

2015-12-01

Implementación de una plataforma móvil utilizando hardware reconfigurable (FPGA) para navegación en entornos dinámicos

Maximiliano Bueno López

Universidad de La Salle, Bogotá, mbueno@lasalle.edu.co

Luis Hernando Ríos

Universidad Tecnológica de Pereira, lhgonza@utp.edu.co

David Ricardo Arango Hurtado

Universidad Tecnológica de Pereira, drarango@utp.edu.co

Follow this and additional works at: <https://ciencia.lasalle.edu.co/ep>

Citación recomendada

Bueno López, Maximiliano; Ríos, Luis Hernando; and Arango Hurtado, David Ricardo (2015)

"Implementación de una plataforma móvil utilizando hardware reconfigurable (FPGA) para navegación en entornos dinámicos," *Épsilon*: Iss. 25 , Article 5.

Disponible en:

This Artículos de investigación is brought to you for free and open access by the Revistas descontinuadas at Ciencia Unisalle. It has been accepted for inclusion in Épsilon by an authorized editor of Ciencia Unisalle. For more information, please contact ciencia@lasalle.edu.co.

Implementación de una plataforma móvil utilizando *hardware* reconfigurable (FPGA) para navegación en entornos dinámicos

MAXIMILIANO BUENO LÓPEZ¹

LUIS HERNANDO RÍOS²

DAVID RICARDO ARANGO HURTADO³

RESUMEN

En este artículo se presenta la implementación de un robot móvil tipo diferencial, utilizando *hardware* reconfigurable (FPGAS-Spartan 3E) para navegación en ambientes dinámicos. En este trabajo se implementaron diferentes algoritmos en VHDL, para las tareas de navegación como control de la velocidad de los motores del robot; lectura del módulo de sensores infrarrojos (el cual se diseñó e implementó), para la detección de obstáculos en el entorno de navegación; lectura de la brújula digital, para determinar la orientación del robot, y una GUI de Matlab para el intercambio de información entre el robot y el controlador. Se muestran los algoritmos y los resultados de algunas pruebas realizadas con el robot en diferentes entornos.

Palabras clave: FPGAS, GUI de Matlab, robot móvil diferencial, VHDL.

¹ Ingeniero electricista. Magíster en Ingeniería Eléctrica. Doctor en Ingeniería. Director del programa de Ingeniería en Automatización, Universidad de La Salle. Correo electrónico: mbueno@lasalle.edu.co

² Ingeniero electrónico. Magíster en Ingeniería Eléctrica. Profesor titular de la Universidad Tecnológica de Pereira. Correo electrónico: lhgonza@utp.edu.co

³ Ingeniero electricista. Correo electrónico: drarango@utp.edu.co

FECHA DE RECEPCIÓN: 16 DE ENERO DE 2015 • FECHA DE APROBACIÓN: 5 DE MAYO DE 2015

Cómo citar este artículo: Bueno López, M., Ríos, L. H. y Arango Hurtado, D. R. (2015). Implementación de una plataforma móvil utilizando *hardware* reconfigurable (FPGA) para navegación en entornos dinámicos. *Épsilon*, (25), 97-119.

Implementation of a Mobile Platform Using Reconfigurable Hardware (FPGA) for Navigation in Dynamic Environments

ABSTRACT

This paper presents the implementation of a differential mobile robot, using reconfigurable hardware (FPGAS-Spartan 3E) for navigation in dynamic environments. Different algorithms were implemented in VHDL, for navigation tasks, such as controlling the speed of the robot motors; reading the infrared sensor module (which was designed and implemented), for detecting obstacles in the navigation environment; digital compass reading to determine the robot's direction, and a Matlab GUI for exchanging information between the robot and the controller. The algorithms and the results of some tests conducted with the robot in different environments are shown.

Keywords: FPGAS, Matlab GUI, Differential Mobile Robot, VHDL.

Implementação de uma plataforma móvel utilizando hardware reconfigurável (FPGA) para navegação em entornos dinâmicos

RESUMO

Neste artigo se apresenta a implementação de um robô móvel tipo diferencial, utilizando hardware reconfigurável (FPGAS-Spartan 3E) para navegação em ambientes dinâmicos. Neste trabalho se implementaram diferentes algoritmos em VHDL, para as tarefas de navegação como controle da velocidade dos motores do robô; leitura do módulo de sensores infravermelhos (para o qual foi desenhado e implementado), para a detecção de obstáculos no entorno de navegação; leitura da bússola digital, para determinar a orientação do robô, e uma GUI de Matlab para o intercâmbio de informação entre o robô e o controlador. Mostram-se os algoritmos e os resultados de algumas provas realizadas com o robô em diferentes entornos.

Palavras chave: FPGAS, GUI de Matlab, Robô Móvel Diferencial, VHDL.

Introducción

PMIR es una plataforma móvil autónoma implementada para la navegación en entornos dinámicos. Está equipada con un anillo de sensores de infrarrojo, con los cuales se puede calcular la posición de los objetos situados dentro de su entorno de trabajo. La inclusión de diferentes tipos de sensores en la plataforma permite una mejor percepción del entorno, ya que por métodos de fusión se pueden aprovechar las mejores características de estos. Tal es el caso de la fusión de los sensores infrarrojos con sensores de ultrasonido, en el que los primeros ofrecen mejor precisión debido a que la longitud de onda involucrada es más corta y presentan menor sensibilidad en condiciones ambientales como la presión y la temperatura (Arkin y MacKenzie, 1994; Omura, Goto y Shidara, 2009). Además, son muy efectivos para distancias de 0,5 a 0,8 m, y no presentan problemas de ruido especular.

Por otra parte, las aplicaciones ultrasónicas basadas en la medición del tiempo de vuelo son más simples y en consecuencia son menos costosas (Hwang, Lo y Liu, 2009). Sin embargo, en estas últimas existe una serie de problemas potenciales que requieren atención: la absorción o atenuación en el medio; el ancho del haz ultrasónico; la presencia de ruido e interferencia; la alta sensibilidad a la temperatura y a la humedad, y la pobre resolución, aunque presentan un campo de acción mucho más amplio —en nuestro caso, en un intervalo de 0,20 y 2,5 m— alrededor del robot mediante la orientación azimutal del cabezal a pasos discretos de $1,8^\circ$.

Medición de distancia con dispositivos infrarrojos

Para la medición de distancia con sensores infrarrojos, el sistema como tal se compone de:

- Un emisor que envía una señal de longitud de onda infrarroja al medio.
- Un obstáculo que sirve como superficie reflectora de la señal (punto de medición).
- Un sistema receptor, el cual, dependiendo de la cantidad de energía con que es excitado, entrega una variación de voltaje proporcional a la distancia.

A partir de la ley de fotometría cuadrática-inversa y un reflector lambertiano perfecto, se puede modelar la salida del sensor como se muestra en (1).

$$s(x, \theta) = \frac{\alpha}{x^2} \cos \theta + \beta \quad (1)$$

El parámetro α define las características del transductor (intensidad de los emisores, sensibilidad al espectro de los fotodiodos y ganancia del sensor), así como del objeto (coeficiente de reflexión del objeto). Los tres primeros factores son constantes para todas las medidas realizadas con el mismo sensor, por lo cual puede expresarse α como el producto de dos factores, uno constante α_0 , en el cual se incluyen aquellos factores dependientes del transductor y se mide en voltios por metro cuadrado; y otro α_i , como coeficiente de reflexión que varía desde 0 para objetos negros hasta 1 para blancos. Por tanto, la expresión para α resulta como se muestra en (2):

$$\alpha = \alpha_0 \times \alpha_i \quad (2)$$

El parámetro β modela el desplazamiento de la señal debido a la luz ambiental. Su valor puede ser obtenido mediante la lectura de la señal en el fotodiodo receptor, con los emisores puestos a cero. Una vez obtenido el valor de β , se podrán realizar medidas, que corresponderán con una señal libre de influencias externas al sensor. De esta forma podemos simplificar la ecuación del sensor como se muestra en (3):

$$y(x, \theta) = s(x, \theta) - \beta = \alpha_0 \frac{\alpha_i}{x^2} \cos \theta \quad (3)$$

Dentro de la literatura de medición de distancia con infrarrojos (Montoya Osorio y Valencia, 2008; Cumplido, 2014) se tiene que para superficies reflectoras que formen un ángulo de 0° con el sensor de infrarrojos se tendrá una medida acertada, conociendo también las características de la superficie reflectora. Esto lleva a plantear la siguiente pregunta: ¿qué ocurre cuando el ángulo de incidencia deja de ser de 0° ? O ¿qué ocurre cuando no se conoce las características de la superficie reflectora? Ante esta situación la medición estará acompañada de un error que depende directamente de la inclinación de la superficie reflectora y sus características. Teniendo en cuenta este aspecto, se podría afirmar que al momento de hacer una medida las principales variables que se consideran en el proceso son la distancia a la que se encuentra el objeto, el coeficiente de reflexión del objeto (dependiente a su vez del color del objeto, del brillo y de la textura) y el ángulo de incidencia del haz

sobre la superficie reflectora. Debe considerarse además la presencia de fuentes de luz, sea solar o artificial, que también adicionan un componente de ruido (Brassai, Enachescu, Losonczy y Marton, 2012; Okura, Matsumoto, Ikeda y Murase, 2003).

De acuerdo con lo anterior, para conocer la distancia entre el sensor y el objeto reflector es primordial conocer *a priori* el ángulo de incidencia θ y la característica de reflexión α_i de la superficie del objeto. Con base en esto se genera la expresión (4).

$$x = \sqrt{\frac{\alpha_0}{y}} \sqrt{\alpha_i \cos \theta} \quad (4)$$

El modelo plantea la necesidad de posicionar dos sensores de la misma naturaleza en forma paralela y en la misma dirección, para así concluir que las variaciones en voltaje entregadas por cada uno de los sensores serán de acuerdo con el ángulo de incidencia que exista con la superficie reflectora.

Descripción de la plataforma

PMIR se compone de las siguientes etapas de *hardware*:

- Conjunto de sensores emisor y receptor infrarrojo
- FPGA. XC3S500E de la familia Spartan-3E, fabricada por Xilinx
- Motores DC, para el desplazamiento de la plataforma y su respectivo circuito de control
- Sistema odométrico
- CN-circuito acondicionador de nivel, encargado de acondicionar los diferentes niveles de voltaje que maneja la plataforma y sus circuitos de control
- Módulo de comunicación X-BEE

Entre las etapas de *software* se tienen los siguientes componentes:

- Componente de control del módulo de motores DC.
- Componente de control del módulo de emisión recepción infrarrojo
- Componente de control del módulo de comunicación
- Componente de control del módulo de memoria
- Componente de procesamiento de la señal

La programación (*software*) de la FPGA (PLD [dispositivo lógico programable]) se hace mediante VHDL (VHSIC, *Hardware Description Language*; VHSIC, *Very High Speed Integrated Circuits*) con el programa Xilinx ISE 8.2i (*Project Navigator*) posteriormente sintetizado en la FPGA. El sistema desarrollado en principio está relacionado con la adquisición de información utilizando el módulo de sensores infrarrojos (Wei y Mahyuddin, 2009). En esta fase se genera la señal emitida por el sensor emisor, se acondiciona la señal obtenida por el receptor, se almacena, se procesa y se calcula la distancia a los objetos más cercanos y se toma la decisión de la trayectoria que se debe seguir.

La otra parte del *software* desarrollado se hizo en Matlab. A través de una interfaz gráfica se encarga de tomar la información proveniente de la plataforma móvil para levantar el respectivo mapa de entorno. En la figura 1 se muestra la plataforma móvil construida y la distribución de sus componentes físicos. Es importante resaltar que dicha plataforma se comunica con un computador central a través de comunicación inalámbrica XBEE.

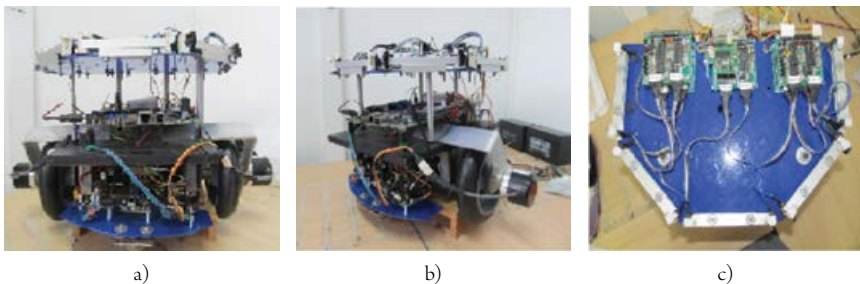


Figura 1. PMIR. Plataforma móvil construida: a) plataforma móvil completa; b) vista lateral y c) vista superior de la plataforma móvil

Fuente: elaboración propia.

El sistema desarrollado controla el movimiento de la plataforma y el sistema de lectura y procesamiento de los módulos de los sensores de infrarrojos para realizar las tareas de navegación por diferentes entornos de trabajo. Una vez la señal es obtenida del entorno, se procesa, se almacena y se envía al PC, el cual interpretará esta información y tomará la decisión de la trayectoria que se va a seguir. Otra parte del *software* ha sido desarrollada en Matlab; esta se encarga de enviar las instrucciones iniciales a la plataforma y luego tomar la información proveniente de los

sensores para calcular y ejecutar la trayectoria por realizar. Es importante resaltar que dicha plataforma se comunica con el computador central (PC) a través de comunicación inalámbrica XBEE. En la figura 2 se presenta un diagrama de bloques general del PMIR.

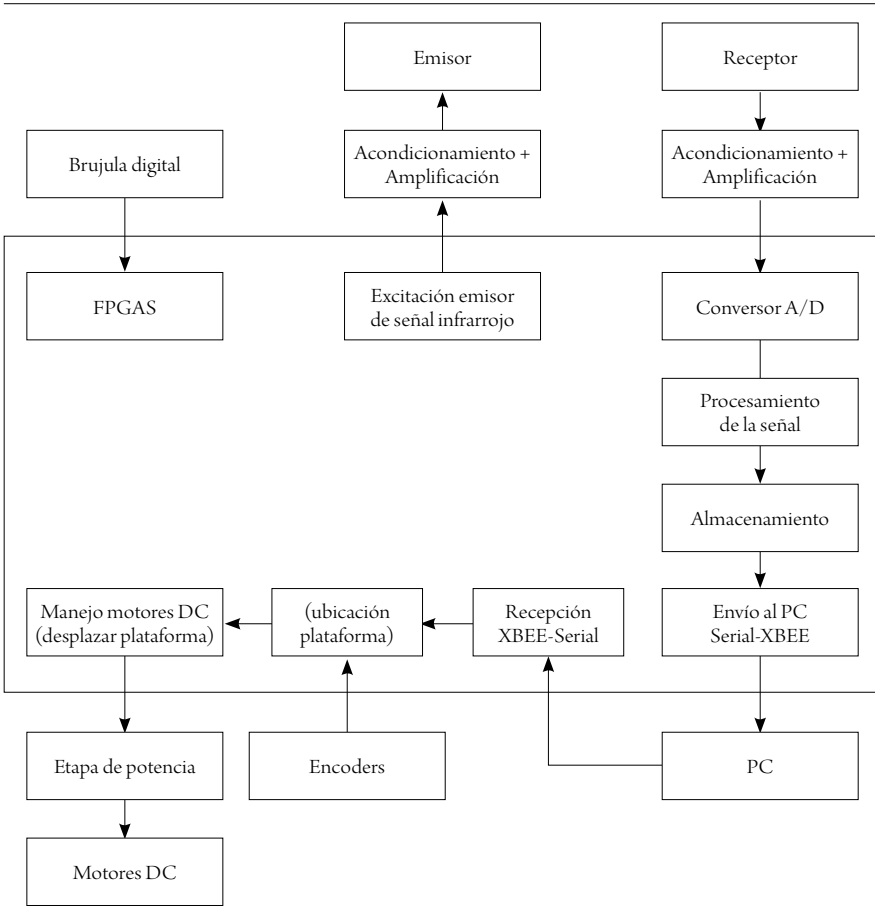
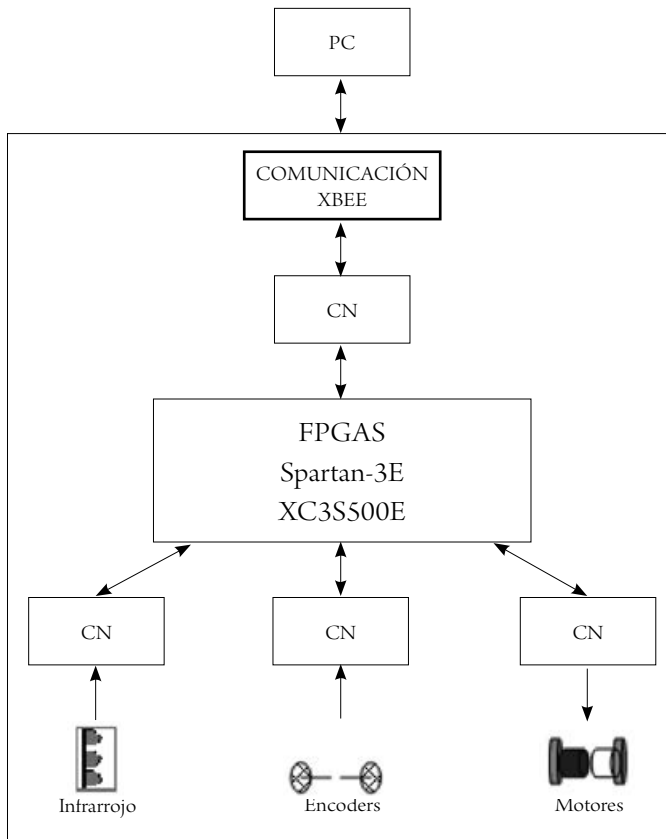


Figura 2. Bloques general de PMIR

Fuente: elaboración propia.

En la figura 3 se presenta la arquitectura de control de la plataforma móvil construida. Esta consiste en un sistema de múltiples componentes *software*, implementados en la FPGAS SPARTAN 3E de XILINX y sus respectivos CN (circuito acondicionador de nivel).



CN: circuito acondicionador de nivel

Figura 3. Arquitectura de control de la plataforma móvil construida

Fuente: elaboración propia.

Movimiento de PMIR

Para localizar un cuerpo rígido en el espacio es necesario contar con una herramienta que permita la localización espacial de sus puntos. La forma más intuitiva y utilizada para especificar la posición de un punto son las coordenadas cartesianas. En un plano, el posicionamiento tiene dos grados de libertad, y por tanto la posición de un punto está definida por dos componentes independientes. Para la descripción del movimiento de la plataforma móvil PMIR en un plano bidimensional se requieren varios aspectos:

- Plantear un sistema de referencia fijo y un sistema de referencia móvil para la plataforma
- Formular una expresión con la cual se pueda determinar la posición del sistema de referencia móvil, en función de un giro y una traslación
- Especificar las posiciones de los elementos que constituyen la plataforma, referenciados al sistema móvil.

El sistema de referencia fijo se escoge como el sistema de coordenadas cartesianas; por otro lado, conviene especificar el sistema de referencia móvil como la posición de un punto representativo en la plataforma, referente a las coordenadas cartesianas y la posición angular de la plataforma respecto al sistema, como se indica en la figura 4.

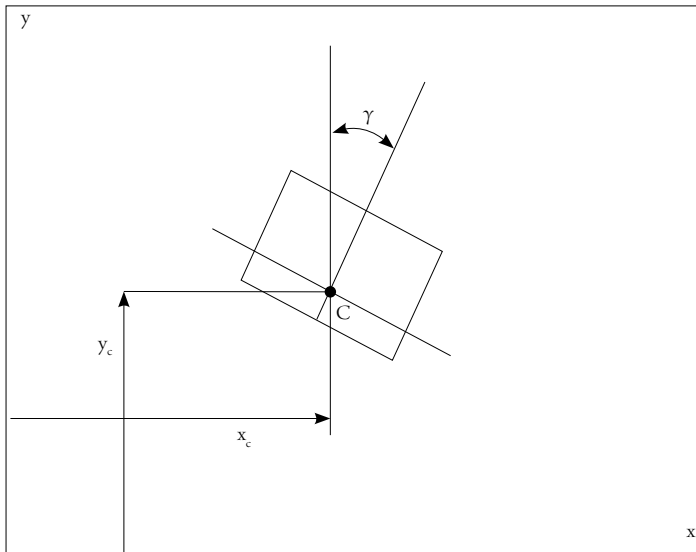


Figura 4. Posición de la plataforma

Fuente: elaboración propia.

El punto c es fijo en la plataforma y tiene coordenadas (x_c, y_c) , y el ángulo γ se toma con referencia al eje vertical y se define positivo el sentido horario. Una nueva posición del centro c puede ser determinada por un cambio en el ángulo γ y un desplazamiento en dicha dirección, como:

γ = Desplazamiento angular de la plataforma o giro.

l = Longitud recorrida por la plataforma.

La figura 5 presenta una descripción de lo anterior. Las coordenadas de c_2 son (x_{c_2}, y_{c_2}) y se calculan según la ecuación (5).

$$\begin{aligned} x_{c_2} &= x_{c_1} + l \cos\left(\frac{\pi}{2} - (\gamma_1 - \theta)\right) \\ y_{c_2} &= y_{c_1} + l \sin\left(\frac{\pi}{2} - (\gamma_1 - \theta)\right) \end{aligned} \tag{5}$$

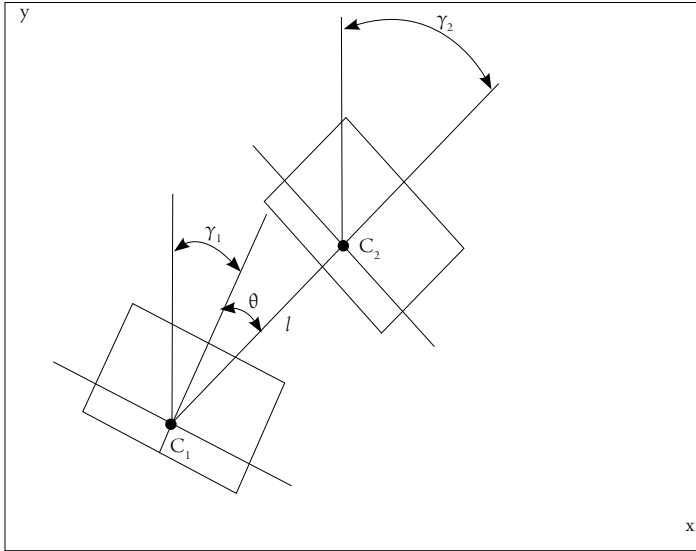


Figura 5. Movimiento de la plataforma

Fuente: elaboración propia.

Luego

$$\begin{aligned} x_{c_2} &= x_{c_1} + l \cos(\gamma_1 + \theta) \\ y_{c_2} &= y_{c_1} + l \sin(\gamma_1 + \theta) \end{aligned} \tag{6}$$

Y la nueva inclinación γ se da por:

$$\gamma_2 = \gamma_1 + \theta \tag{7}$$

De esta forma se puede describir la cinemática de la plataforma, dado que el desplazamiento puede darse directamente o respecto a las velocidades, con el fin de hallar el giro γ y el desplazamiento l y evaluar (1). Con las ecuaciones (5) y (6) se pueden calcular además la posición de cualquier elemento en la plataforma, siguiendo el método de girar y luego avanzar.

Medidor de distancia, implementado utilizando dispositivos de lógica programable FPGAS (SPARTAN 3E)

Para el funcionamiento del circuito medidor de distancia se necesitan dos fuentes duales y un generador de audio con un circuito integrado LM555. Esto denota una cantidad de *hardware* que para la implementación del sensor infrarrojo en una plataforma móvil ocuparía demasiado espacio. La implementación con dispositivos de lógica programable (FPGAS) permite optimizar aplicaciones en las que con un diseño que describa el funcionamiento del sistema completamente se elimine *hardware* innecesario. En este caso se ha sustituido el generador de audio y el circuito modulador implementado con el integrado LM555 por la señal generada por la tarjeta FPGA SPARTAN 3E. La figura 6 presenta el circuito de recepción diseñado.

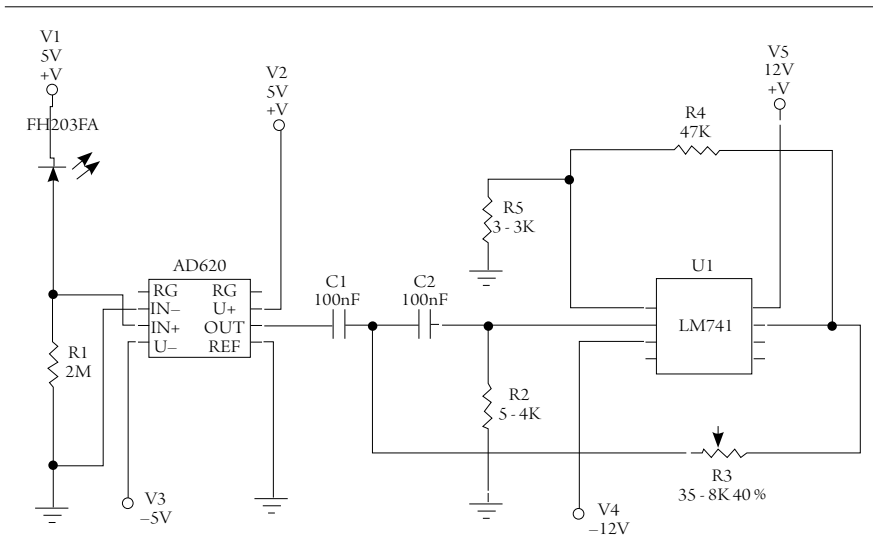


Figura 6. Circuito de recepción

Fuente: elaboración propia.

Generación de la señal modulada

Para optimizar el medidor de distancia se diseñó en VHDL la descripción de *hardware* para generar la señal modulada. Dentro de la descripción se generan dos divisores de frecuencia de 500 Hz y 40 kHz, respectivamente. Luego se hace un producto de ambas frecuencias generando así la señal modulada. Cuando la señal de control se activa con '1' lógico, la envolvente es generada para excitar el sensor y emitir una señal de rayos infrarrojos al medio. Con el oscilador de 50 MHz de la SPARTAN 3E se inicia el generador de 40 kHz de frecuencia con un contador de flancos de subida hasta 1250 pulsos y la salida designada por *emiter_signal_out* se coloca en '1' y se mantiene durante 1250 pulsos más de la señal de reloj de 50 MHz para así generar la frecuencia de 40 kHz. Luego el contador de ciclos se activa y cuenta 40 ciclos de la señal de *emiter_signal_out* y se cambia a cero durante 40 ciclos más de la frecuencia generada de 40 kHz, para así obtener la envolvente de 500 Hz con modulación a 40 kHz debido a que 40 pulsos de 40 kHz equivalen a un pulso de 500 Hz.

La impedancia vista por la base del transistor Q1 del circuito emisor de la figura 7 se hace baja sin tener que utilizar el acoplador de impedancia FET TL-082, por lo que se elimina del sistema de emisión de infrarrojos y se conecta directamente del acoplador PC817 a la base de Q1. Para aumentar el nivel de potencia emitida por los sensores IR se tomó para cada emisor excitación independiente; es decir, por cada emisor hay un transistor 2N3904. El sistema de emisión queda optimizado como se muestra a continuación para el circuito emisor del medidor de distancia 1. La figura 7 presenta el circuito emisor diseñado.

Se continúa con el circuito de recepción, el cual tiene una etapa de acondicionamiento y una de amplificación.

Adquisición de señal, conversión análogo/digital e interfaz gráfica con PC

Prueba para tratamiento de señal proveniente del receptor infrarrojo

Una vez adquirida la señal proveniente del SFH203FA, y acondicionada por el amplificador de instrumentación AD620, se procede a digitalizarla. La digitalización de la señal adquirida se hace por medio de un conversor análogo/digital. Para este

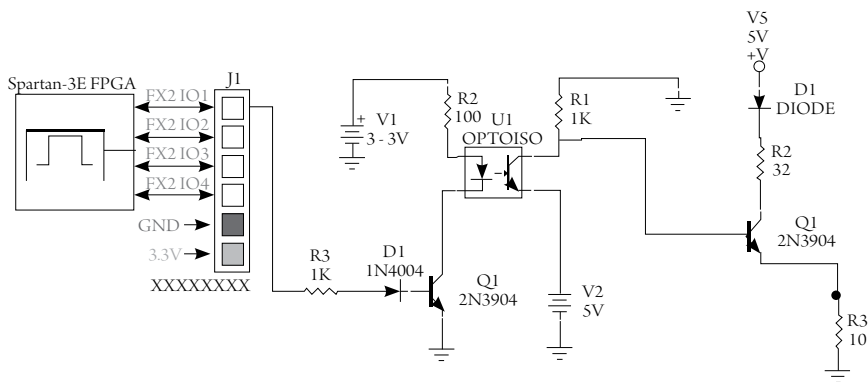


Figura 7. Circuito emisor

Fuente: elaboración propia.

caso, se utiliza un convertor externo a la tarjeta contenido en un microcontrolador Motorola de la familia HC08JK8, el cual contiene ocho canales de conversión y se programa para que funcione a una velocidad máxima de muestreo de 71,5 kHz, de tal forma que el tiempo entre muestras sea de 14 μ s. Dado que la señal de trabajo es de 500 Hz (2 ms), el criterio de Nyquist se cumple plenamente para recuperar la señal obtenida.

Tanto en el microcontrolador como en la tarjeta SPARTAN 3E debe implementarse una trama de comunicación para la transmisión-recepción de los datos, además de una interfaz que convierta niveles de voltaje de 5 V a 3,3 V. Debido a que los datos enviados a la tarjeta SPARTAN 3E son adquiridos por una sola línea de entrada, se debe implementar en VHDL la descripción para recibir la trama de datos provenientes del microcontrolador. El paso siguiente es verificar que la señal haya sido efectivamente adquirida enviando la cantidad de datos muestreados a una interfaz gráfica que permita ver que la señal ha sido adquirida y filtrada. El proceso final es la aplicación de la interfaz gráfica por medio de comunicación serial entre la SPARTAN 3E y el PC.

Interfaz gráfica

Se realizó en Matlab una interfaz gráfica para controlar los movimientos del robot y visualizar tanto su posición proporcionada por los *encoders* y la brújula digital como la información proporcionada por el anillo de sensores infrarrojos. Para iniciar

la tarea de navegación se debe primero definir la posición inicial de la plataforma con respecto a un origen. Para ello, antes de iniciar la navegación de la plataforma se define la posición en coordenadas polares y la rotación de la plataforma con respecto a su propio eje como se presenta en la figura 5.

Una vez definidos los parámetros de la posición inicial se procede a realizar la tarea de navegación enviando las coordenadas de los puntos de la próxima posición que se va a alcanzar por la plataforma móvil con respecto al sistema de referencia móvil. En el momento en que la plataforma llega a la posición parcial deseada, envía la nueva información de su posición y ángulo de rotación con respecto al sistema de referencia de origen. Para la generación de trayectorias se implementó una interfaz gráfica en Matlab que interactúa de forma amigable con el usuario (figura 8). Esta interfaz le muestra al usuario el estado de las variables del robot por medio de un panel frontal y le permite el ingreso de datos de forma fácil y directa. El objetivo es posicionar a la plataforma en un sistema coordenado, para que así el robot móvil pueda desplazarse desde una posición inicial hasta una final previamente establecidas.

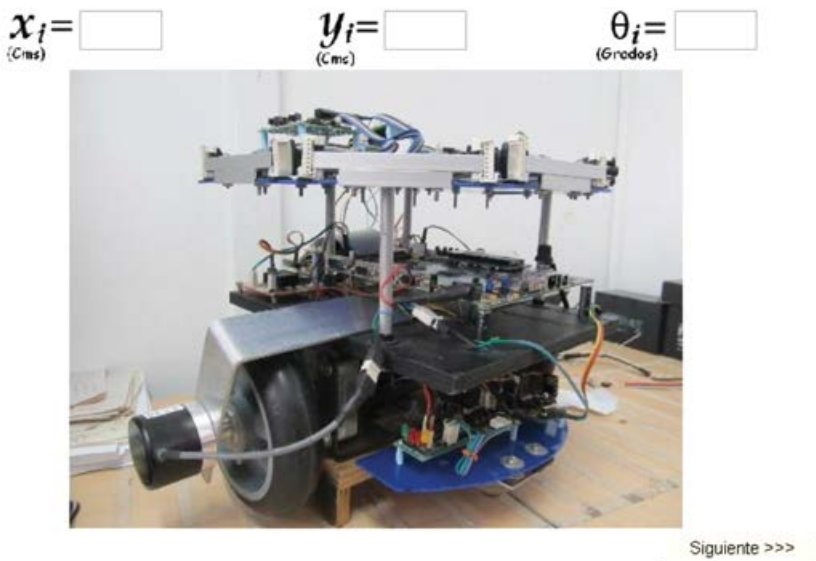


Figura 8. Panel frontal 1 para generación de trayectoria

Fuente: elaboración propia.

La interfaz realizada permite la generación de trayectoria. Dicha trayectoria se muestra en una gráfica 2D. La figura anterior muestra el panel frontal en el que son asignadas las condiciones iniciales determinadas por el usuario. Luego de que la plataforma cumple con la orden inicial, se da paso al segundo panel frontal (figura 9). Este panel frontal corresponde al desplazamiento que realizará la plataforma después de haber determinado su posición inicial; de esta forma, se podrán asignar tantas trayectorias como el usuario desee.

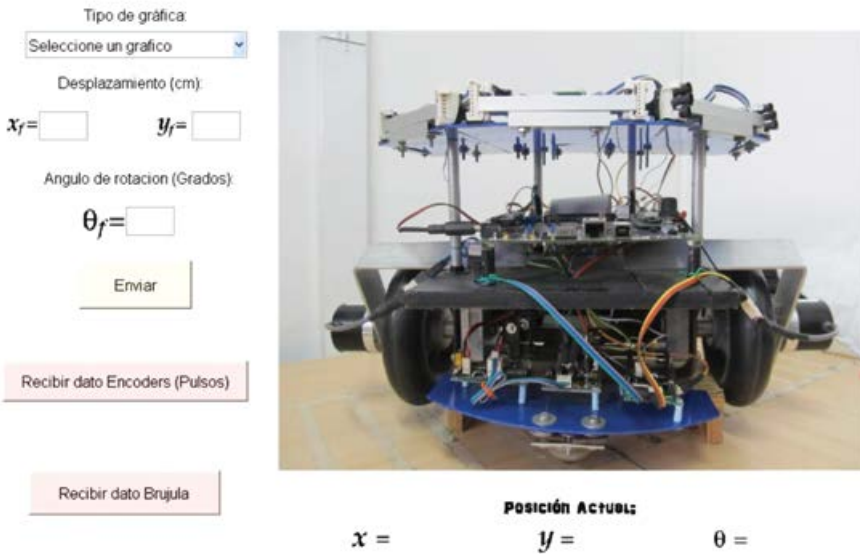


Figura 9. Panel frontal 2 para generación de trayectoria

Fuente: elaboración propia.

Funciones de la interfaz para generación de trayectorias

En el panel frontal 1 se pueden observar tres variables que corresponden al desplazamiento inicial en X, dado en centímetros (X_i), desplazamiento inicial en Y, dado en centímetros (Y_i) y un ángulo Θ_i que corresponde al barrido que debe realizar la brújula después de cumplir con su trayectoria inicial. Al ingresarle los valores de X y Y, la distancia que recorrerá la plataforma es la correspondiente al valor de su hipotenusa,

$$D = \sqrt{X^2 + Y^2} \quad (8)$$

Debido a que dicha distancia es la hipotenusa de los valores X y Y, se calcula previamente un ángulo α , utilizando la ecuación (9). Dicho ángulo de posicionamiento inicial de la plataforma se obtiene haciendo uso de los *encoders* que esta posee:

$$\alpha = \arctang(y/x) \tag{9}$$

El ángulo α es el ángulo hacia donde debe apuntar la plataforma para iniciar su recorrido después de realizada la trayectoria inicial. Finalmente, la plataforma, haciendo uso de la brújula, realiza el barrido correspondiente y se ubica en Θ_i (ángulo inicialmente establecido en el panel frontal 1). Para que la plataforma realice la trayectoria mencionada anteriormente y se ubique en la posición inicial deseada por el usuario, se debe dar paso a la opción *Siguiente*, determinada en el panel frontal 1 y así realizar el envío de datos desde la interfaz hacia la tarjeta SPARTAN-3E. Luego de realizada esta tarea inicial se da paso al segundo panel frontal, mostrado en la figura 10. En la pestaña *Seleccione un gráfico* se debe seleccionar la opción *Ruta*, que mostrará la trayectoria realizada por la plataforma en un gráfico 2D (figura 10).

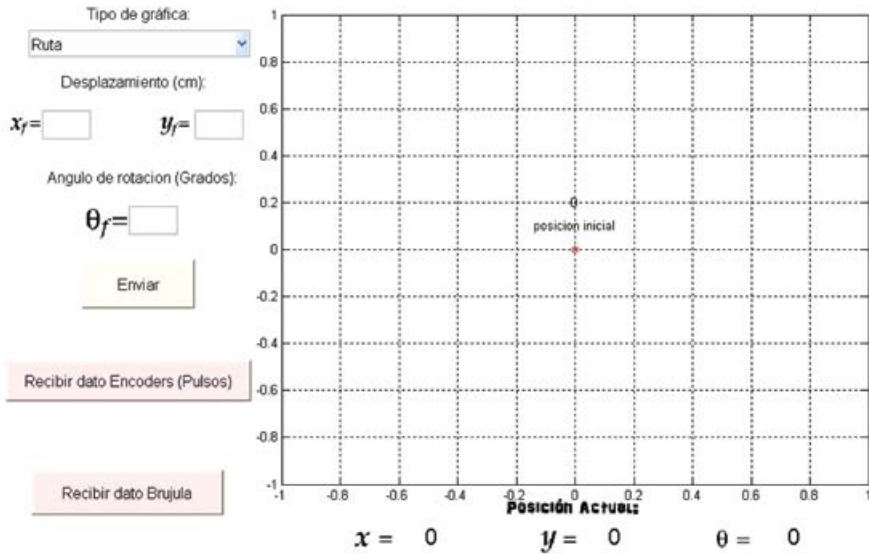


Figura 10. Trayectoria realizada, gráfico 2D

Fuente: elaboración propia.

Posteriormente se ingresan los valores de X_f y Y_f . Esta es la trayectoria final a la que se desea llegar una vez se han establecido los valores iniciales del posicionamiento de la plataforma y el barrido final Θ_f . De igual manera, la plataforma se ubicará en su norte magnético que en este caso corresponde al ángulo Θ anteriormente estipulado. Lo mismo ocurrirá con las otras órdenes de trayectorias que luego se le establezcan. Haciendo uso de los *encoders* se ubica para ejecutar el giro del ángulo α determinado por la ecuación (8), y después de localizado dicho ángulo recorre la distancia D (que corresponde a la hipotenusa dada por las distancias asignadas en X y Y) determinada en la ecuación (9), y finalmente realiza el barrido con la brújula hasta ubicarse en el Θ_f deseado y planteado por el usuario.

Para que esta trayectoria se realice, se debe activar la opción *Enviar* determinada en el panel frontal 2 y así realizar el envío de datos desde la interfaz hacia la tarjeta SPARTAN-3E. La interfaz también muestra dos opciones más, que son: *Recibir dato Encoders* y *Recibir dato Brújula*; estas opciones entregan la lectura de los *encoders* (dadas en pulsos) durante la ejecución de la orden y la medida final de la brújula (dada en grados).

Los datos mostrados por medio de los *encoders* serán los obtenidos durante la realización de la trayectoria D determinada en la ecuación (4). De esta forma, el usuario se podrá asegurar de que la información enviada desde la interfaz se haya recibido de forma correcta y cuál es el valor en pulsos que necesitan determinar los *encoders* para recorrer dicha distancia. Por otra parte, los datos mostrados en la interfaz de la medida de la brújula deben ser muy cercanos al dato consignado en Θ , ya que este es el valor que se le ha dado con anterioridad para que haga el correspondiente barrido de la plataforma. Este valor es de gran importancia, puesto que permite determinar el error alcanzado para detener la plataforma en dicho ángulo.

Después de ser ejecutada cada orden que se le establece a la plataforma, el gráfico 2D muestra todas las trayectorias realizadas y determinadas por el usuario. Del mismo modo, en la parte inferior del gráfico 2D se muestra de manera numérica la posición actual en la que se encuentra la plataforma (X actual y Y actual). También se presenta de forma numérica un seguimiento al ángulo Θ , el cual muestra un acumulado de todos los barridos realizados por la brújula y que fueron asignados por el usuario. Como la plataforma diseñada es una plataforma de experimentación, se deben emplear mecanismos de comunicación entre la plataforma, el PC y viceversa.

Dicha comunicación se realiza haciendo uso de los protocolos de comunicación RS-232; para esta se hace uso de un conector DB9 hembra.

Resultados

En este apartado se presenta una serie de resultados, los cuales son producto de las pruebas realizadas a las diferentes partes que constituyen la plataforma móvil que constatan la veracidad de las herramientas empleadas para lograr los objetivos planteados en el proyecto. Para ejecutar estas pruebas se requieren conceptos básicos sobre robótica móvil, navegación y percepción sensorial. Una vez obtenidos, son suficientes para la iniciación de dichas pruebas.

Resultados iniciales

Debido a que se contaba con la plataforma previamente construida, se llevó a cabo una revisión minuciosa de todos los componentes y partes de esta, tales como cableado, circuitos impresos, elementos circuitales y conexión entre estos, para luego realizar las diferentes pruebas que fueron divididas en varios niveles y obtener así el resultado final.

Resultados de la puesta en funcionamiento de motores y encoders

Para el desarrollo del proyecto se verificó el correcto acople entre los motores y los *encoders*, y de esta forma se garantizó el buen funcionamiento de estos. Por ello se realizaron pruebas de odometría implementando un control con la tarjeta FPGA, y así poder visualizar los errores que se presentan al momento de realizar una trayectoria.

Resultados prueba lineal recorrida

La primera prueba de la aplicación fue llevar a cabo el recorrido de un metro en línea recta para calcular el error en la medida y la desviación en grados respecto a la línea recta trazada, para verificar que el código en VHDL del control esté acertado. En la figura 11 se observa el escenario de prueba y los resultados obtenidos por cada trayectoria de la plataforma. En la tabla 1 se observan los datos obtenidos.

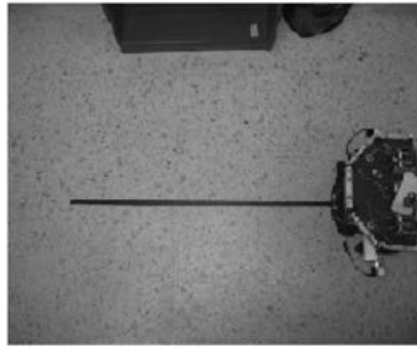
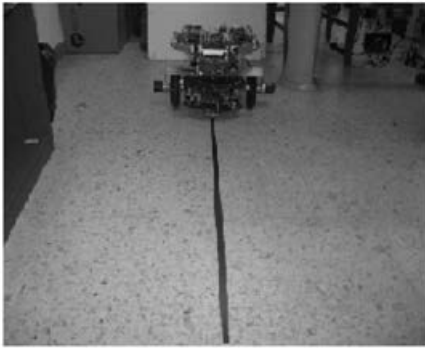


Figura 11. Escenario de prueba distancia lineal recorrida

Fuente: elaboración propia.

Tabla 1. Datos de la prueba para distancia lineal recorrida

NÚM. DE PRUEBA	DISTANCIA RECORRIDA (CM)	ÁNGULO DE DESVIACIÓN (°)
1	96,4	7,0
2	97,0	3,0
3	94,2	9,0
4	100,2	4,0
5	98,8	4,0
6	98,1	0,0
7	96,5	3,0
8	98,5	3,0
9	98,0	3,5
10	97,0	6,0
Promedio	97,47	4,25

Fuente: elaboración propia.

En la tabla 1 se puede observar que la distancia recorrida por la plataforma es aproximada a 100 cm, orden que debía ejecutar, con un error del 2,53 % satisfactorio para lo esperado en el proyecto. La desviación en el recorrido realizado fue de 4,25°, presentando un máximo de 9° y un mínimo de 3°.

Resultados prueba de giro recto

La segunda prueba consiste en programar en la tarjeta FPGA un código cuyo fin sea poner a la plataforma móvil a recorrer una trayectoria en L y verificar si el giro es de 90° (ángulo recto). Se tomaron datos de desplazamiento en línea recta, ángulo de giro y ángulo de desviación con respecto al último trayecto recorrido por la plataforma después del giro recto (figura 12).

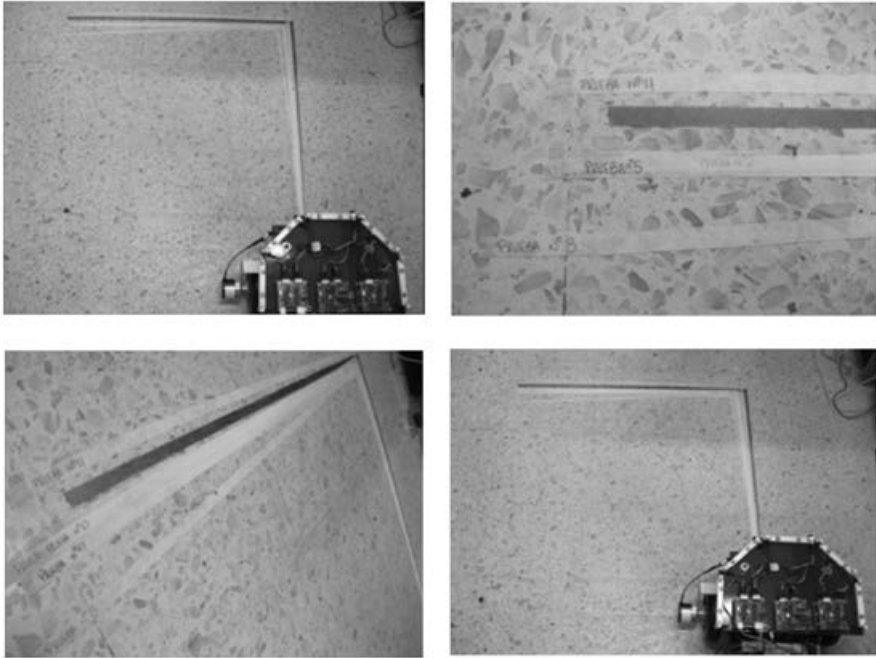


Figura 12. Escenario de prueba de giro recto

Fuente: elaboración propia.

Para esta prueba se determinó que para una distancia recorrida de 80 cm, la cual le fue programada previamente a la plataforma, el valor promedio para 10 pruebas de desplazamiento es de 80,58 cm con un ángulo de desviación promedio de 3° . Con esto se tiene que el error en el trayecto realizado para estas diez medidas es del 0,725 %, un error relativamente bajo y satisfactorio para lo esperado en el proyecto. Para el caso del ángulo de giro, se obtuvo un error del 3,9 % aproximadamente (tabla 2).

Tabla 2. Datos obtenidos en la prueba de giro recto

NÚM. DE PRUEBA	DISTANCIA RECORRIDA (CM)	ÁNGULO DE DESVIACIÓN (°)	ÁNGULO DE GIRO (°)
1	78,9	2	87,0
2	78,5	4	89,0
3	81,0	6	85,0
4	81,6	1	86,0
5	80,0	3	85,0
6	80,0	4	86,0
7	81,3	3	87,0
8	82,2	2	87,0
9	81,3	1	88,0
10	81,0	4	85,0
Promedio	80,5	3	86,5

Fuente: elaboración propia.

Conclusiones y recomendaciones

Se evidencian en la prueba de odometría algunos errores generados por una pequeña desviación existente en el eje de las ruedas, la cual produce en la plataforma un desbalance reflejado en un pequeño error dado en centímetros. La implementación y acople de la brújula digital al sistema de navegación de la plataforma móvil permite proporcionar las órdenes necesarias para la ubicación dentro de un plano coordenado y la corrección del rumbo magnético que se desea en la plataforma.

Al realizar las pruebas del acople entre los motores y la brújula digital, se puede observar que existe un error en la detención de la plataforma con la medida esperada; esto debido al control de inercia del frenado de la plataforma y a la velocidad generada por los motores en su recorrido. Las pruebas de funcionamiento de la plataforma móvil en los trabajos de navegación y evasión de obstáculos con la ayuda de sensores infrarrojos demuestran que los códigos implementados permiten el correcto funcionamiento de la plataforma, así como la recepción de los datos provenientes de los elementos externos a la FPGA. La implementación y utilización de la interfaz gráfica realizada en Matlab permite al usuario interactuar con la plataforma móvil de forma correcta para la generación de trayectorias en diferentes entornos.

Se puede afirmar que la trama de comunicación serial del PC a la plataforma y viceversa es recolectada de forma correcta para el control de estrategias de navegación y la visualización de los resultados obtenidos, luego de la generación de la trayectoria. La implementación de *hardware* reconfigurable para la navegación y generación de trayectorias permite motivar la utilización de este tipo de tecnología y mejorar sus aplicaciones, no solo en el campo de la robótica móvil, sino también en el hecho de expandir su potencial hacia los campos de la investigación en el área de la instrumentación y control, además de permitir su utilización en el área de los sistemas de potencia.

Agradecimientos

Los autores agradecen la colaboración del grupo de Sensorización y Robótica Industrial de la Universidad Politécnica de Valencia, dirigido por el doctor Ginés Benet Gilabert.

Referencias

- Arkin, R. y MacKenzie, D. (1994). Temporal coordination of perceptual algorithms for mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 10(3), 276-286.
- Brassai, S., Enachescu, C., Losonczi, L. y Marton, L. (2012). FPGA based embedded support for mobile robot sonar based navigation. *16th International Conference on System Theory, Control and Computing (ICSTCC)* (pp. 1-6). Sinaia: IEEE.
- Cumplido, R. (2014). *Computer Science Department*. Recuperado de <http://ccc.inaoep.mx/~rcumplido/research.html>
- Hwang, K.-S., Lo, C.-Y. y Liu, W.-L. (2009). A modular agent architecture for an autonomous robot. *IEEE Transactions on Instrumentation and Measurement*, 58(8), 2797-2806.
- Montoya Osorio, A. y Valencia, J. (2008). *Diseño y construcción de un anillo de sensores infrarrojos con dispositivos de lógica programable (FPGAs) para el levantamiento de mapas de entorno y navegación de plataformas móviles* (tesis de pregrado). Pereira: Universidad Tecnológica de Pereira.
- Okura, M., Matsumoto, A., Ikeda, H. y Murase, K. (2003). Artificial evolution of FPGA that controls a miniature mobile robot Khepera. *SICE 2003 Annual Conference* (pp. 2858-2863). Fukui, Japón: IEEE.

- Omura, Y., Goto, A. y Shidara, N. (2009). Surface-trace feasibility for IR-based position-sensing devices. *IEEE Sensors Journal*, 9(10), 1262-1269.
- Wei, C. Z. y Mahyuddin, M. (2009). Neuro-fuzzy algorithm for obstacle avoidance mission of a mobile robot using FPGA. *Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2009*, (pp. 305-310). s.d.

