. INTRODUCCIÓN

La resolución de laberintos es un problema clásico en robótica y teoría de grafos, en el cual el desafío radica en determinar una ruta eficiente para que un robot móvil se desplace desde su posición inicial hasta un destino específico dentro de un entorno complejo y lleno de obstáculos. Este entorno presenta múltiples rutas posibles, y el objetivo es que el robot identifique y siga la mejor ruta disponible. Para lograrlo, el robot no solo debe evitar colisiones con los obstáculos que limitan sus movimientos, sino también seleccionar una trayectoria óptima que minimice la distancia recorrida, el tiempo de desplazamiento, o ambos, según los criterios de optimización definidos. Además, en escenarios dinámicos, donde los obstáculos pueden moverse o aparecer de manera impredecible, el robot debe adaptarse en tiempo real, ajustando su ruta para asegurar su llegada al destino de forma segura y eficiente. Este problema tiene aplicaciones relevantes en sistemas de navegación autónoma, logística en almacenes, y vehículos autónomos, entre otros campos. Existen diversas propuestas para abordar este problema, [1], [2], en las cuales los algoritmos son implementados en microcontroladores. En este trabajo, se propone testear distintos algoritmos de escape [3] en un dispositivo FPGA con el fin de aprovechar la capacidad de paralelismo que ofrece dicho hardware, maximizando así la eficiencia y velocidad de los algoritmos. Para validar esta propuesta, se desarrolla un robot equipado con los sensores necesarios y un laberinto de prueba diseñado específicamente para evaluar su desempeño.

II. ENTORNO DE PRUEBAS

II-A. Robot

El robot implementado se muestra en la Fig. 1. Este dispositivo está equipado con dos motores de corriente continua que permiten su movimiento, alimentados mediante un puente H. Además, el robot cuenta con un sensor frontal digital, un sensor digital para la detección de línea, y dos sensores laterales infrarrojos analógicos, tal como se ilustra en la figura.

Identify applicable funding agency here. If none, delete this.

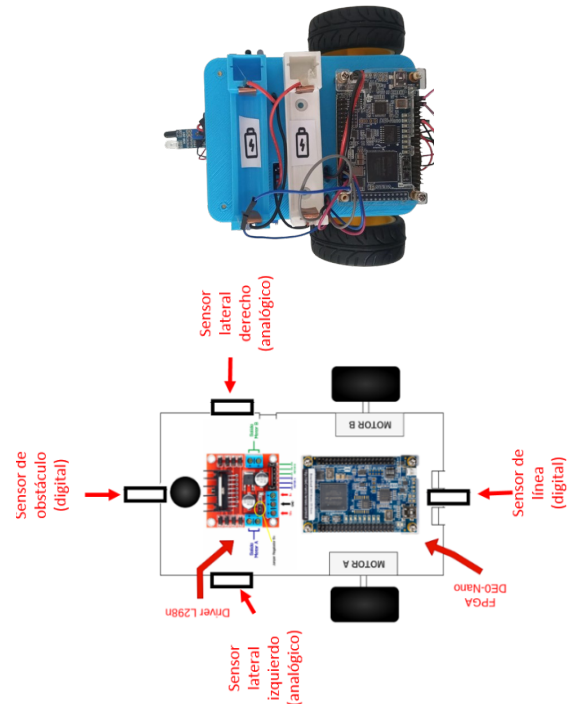


Figura 1. Robot implementado.

El control y procesamiento de los datos de los sensores, así como el accionamiento de los motores, se gestionan en una DEO-Nano Field Programmable Gate Arrays (FPGAs) de Altera [4]. En esta FPGA se implementa el algoritmo de navegación, el cual regula los movimientos del robot en función de las lecturas de los sensores. Los sensores infrarrojos laterales están conectados a dos canales de un convertidor Analógico-Digital (ADC) integrado en la placa de desarrollo. Este ADC es gestionado por la FPGA para recibir en tiempo real la información de distancia de los laterales del robot, permitiendo así un desplazamiento recto y estable.

II-B. Laberinto

El laberinto implementado, que se muestra en la Fig. 2, está construido en madera, con columnas fabricadas con impresora 3D y diseñado con cuadrículas delimitadas por líneas negras, que el robot puede detectar fácilmente. Además, cada cuadrícula está configurada para permitir la incorporación de muros, lo que posibilita ajustar y variar la complejidad del laberinto para diferentes pruebas y escenarios de navegación.



Figura 2. Laberinto implementado con paredes rebatibles.

III. CONTROL DEL ROBOT

La Fig. 3 muestra un esquema del control implementado en la FPGA, destacando las funcionalidades básicas necesarias para el desplazamiento del robot: detección de obstáculos y giro, desplazamiento centrado, control de motores, y conteo de los casilleros atravesados. En el bloque de Control General se implementan diversos algoritmos de escape, desde métodos básicos, como el giro aleatorio y el seguimiento de paredes laterales, hasta algoritmos más avanzados, como el de llenado. Esta estructura permite evaluar y comparar el rendimiento de distintas estrategias de navegación en el laberinto.

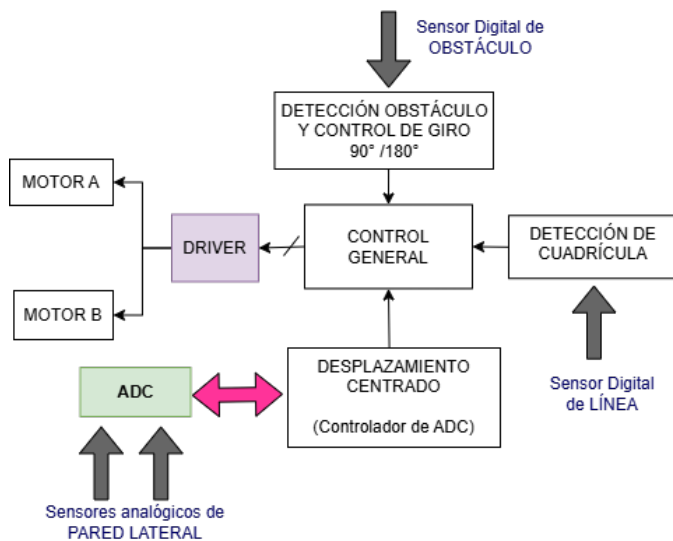


Figura 3. Esquema del control implementado.

- Desplazamiento centrado El robot debe ser capaz de avanzar en línea recta, para ello cuenta con los sensores laterales. Se deberá tomar la señal de estos sensores (entradas analógicas conectadas al ADC de la placa), y en caso de que el valor entregado por los sensores sea menor a cierto umbral accionar sobre las ruedas a través del motor para alejar al robot de la pared u obstáculo.
- Detección de obstáculo y giro El robot debe ser capaz de realizar giros de 90° y 180° cuando detecte un muro en frente, para ello se deberá emplear el sensor de frente, y cuando detecte el muro accionar las ruedas a través de

los motores y con un contador lograr que gire el ángulo deseado.

- Detección de cuadrícula El robot debe ser capaz de “saber” en qué cuadrícula se encuentra. Para ello debe ser capaz de detectar cuando el sensor de línea (conectado debajo del robot) detecta la línea y actualizar algún mecanismo de guardado.
- Control General. Esto incluye Lógica de decisión. El robot debe ser capaz de salir del laberinto, para ello se debe decidir qué estrategia emplear.

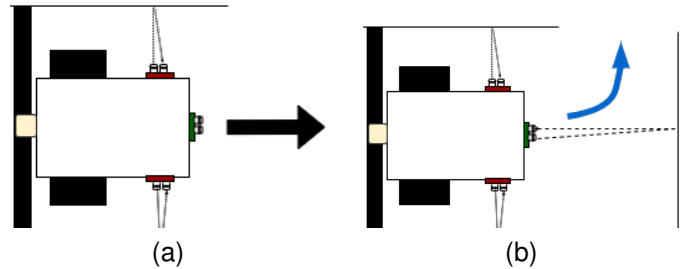


Figura 4. Funcionalidades: a) Desplazamiento centrado b) Detección de obstáculos y giro.

IV. CONCLUSIONES Y TRABAJO FUTURO

Se ha logrado integrar exitosamente los sensores y motores del robot, controlados mediante una FPGA, permitiendo una respuesta en tiempo real que es fundamental para la navegación en el laberinto. Esta base sólida posibilita la implementación de algoritmos de escape, los cuales comenzarán con métodos simples como el giro aleatorio y el seguimiento de paredes, y se extenderán a estrategias más avanzadas, como el algoritmo de llenado. A medida que se implementen, estos algoritmos se evaluarán en distintas configuraciones del laberinto para comparar su eficiencia en términos de tiempo de navegación y uso de recursos en la FPGA. Posteriormente, se buscará optimizar los algoritmos para explotar mejor el paralelismo del hardware, lo que aumentará la velocidad y eficiencia del sistema. En el largo plazo, se evaluará la escalabilidad del sistema, adaptándolo a entornos más complejos y considerando su aplicación en escenarios industriales, como la automatización en almacenes, lo cual podría requerir pruebas en entornos reales para confirmar su viabilidad.

V. AGRADECIMIENTOS

Esta investigación fue financiada por la Agencia Nacional de Promoción Científica y Tecnológica (PICT2019-3024) y la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata (FI-UNMDP).

REFERENCIAS

- [1] O. Kathe, V. Turkar, A. Jagtap, and G. Gidaye, “Maze solving robot using image processing,” in *2015 IEEE Bombay Section Symposium (IBSS)*. IEEE, 2015, pp. 1–5.
- [2] I. Elshamarka and A. B. S. Saman, “Design and implementation of a robot for maze-solving using flood-fill algorithm,” *International Journal of Computer Applications*, vol. 56, no. 5, 2012.
- [3] B. Gupta and S. Sehgal, “Survey on techniques used in autonomous maze solving robot,” in *2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence)*. IEEE, 2014, pp. 323–328.
- [4] *DE0-Nano User Manual*, Terasic Inc., 2016.