Revista Tekhnê
Julio - Diciembre 2016, Vol. 13, No. 2, pp. 49– 63

ISSN 1692-8407

Robot semi-autónomo para el transporte de pacientes con movilidad reducida

Semi-autonomous robot for the transport of patients with reduced mobility

Óscar E. González R.
Universidad Pedagógica Nacional
oeg8090@hotmail.com

Diego M. Acero S.
Universidad Pedagógica Nacional
dacero@pedagogica.edu.co

Este artículo documenta el diseño y desarrollo de un prototipo robótico semi-autónomo para el transporte de pacientes con movilidad reducida en un lugar específico, como vivienda, lugar de estudio o trabajo. El perfil de diseño consideró un módulo de tracción diferencial, un módulo de sensores de proximidad, un módulo de control de sistemas de tracción, comando y el modelo del entorno. El sistema posee herramientas embebidas soportadas en microcontroladores, procesadores con arquitectura ARM, y un computador personal. El código se desarrolló en Python sobre sistemas Linux Debian. Las pruebas de desempeño se aplicaron a un prototipo real de laboratorio, sobre el cual se demostró su funcionamiento.

Palabras clave: Control, creación de ruta, discapacidad, prototipo robótico, sensores

This paper documents the design and development of a semi-autonomous robotic prototype for the transport of patients with reduced mobility in a specific place, such as housing, place of study or work. The design profile considered a differential traction module, a proximity sensors module, a control of traction systems module, command and the model of the environment. The system has embedded tools supported in microcontrollers, processors with ARM architecture, and a personal computer. The code was developed in Python on Debian Linux systems. The performance tests were applied to a real laboratory prototype, on which its functioning was demonstrated.

Keywords: Control, disability, robotic prototype, route creation, sensors

Tipología del artículo: Investigación

Fecha recepción del manuscrito: Noviembre 4, 2016
Fecha aceptación del manuscrito: Diciembre 5, 2016

Investigación financiada por: Universidad Pedagógica Nacional.

Edición digital: http://revistas.udistrital.edu.co/ojs/index.php/tekhne/issue/view/798
Cómo citar: González, O. y Acero, D. (2016). *Robot semi-autónomo para el transporte de pacientes con movilidad reducida*. Revista Tekhnê, 13(2), 49-63.

Introducción

El diseño e implementación de un prototipo robótico semi-autónomo se realiza con el propósito de dar una propuesta tecnológica que ayude al traslado y desplazamiento de personas con movilidad reducida, que no cuentan con la movilidad suficiente para accionar los dispositivos ya implementados como joystick, botones, palancas cefálicas entre otras (Challagundla, Yogeshwar, y Harsha, 2014).

Según el *Informe mundial sobre la discapacidad* realizado por la Organización Mundial de la Salud (OMS y BM, 2011), más de mil millones de personas viven en todo el mundo con alguna forma de discapacidad. De ellas, casi 200 millones experimentan dificultades considerables en su funcionamiento. En Colombia según el DANE el 11 % de la población colombiana padece algún tipo de discapacidad, y el 4 % presentan una discapacidad física que les impide movilizarse.

La implementación del prototipo robótico es relevante y marca procesos de desarrollo técnico interesantes (Reddy, Bhatia, y Venkateswara, 2016). En el diseño se utilizaron herramientas computacionales, tarjetas de desarrollo, sensores de ultrasonido y tipo láser de bajo costo como sensores de soporte (Ollero, 2001). Se determinó la mejor forma adquisición de datos sobre el entorno donde se desplaza el prototipo, de igual forma se diseñó un algoritmo computacional el cual calcula por el algoritmo de Tremaux con el método pathfinding A* para la navegación (Florczyk, 2005).

Este es un algoritmo geométrico que ayuda a definir la ruta más corta desde el punto donde está el prototipo, al punto donde se quiere llegar (Barrientos, Peñin, Balaguer, y Aracil, 2007). El algoritmo fue implementado en lenguaje Python, Matlab y C#, lo que garantiza operación multiplataforma y flexibilidad. Sin embargo, el prototipo final solo utiliza lenguaje Python.

La aplicación del proyecto es clara y busca aprovechar las herramientas tecnológicas y computacionales para brindar a un tipo de población específica una herramienta que facilite su desplazamiento autónomo en lugares previamente designados (Rasbridge, 2016). En la parte técnica se encontraron retos como, la navegación y ruta del prototipo de un punto A a un punto B (Crawley, 2012). Estas soluciones incluyen sensores con láser formados con cámaras de bajo coste (García, 2012).

El artículo se encuentra organizado de la siguiente forma. En la Sección 2 se plantea y justifica la necesidad evidenciada, el perfil funcional del prototipo, y algunas otras consideraciones teóricas. En la Sección 3 se describen dos prototipos, y el producto final; en este se plantean los problemas técnicos así como las soluciones propuestas, se detalla la implementación de los sensores de distancia láser y ultrasónicos, el método de creación de la ruta entre los puntos deseados, los alcances de cada prototipo y las mejoras finales. En la Sección 4 se encuentran las pruebas simuladas y reales sobre el prototipo, adicionalmente se proponen futuros desarrollos. Finalmente, la Sección 5 concluye el artículo.

Formulación del problema

Aunque en la actualidad existen muchas alternativas tecnológicas que facilitan la vida de aquellos que presentan alguna disminución de las capacidades motoras aún hay falencias para aquellos que tienen altos grados de discapacidad donde se necesitan ayudas técnicas tan específicas que se hace difícil encontrar una solución en las ya creadas. Este es el caso de patologías o traumas tan severos que limitan el movimiento o lo hacen inexacto.

Estas personas sin una ayuda técnica adecuada estarían relegadas a permanecer confinados en un establecimiento médico o permanentemente en una cama de su hogar, limitando sus diferentes ámbitos de su vida.

El diseño e implementación del prototipo de robot semi-autónomo para el transporte de pacientes con movilidad reducida, se hace necesario ya que cubre las necesidades de desplazamiento de aquellas personas que por razones médicas no se pueden movilizar por sus propios medios en los ambientes habitacionales, donde más se desenvuelve como vivienda, lugar de estudio o trabajo. Por ser un prototipo semi-autónomo garantiza que con una mínima interacción del usuario se pueda llegar de un punto A previamente designado a un punto B. Este prototipo busca dar mayor independencia de movimiento.

Se busca que el robot sea programable a un lugar específico, ejemplo vivienda, lugar de estudio o trabajo. Dentro de estos lugares se espera que se pueda identificar los lugares más relevantes con puntos específicos donde el usuario podrá seleccionarlos por medio de un celular, donde se mostrará el plano del lugar resaltando los posibles lugares por donde se puede desplazar el prototipo. El sistema debe ser capaz de buscar la mejor ruta, identificar obstáculos y re-definir la ruta hasta llegar al lugar seleccionado.

Este prototipo de robot se divide en los siguientes módulos:

Módulo de Control

El módulo de control tendrá como objetivo conducir el prototipo hasta una posición previamente configurada mientras evita los choques con los obstáculos detectados por los sensores. Además estará encargado de generar un modelo del entorno para tener un desplazamiento más eficaz y dinámico.

Este módulo de control se compone de un computador portátil en el cual se guarda la matriz que representa el plano del sitio previamente configurado. Sobre ella se realiza

ISSN 1692-8407

el proceso de búsqueda de ruta y toma de decisiones. Éste módulo también está integrado por la tarjeta de desarrollo Raspberry Pi ® que se encarga de la medición de distancia por medio de los ultrasonidos, de encender los láser de acuerdo a lo indicado por el computador, y de la comunicación del computador y el microcontrolador PIC ® encargado de los *encoder* y PWM de los motores.

Módulo de desplazamiento

Posee dos motores eléctricos de corriente continua con tracción diferencial. Los dos motores de tracción serán controlados por un *driver* de 24 V a 12 A, y dos ruedas libres que permitirán la dirección.

Módulos de comunicación

Este módulo está integrado con diferentes tipos de comunicación. La primera entre el computador y la Raspberry Pi. El sistema se conforma por un cable USB a RS232. La segunda está integrada por un sistema *bluetooth* que conecta al celular y el computador. Finalmente, el tercero es de comunicación paralela entre la Raspberry Pi y el PIC que controla las funciones de desplazamiento del motor.

Metodología

Primer prototipo

El desarrollo del prototipo se realizó sobre los microcontroladores PIC 16F877A, 18F2550 y 18F4550. Se seleccionó también motores Pololu de 1 kg, sensores ultrasónicos SH-04, una base de madera, batería lipo de dos celdas, *driver* para los motores L293 y reguladores de 5V (Pololu, 2008).

El mapa por donde se desplazara el prototipo es representado en una matriz $m \times n$ determinada por el ambiente, donde el numero uno serán paredes u obstáculos. El número cinco será el punto de llegada, este punto de llegada en este prototipo solo se ingresa por medio de programación en lenguaje C#. el punto de partida se determinó con el número 30, y los lugares por donde se podría llegar a desplazar son designados con cero. Estos valores se determinaron aleatoriamente y no tienen ningún trasfondo técnico (Fig. 1).

```
mapa =

     1     1     1     1     1     1     1     1
     1     0     0     0     0     0     5     1
     1     0     0     0     0     0     0     0
     1    30     1     0     1     1     1     1
     1     0     0     0     0     0     0     1
     1     1     1     1     1     1     1     1
```

Figura 1. Matriz representativa de mapa.

Como primer paso se halla la posición de salida por medio de un ciclo que entrega fila y columna donde se encuentra el prototipo. Luego se inicia el proceso para crear el vector de la ruta idónea. Este proceso inicia preguntando si la posición de llegada es la misma de partida, ya que puede ocurrir que el usuario determine el mismo punto. Si no lo es, el algoritmo determina si hay espacio para desplazarse hacia la izquierda, derecha, atrás o adelante (Fig. 2).

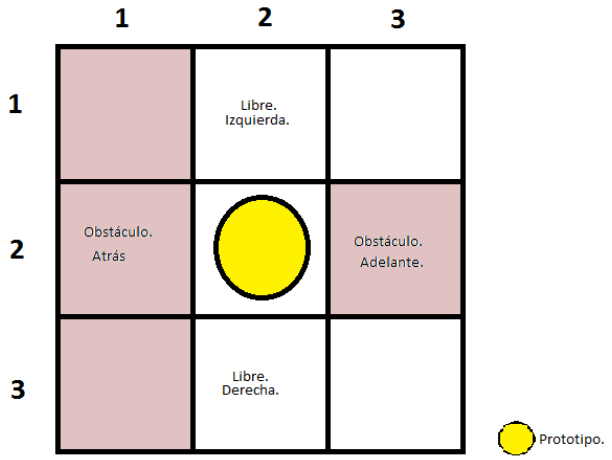

Figura 2. Representación de obstáculos fijos y espacios libres en el mapa cargado al prototipo.

Si hay más de una opción, se guarda la posición, ya que si el camino seleccionado no es el correcto se retrocederá a esta posición para buscar otra posible opción de ruta. Luego, se avanza según la secuencia determinada que marca la siguiente prioridad: adelante, derecha, atrás e izquierda (esta secuencia es arbitraria definida por el programador). Por ejemplo, si el prototipo pudiera girar a la derecha y a la izquierda, el programa determinaría que se dirija a la derecha.

No obstante esto, no implica que el prototipo tenga que realizar estos desplazamientos, ya que estos procesos se realizan en la matriz representativa del mapa. Cuando se selecciona la posición, se reinicia el proceso de preguntar si las posiciones aledañas están disponibles, así hasta que encuentre la ruta hasta la posición deseada. Cuando se crea el vector de ruta otro algoritmo se encarga de determinar los giros y desplazamientos del prototipo (Figs. 3 y 4).

Algoritmo de dirección de acuerdo a vector de ruta hallada. Antes de iniciar el desplazamiento, el prototipo lee los sensores de ultrasonido. Si no hay ningún obstáculo a menos de 20 cm, se inicia la marcha. Cada vez que se ejecute una orden de movimiento el sensor medirá la distancia.

El vector de dirección entregado en el proceso anterior da los valores de la posición del prototipo hasta el punto de llegada. No obstante, estos valores no son interpretados, ni da las características de los giros que debe dar el prototipo, ya

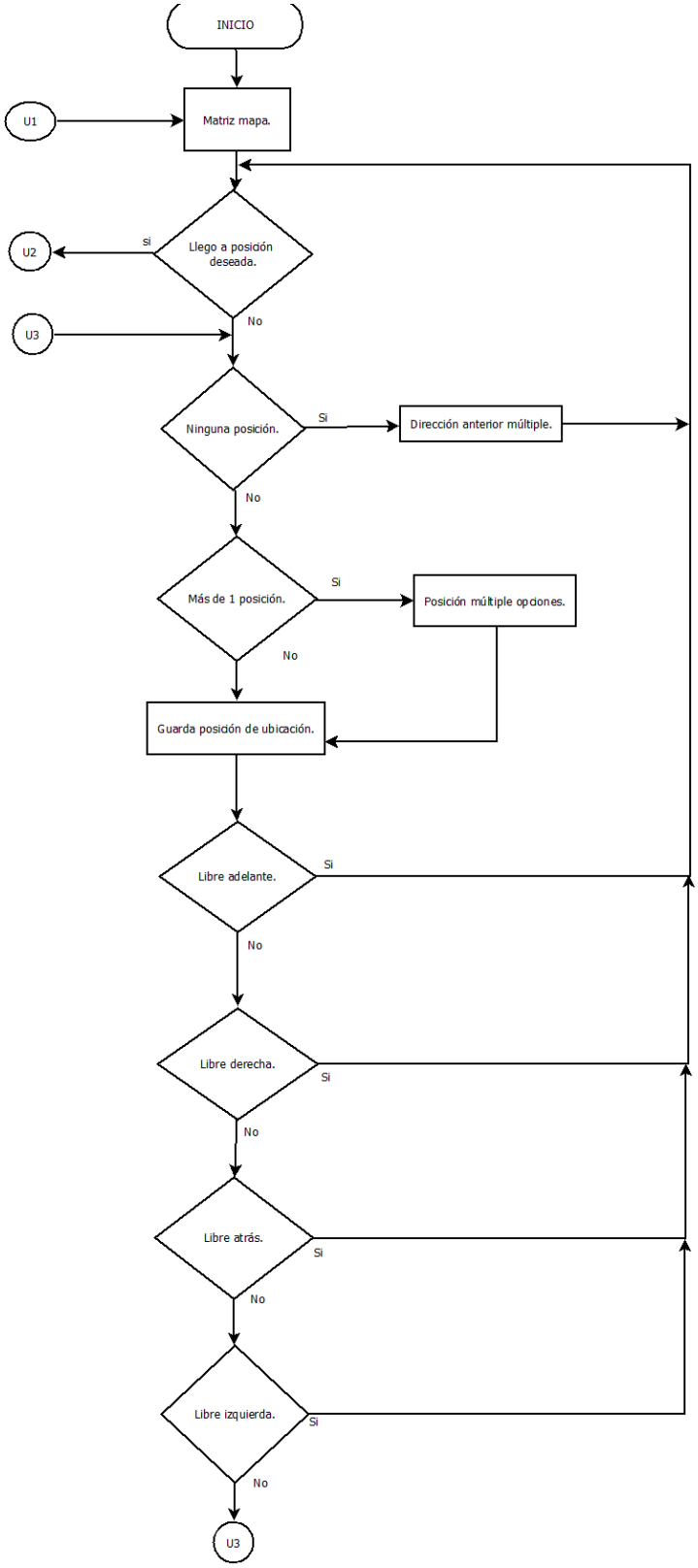

Figura 3. Flujograma proceso creación de ruta del prototipo.

ISSN 1692-8407

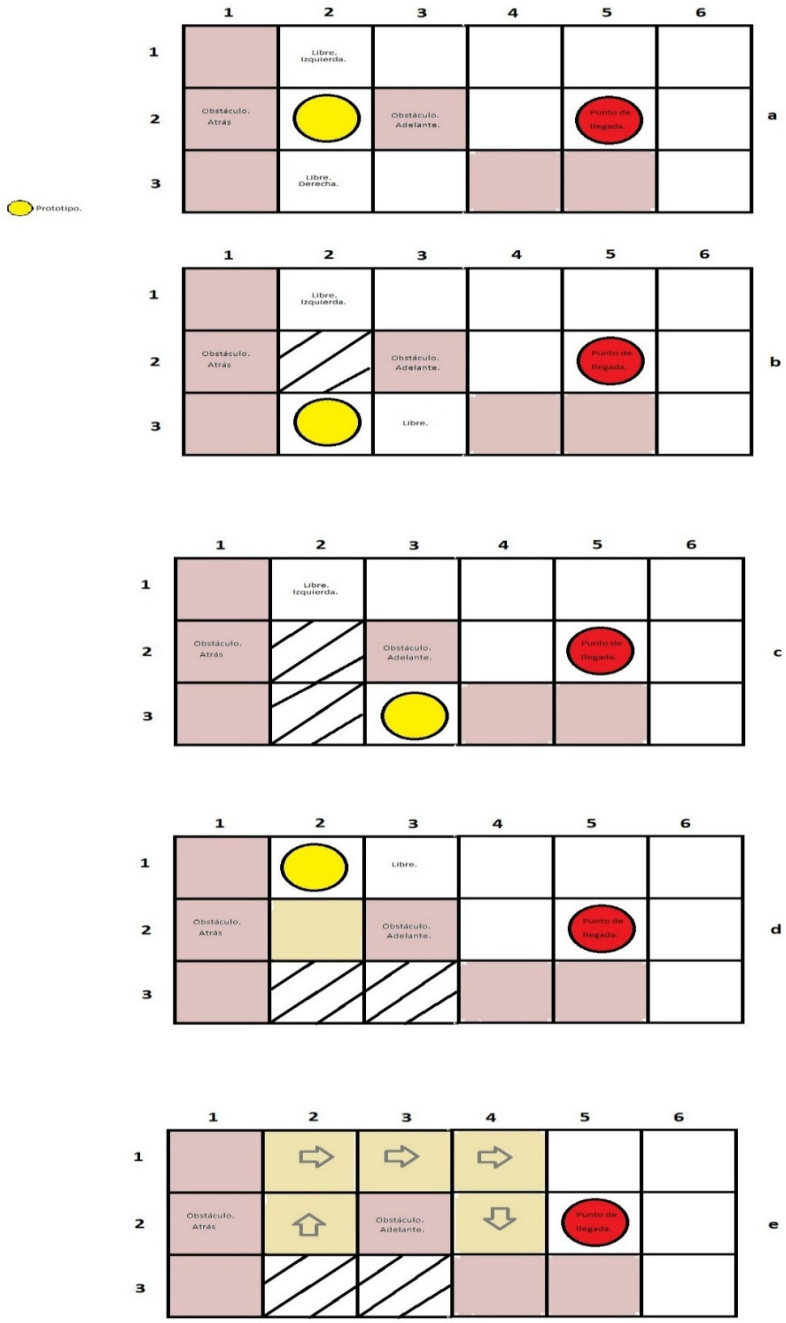

Figura 4. Ejemplo creación de ruta.

que este no es capaz de desplazarse en todas las direcciones sin hacer giros sobre su propio eje (Fig. 5 y tabla 1).

Tabla 1
Matriz de posiciones del ejemplo.

Fila	2	1	1	1	2	2
Columna	2	2	3	4	4	5

En este ejemplo de mapa y vector resultante de dirección de posiciones, se observa que el prototipo estaría de frente a un obstáculo. Si aumenta la posición de la columna, provocaría que se estrellara con el obstáculo, o simplemente no se generaría movimiento debido a la condición de la mínima distancia de los sensores ultrasónicos. Para generar un desplazamiento acorde se debería seguir la siguiente secuencia: giro a la izquierda, avance adelante, giro derecha, avance adelante, avance adelante, giro derecha, avance

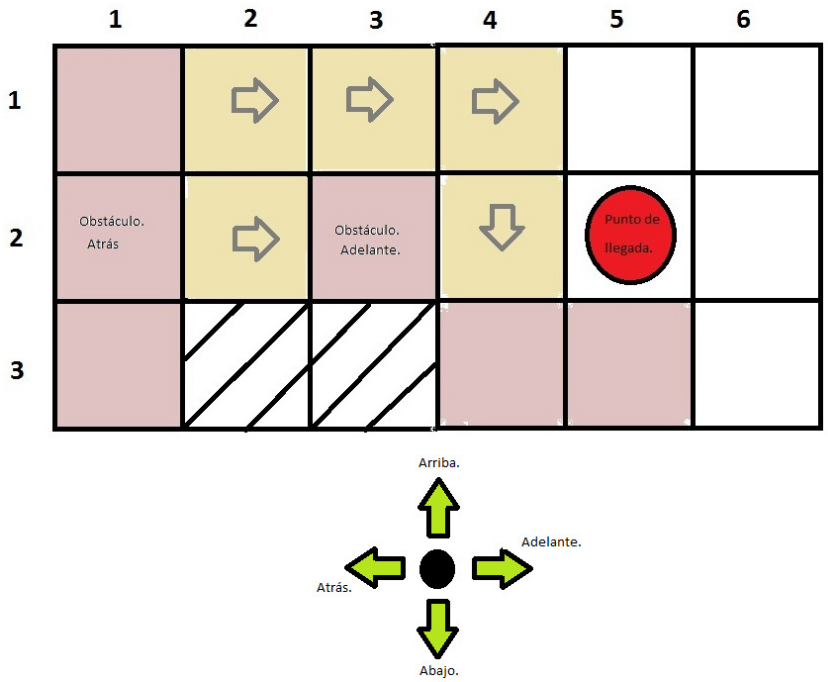

Figura 5. Ruta hallada con algoritmo implementado.

adelante, giro a la izquierda y por ultimo avance adelante. El algoritmo que se diseñó se basa en el sentido del prototipo y los cambios entre las filas y columnas. Este algoritmo tiene una complejidad especial ya que al dar un giro sea cual sea el sentido este cambia la disposición del mapa (Fig. 6).

El sentido del prototipo es importante. Para procesos de orientación se designarían cuatro sentidos: frente, atrás, arriba y abajo. Con esto es posible saber la orientación frente al mapa. Luego de cada posición, y de acuerdo a los cambios entre filas y columnas, se puede determinar si al avanzar por la ruta hallada se origina un nuevo cambio de sentido.

Hardware. La base del circuito impreso utilizado para el prototipo PIC es de 10 cm × 10 cm. En la parte delantera se ubicaron los dos motores Pololu con una típica configuración tracción delantera, buscando una mejor maniobrabilidad. Las llantas acopladas a estos motores fueron las llanta Pololu 42x19. En la parte trasera se ubicó la rueda libre, con esta el prototipo tiene tres apoyos a la superficie por donde se desplaza. En la parte inferior se posicionó el *driver* L293, el cual da la potencia a los motores. En la parte superior se ubicó la tarjeta del PIC 18F4550, los reguladores de voltaje y la batería. Este prototipo cuenta con tres sensores ultrasónicos, los cuales están ubicados uno en el frente del prototipo, el cual evita choques frontales. Los otros dos se ubican en cada parte lateral (izquierda y derecha) (Fig. 7).

Alcance del prototipo. Con el desarrollo de este prototipo se pudo tomar decisiones en cuanto a la implementación de los algoritmos que hallan la mejor ruta, evidenciando también vacíos y una redundancia de pasos

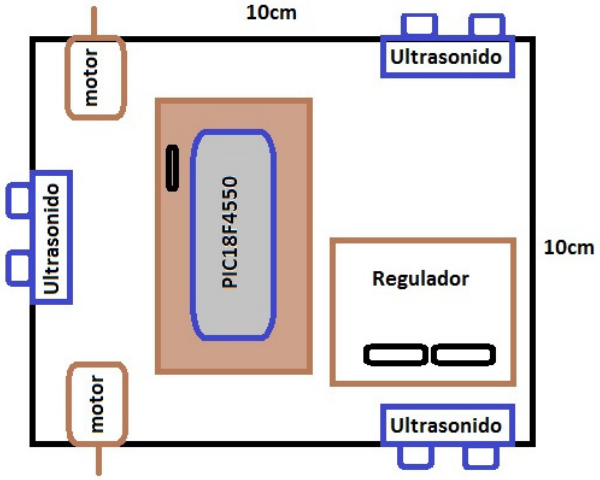

Figura 7. Distribución física de componentes en primer prototipo.

que posteriormente fueron corregidos en el prototipo final. Otro aprendizaje dado en la construcción de este primer prototipo es la necesidad de implementar los diferentes sensores de acuerdo a la altura media del prototipo, al igual que el número mínimo de sensores para lograr un óptimo desplazamiento.

Segundo prototipo

El desarrollo del prototipo se realizó sobre la tarjeta Raspberry Pi (512 Mb, 700 Mhz, sistema operativo Linux

ISSN 1692-8407

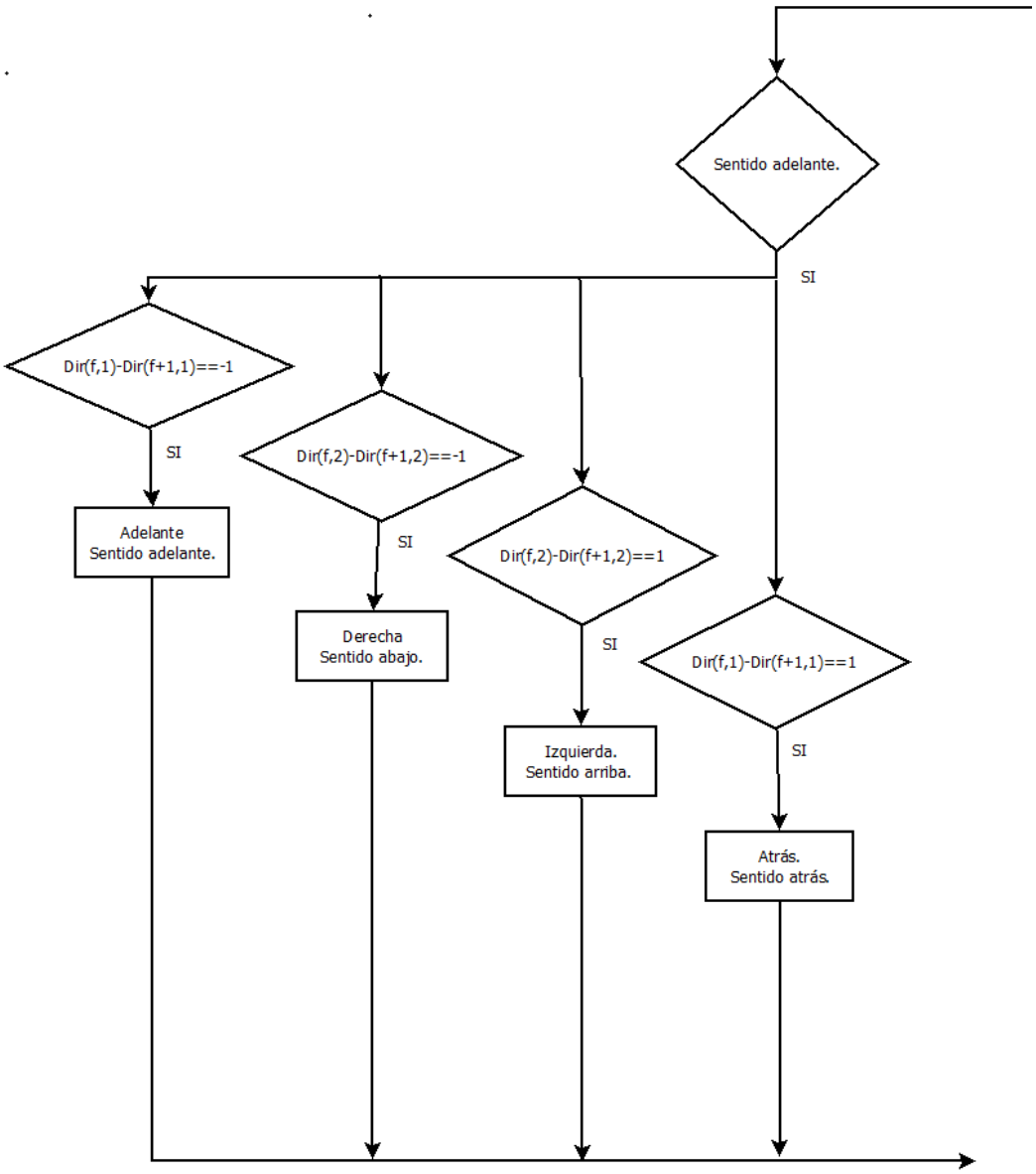

Figura 6. Flujograma proceso de desplazamiento.

Debian Jessy). Se utilizaron también una webcam genérica de 640 x 380 pixeles, sensores láser de baja potencia, ultrasonidos SH-04, y un *driver* para el láser. Los programas se desarrollaron en Python.

Dentro del desarrollo y montaje del prototipo se planteó que la tarjeta Raspberry Pi desarrollara todo el procesamiento central. Sin embargo, dentro del trabajo de implementación del sensor con la webcam, se evidenció que la captura y procesamiento de imágenes en esta plataforma es muy lenta. En promedio entre captura y la entrega de resultados se tienen retardos de 5 segundos. Si se toma este tiempo y se multiplica por las cuatro cámaras conectadas, se tiene un tiempo muy grande, algo que sería totalmente ineficiente. Por otro lado, con este prototipo se implementó un sensor láser donde la webcam captura tres láseres encendidos con una secuencia específica. No obstante, como estas cámaras webcam tienen un plano focal reducido, se limita la medición de los láser que se ubican a los costados (Figs. 8 y 9).

Descripción del prototipo. En este prototipo se desarrollaron los algoritmos que entregan la medición de los ultrasonidos utilizando los pines GPIO, al igual que la secuencia que enciende los láser según la necesidad del programa principal. Igualmente en esta tarjeta se diseñó la primera versión de código de la solución de ruta en el mapa

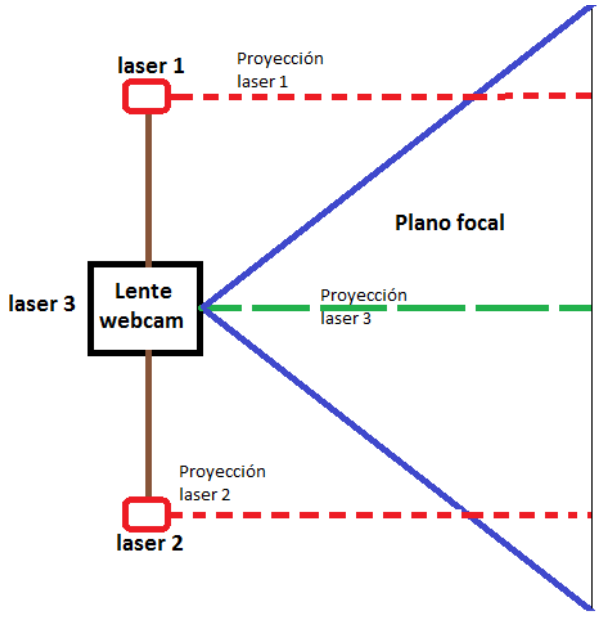

Figura 8. Esquema modelo sensor 3 de láser y una webcam.

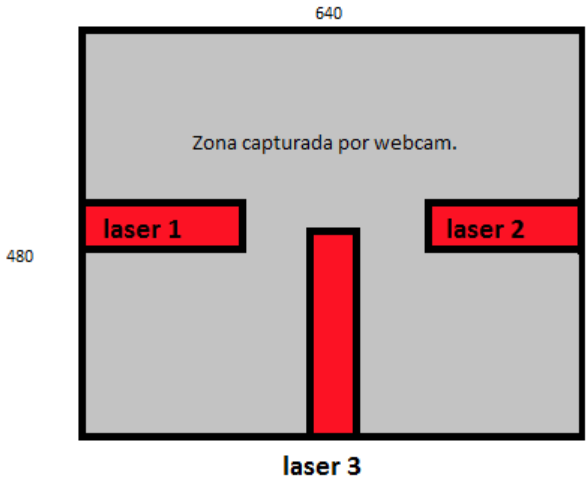

Figura 9. Recorrido probable de los 3 laser en imagen capturada.

cargado. Todo este código se programó en Python utilizando librerías *open source*.

Alcance de prototipo. Ya teniendo la lógica clara sobre la adquisición de la ruta, la representación de un entorno por medio de una matriz, el algoritmo de dirección para interpretar la ruta, el principio de localización por triangulación de ultrasonido, y el control básico de dos unidades motoras para desplazamiento que se evidenció con la construcción del prototipo PIC, se puede migrar esta lógica a una plataforma de desarrollo más robusta con el propósito de incluir ciertas características, en este caso la Raspberry Pi. No obstante, como ya se mencionó anteriormente, con las repuestas dadas en velocidad y eficiencia, no es viable continuar el diseño con esta tarjeta de desarrollo. Sin embargo, sí se pudieron hacer avances importante en la programación, fue por ello que se decidió no eliminar totalmente la Raspberry Pi del prototipo final, sino acoplarla a un computador que realizara los procesos de mayor costo computacional.

Prototipo final

El desarrollo del prototipo se realizó sobre un computador portátil con procesador AMD Athon 64x, con 4 GB de RAM, corriendo un sistema operativo Linux Ubuntu 12.4. Se seleccionó también la plataforma Raspberri pi B+ con la configuración del segundo prototipo, un celular con Android, cuatro webcam, cuatro láser de baja potencia, cuatro sensores ultrasónicos, motores de alta potencia, driver dual de alta potencia, dos baterías de 12 V y 7 Ah, y un marco silla de ruedas para montar sistema de control y propulsión.

Sensor láser. El sensor láser diseñado para el prototipo se compone de una webcam marca Unitec con una resolución nativa de 640 x 480 pixeles, con un ángulo de apertura de lente 45 grados en su eje horizontal, y de 40 grados en su eje vertical, y un láser de 100 mw.

El proceso para hallar la distancia es el siguiente: El programa principal envía la señal a la Raspberry Pi donde por medio de los pines GPIO enciende el láser elegido. La webcam captura la imagen, donde queda plasmado el haz del láser. Cuando la imagen es capturada, el programa principal da la orden a la Raspberry Pi de apagar el láser. Luego de esto, la imagen es procesada de la siguiente forma: la imagen ya capturada es cargada como matriz, esta matriz de 640 x 480 tiene las tres componentes RGB. Como el láser es naturalmente rojo, se puede quitar las componentes verdes y azules, esto agiliza el proceso computacional, ya con la componente roja se define la zona donde se presume que el haz del láser va a quedar plasmado. Esta es del centro inferior hacia abajo como se observa en la Fig. 10.

Cuando se toma solo esta zona se elimina en gran parte las posibles lecturas erróneas que se pueden presentar por la captura de diferentes superficies luminosas o de color rojo. El algoritmo busca las zonas con mayor intensidad, esta intensidad se define en la matriz por representaciones numéricas que van desde 0 a 255. Dentro del programa se puede definir cuál es el mínimo rango que se tomará para evaluar la posición del haz, con respecto a los pixeles. Como el haz del láser ocupa más de un pixel se realiza un promedio para tener un solo dato del valor de posición, con este valor de posición y la ecu. 1.

$$D = \frac{h}{tan\,(pfc \times rpc \times \rho)} \tag{1}$$

Ecuación que da la distancia del prototipo al objeto. Este sensor láser es capaz de medir distancias mínimas de 23 cm

ISSN 1692-8407

Figura 10. Zona de desplazamiento del láser capturado.

a máximo 2.8 m. Este sensor tiene variaciones en la medida real y la medida capturada, entre mayor es la distancia del objeto mayor será el porcentaje de error (Fig. 11).

El sensor láser es construido con una webcam y un láser que se ubican de la forma mostrada en la Fig. 12.

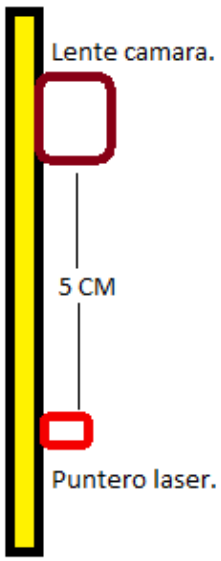

Figura 12. Distribución física entre el lente de la cámara y puntero láser.

El láser está a una distancia de cinco centímetros del lente de la cámara exactamente en la parte inferior. Esta ubicación se debe a que entre menos distancia un obstáculo esté se reflejará en los pixeles más lejanos del punto medio del plano focal (Fig. 13).

Sensores ultrasónicos. Los sensores ultrasónicos utilizados fueron los HC SR04, capaces de medir distancias

Figura 13. Sensor láser.

mínimas de 2 cm hasta 4 m. Estos sensores fueron creados específicamente para medir distancias, por ello son muy fáciles de implementar. No obstante hay que tener cuidado con ciertas características técnicas que entran en conflicto con la Raspberry Pi, ya que los voltajes que manejan son diferentes. El pin *echo* que entrega el sensor ultrasónico tiene niveles de 5 V, si estos voltajes son entregados directamente a los pines GPIO de la Raspberry Pi ocasionaría daños en la tarjeta. La señal *trigger* entregada por la Raspberry Pi es de 3.3 V, y aunque el fabricante recomienda 5 voltios para ser disparada, en las pruebas se observó que con el voltaje entregado es suficiente.

Mapa. Las características del mapa por donde se desplazará el prototipo y con en el cual se realizaron todas las pruebas de del algoritmo que halla la ruta más eficaz cuenta con las siguientes medidas: 7 m de ancho y 20 m de largo. En esta área se dibujó un plano de una ambiente real, con 12 posibles posiciones. Ya teniendo claro las barreras y obstáculos fijos, se dividió el área total generando una cuadricula con espacios de 30 cm, que es la unidad mínima de desplazamiento del prototipo. Con estas divisiones se creó la matriz representativa del mapa. Dentro de la matriz los espacios libres fueron representados con ceros y los obstáculos con 100 (Figs. 14 y 15, y tabla 2).

Algoritmo generador de ruta. Este algoritmo se basó en dos métodos conocidos: Tremaux con el método pathfinding A*. Al tener la matriz que representa el terreno, el punto de partida y el punto de llegada del prototipo, se procede a crear una matriz auxiliar donde cada espacio de la matriz se llena con un valor decimal. Este valor decimal representa que tan lejos está el punto de llegada del punto de

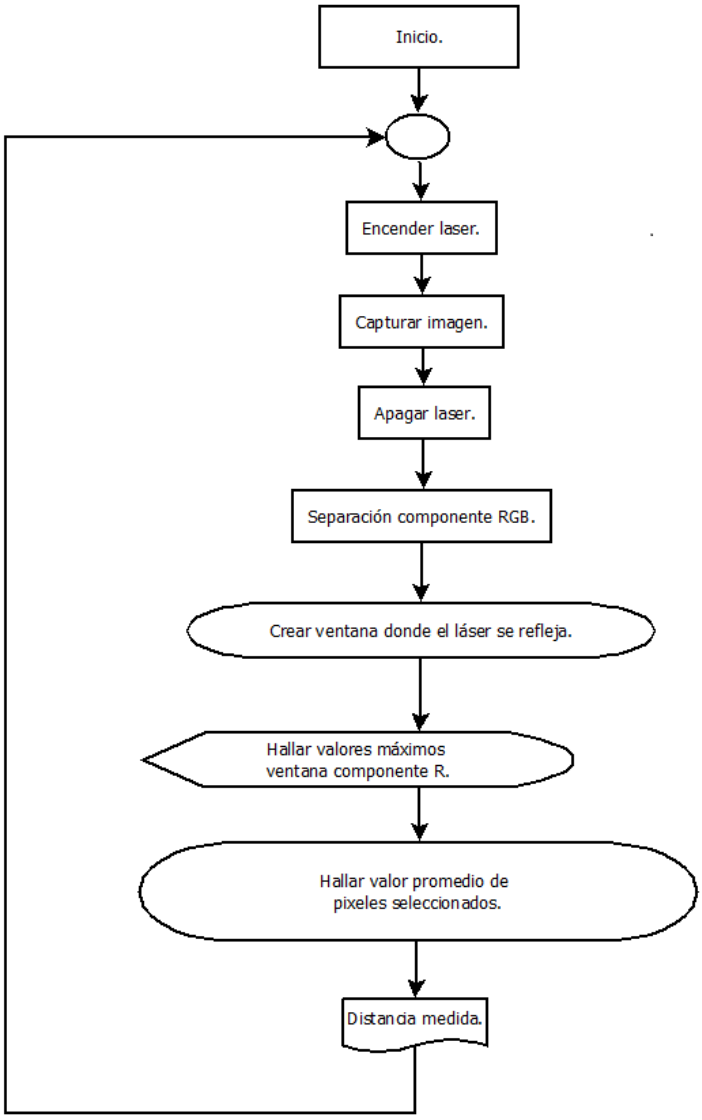

Figura 11. Algoritmo para hallar distancia con sensor laser.

Figura 14. Plano lugar de implementación del prototipo.

ISSN 1692-8407

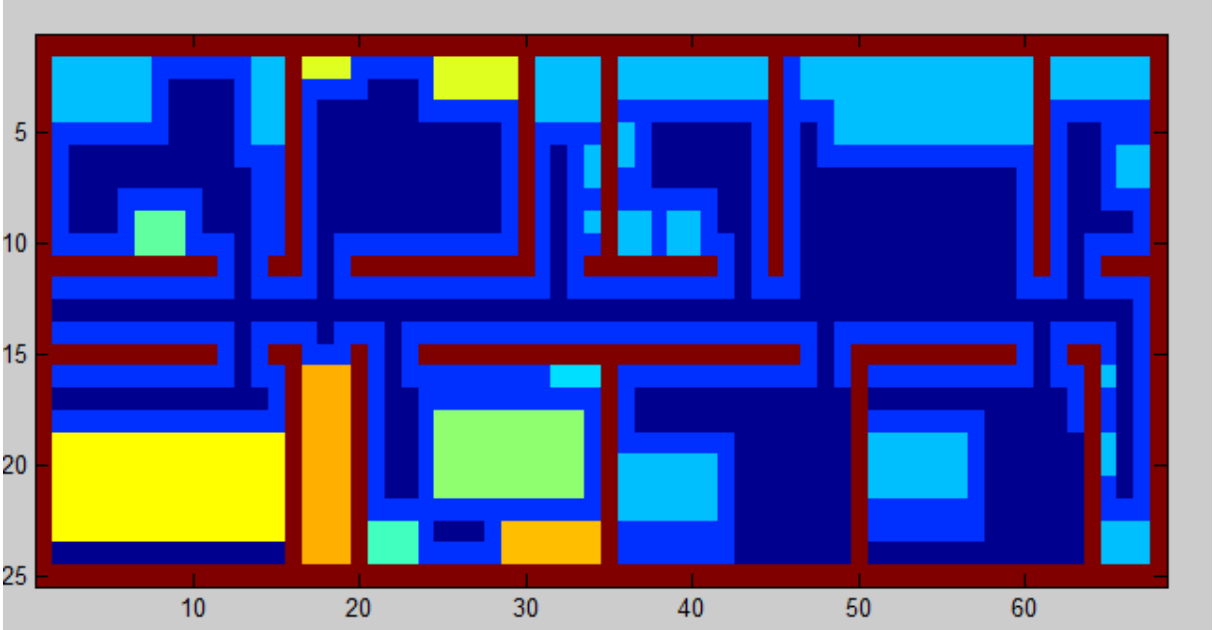

Figura 15. Representación matricial plano.

Tabla 2
Matriz de prioridad. 0 ubicación lugar de llegada.

	1	2	3	4	5	6	7	8	9	10	11
1	110	100	90	80	70	60	50	40	30	20	30
2	100	90	80	70	60	50	40	30	20	10	20
3	90	80	70	60	50	40	30	20	10	0	10
4	100	90	80	70	60	50	40	30	20	10	20
5	110	100	90	80	70	60	50	40	30	20	30
6	120	110	100	90	80	70	60	50	40	30	40
7	130	120	110	100	90	80	70	60	50	40	50
8	140	130	120	110	100	90	80	70	60	50	60
9	150	140	130	120	110	100	90	80	70	60	70
10	160	150	140	130	120	110	100	90	80	70	80
11	170	160	150	140	130	120	110	100	90	80	90
12	180	170	160	150	140	130	120	110	100	90	100
13	190	180	170	160	150	140	130	120	110	100	110

inicio, siendo el punto más lejano el que obtiene el mayor valor decimal (Fig. 16 y tablas 3 y 4).

Esto ayuda a que si se tiene varias opciones de desplazamiento el algoritmo seleccione la ubicación donde está el menor valor al cual se puede desplazar, pero este desplazamiento dentro de la matriz no garantiza una ruta entre los puntos de partida y llegada. Por ello se tomó en cuenta el algoritmo de Tremaux en el cual se van marcando las posibles posiciones por donde se podría ir desplazando el prototipo. Si la ruta seleccionada no llega al punto deseado, este regresara a la última posición que cuente con otra posible opción de ruta. Si la nueva ruta marcada no llega, el algoritmo buscara todas las posibles opciones que se puedan llegar a realizar, obviamente con el apoyo de la matriz de prioridades descrita anteriormente, lo que agiliza la consecución de la ruta. Sin embargo, este algoritmo en algunas ocasiones genera rutas no tan eficientes lo que conllevaría a gasto innecesario de desplazamiento y tiempo (Fig. 17).

Sensores y actuadores. El prototipo cuenta con ocho sensores que facilitan el desplazamiento durante el movimiento del modelo para evitar que se choque con obstáculos fijos y obstáculos en movimiento, estos sensores están ubicados en parejas, un sensor ultrasónico acompañado de un sensor láser (Fig. 18).

Para lograr el desplazamiento del prototipo se crearon tres posibles acciones con lo cual se garantiza el movimiento dentro del ambiente controlado. Estas tres funciones son:

- **Avance adelante**: Como inicialmente se planteó que la unidad mínima de avance fuera de 30 cm, se diseñó un

Tabla 3

Matriz representativa punto de inicio y final.

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	1	1	1
2	1	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1100	0.1100	0.1100	0.1100
3	1	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1100	0	50	0
4	1	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.1100	0	0	0
5	1	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0	0	0
6	1	0.1100	0	0	0	0	0	0	0	0	0
7	1	0.1100	0	300	0	0	0	0	0	0	0
8	1	0.1100	0	0	0	0.1100	0.1100	0.1100	0.1100	0.1100	0
9	1	0.1100	0	0	0	0.1100	0.3000	0.3000	0.3000	0.1100	0
10	1	0.1100	0.1100	0.1100	0.1100	0.1100	0.3000	0.3000	0.3000	0.1100	0.1100
11	1	1	1	1	1	1	1	1	1	1	1
12	1	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100

Tabla 4

Matriz representativa de prioridad.

	1	2	3	4	5	6	7	8	9	10	11
1	110	100	90	80	70	60	50	40	30	20	30
2	100	90	80	70	60	50	40	30	20	10	20
3	90	80	70	60	50	40	30	20	10	0	10
4	100	90	80	70	60	50	40	30	20	10	20
5	110	100	90	80	70	60	50	40	30	20	30
6	120	110	100	90	80	70	60	50	40	30	40
7	130	120	110	100	90	80	70	60	50	40	50
8	140	130	120	110	100	90	80	70	60	50	60
9	150	140	130	120	110	100	90	80	70	60	70
10	160	150	140	130	120	110	100	90	80	70	80
11	170	160	150	140	130	120	110	100	90	80	90
12	180	170	160	150	140	130	120	110	100	90	100
13	190	180	170	160	150	140	130	120	110	100	110

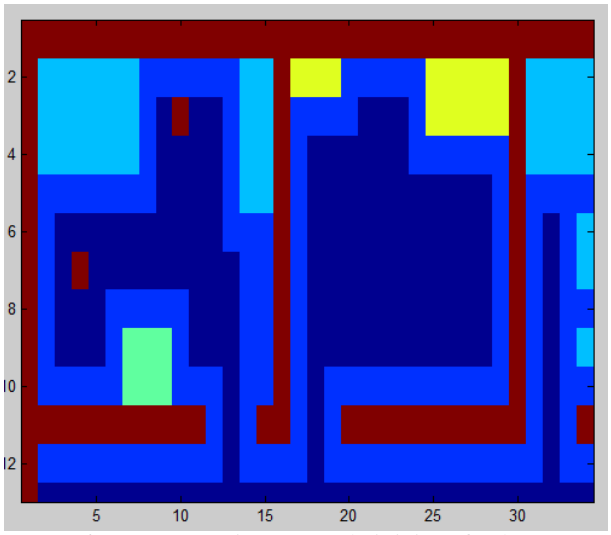

Figura 16. Representación punto de inicio y final en mapa.

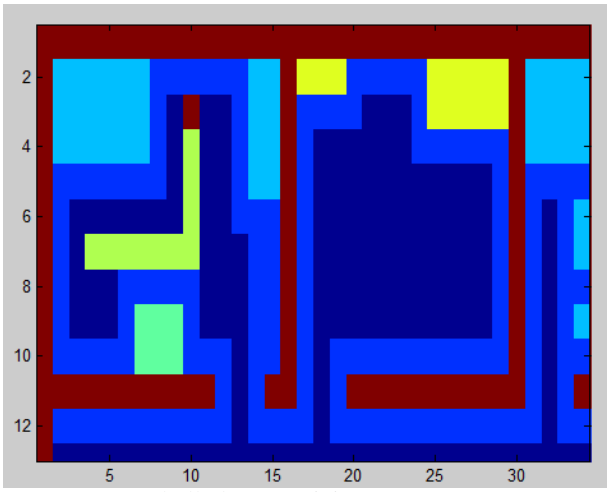

Figura 17. Ruta hallada en posiciones propuestas.

controlador para los motores que desplaza en forma recta al prototipo, esto con la ayuda del encoder sobre cada uno de los motores.

- **Giro derecha e izquierda**: Como el prototipo es de tracción diferencial, los giros se harán sobre el propio eje.

ISSN 1692-8407

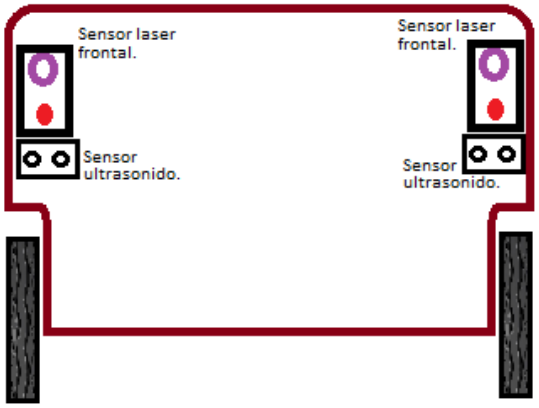

Figura 18. Vista frontal inferior de prototipo.

Una de las llantas gira en un sentido y la otra gira en sentido contrario, esto garantiza que el prototipo no pierda su referencia de ubicación.

Se planteó que el prototipo utilizara tracción diferencial, por ello se utilizaron dos motores muy comunes en el ámbito de vehículos auto propulsados con enfoque médico. Se selecciona un motor sincrónico de imán permanente accionado por una fuente de corriente continua, a 24 V - 7 A, y 10 amperios pico, con un consumo de 180 W. Este motor está acoplado a una caja de velocidades que tiene una relación 32:1 y una llanta de un metro de circunferencia, lo que entrega una velocidad máxima de 6,4 kilómetros por hora. No obstante esta velocidad máxima no será implementada en el prototipo ya que es una velocidad muy elevada para ambientes estrechos. De fábrica tiene ensamblado un freno en el eje superior del rotor el cual fue retirado, y en este lugar se acopló un encoder de 20 muescas por vuelta y un sensor infrarrojo capaz de funcionar a 100 kHz (Fig. 19).

Resultados

Dentro del desarrollo del prototipo se e realizaron dos tipos de pruebas, las simuladas y las físicas. Las simuladas se enfocan en la solución de la ruta en el mapa propuesto, estas se realizaron con el apoyo del software Matlab.

Simulación

Dentro de las pruebas simuladas se ingresa la posición inicial y la posición final por medio del teclado. Luego del primer recorrido solo se ingresara la posición final, como ocurre en el proceso del prototipo real, ya que la posición final se convierte en posición inicial y de nuevo comienza el proceso para hallar la ruta.

Se realizó un gran numero de simulaciones con el algoritmo, el ejemplo que se ilustra a continuación toma como posición inicial, la habitación 1 y como posición final la opción 2 que corresponde al garaje. Por posición definida las coordenadas de inicio son fila 6 y columna 4, las coordenadas de posición final son fila 16 y columna 6 (Fig. 20).

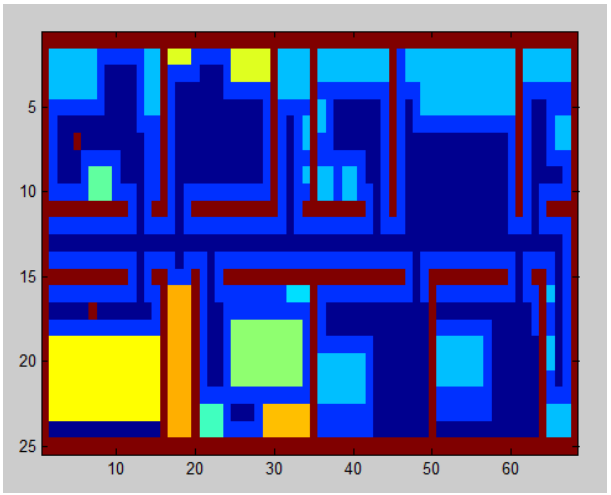

Figura 20. Representación punto de inicio punto final.

En la Fig. 20 se observa los puntos previamente establecidos de inicio y final dentro del mapa. Dentro del proceso se procede a la creación de la ruta opcional número 1, esta ruta se guarda en un vector el cual será comparado posteriormente, se observa que los desplazamientos se realizan en línea recta y con giros de 90 grados.

En este ejemplo se observa que la ruta del punto de inicio al punto final (ruta 1) como la ruta del punto final al punto de inicio (ruta 2) es idéntica, en este caso el algoritmo realiza una redundancia en este proceso. Sin embargo, en algunos casos donde la coordenada del punto de inicio es mayor a la coordenada de punto final, la ruta generada contiene mayores pasos donde el prototipo recorre lugares innecesarios.

Pruebas reales sobre prototipo

En estas pruebas se ingresaron los datos por medio del celular y la aplicación BlueTerm. En esta aplicación se imprime los sitios a donde se puede desplazar el prototipo y solicita el punto de inicio y punto final. Sin embargo en las pruebas realizadas evidencian que la comunicación Bluetooth no se realiza de forma automática con el computador, este necesita de procedimientos externos en las dos terminales para que haya intercambio bidireccional de datos, algo que sería tedioso para el usuario final.

Cada vez que el prototipo inicia su marcha y pide información a los sensores láser, se evidencia un tiempo considerable entre la lectura de la información y la puesta en marcha de los sensores. El tiempo aumenta si el movimiento que debe realizar el prototipo son giros, ya que no solo tendrá

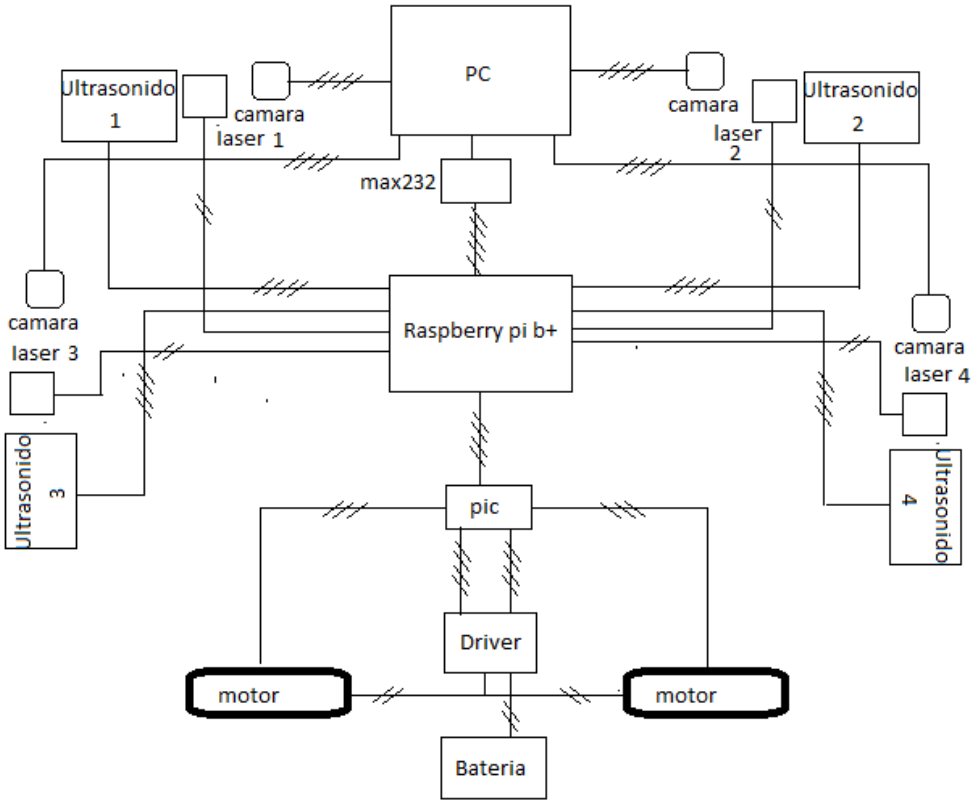

Figura 19. Conexión eléctrica del prototipo.

que sensar dos sensores láser, sino los cuatro disponibles en el prototipo.

Durante el desplazamiento se observó que al prototipo se le debe adicionar un mejor sistema de posicionamiento, ya que el utilizado (odometría) no es el más eficiente en desplazamientos por superficies donde no se controla la fricción, lo que ocasiona que las llantas no siempre se desplacen de acuerdo a lo calculado por los encoders. Si se desactivan los sensores láser y se deja la toma de datos solo a los sensores ultrasónicos el desempeño en tiempo, del robot mejora considerablemente. De hecho, si solo se utilizase sensores ultrasónicos en el prototipo el proceso computacional lo podría realizar la plataforma Raspberry Pi. La Fig. 21 muestra el prototipo final desarrollado.

Figura 21. Prototipo final de laboratorio.

Futuros desarrollos

Ergonomía, el diseño e implementación del prototipo no tuvo en cuenta el aspecto ergonómico, esto ya que no hace parte de la ciencia en estudio. No obstante, es evidente que el prototipo para ser utilizado por una persona que cuente con limitaciones físicas severas necesita apoyos adicionales que le brinden confort.

La representación y adquisición del mapa es otro desarrollo, ya que en el prototipo creado solo se puede introducir la matriz en el programa de una manera manual, lo que implica que muy pocos usuarios tendrán la posibilidad de ingresar sus propios mapas.

ISSN 1692-8407

Conclusiones

El diseño e implementación del prototipo robótico abarca muchas áreas. En este proyecto se tomaron en cuenta las más básicas como son: módulo de desplazamiento, de control, de navegación de rutas, y de sensado, principalmente. Si bien el desarrollo se centró en estos módulos, planteando soluciones finales funcionales de cada uno de ellos, se estima que como primer prototipo aun están susceptibles a muchas mejoras. Un ejemplo es la navegación y posicionamiento, como anteriormente se mencionó, la odometría utilizada aunque es muy eficiente no resulta tan eficaz en terrenos de diversa superficie, algo muy común en los ambientes propuesto de aplicación. En cuanto a los sensores, la aplicación de sensores láser se podría re-evaluar ya que por costos serian inviables, y aunque la elaboración de los mismos con webcam ahorra mucho dinero, estos son muy susceptibles a variables propias del ambiente como luces, reflejo de color rojo en el área de plano focal determinada, entre otros. Un remplazo eficiente podría llegar a ser un sensor ultravioleta de medida de propósito general, los cuales se encuentran fácilmente en el mercado a un precio razonable.

El prototipo brinda un aporte y soluciona una necesidad de una población específica. Esta herramienta tecnológica permitirá una mayor grado de independencia para personas con dificultades de movilidad. El proyecto se puede implementar de forma masiva, ya que es técnicamente y económicamente viable.

Referencias

Barrientos, A., Peñin, L., Balaguer, C., y Aracil, R. (2007). *Fundamentos de róbotica*. McGraw Hill.

Challagundla, M., Yogeshwar, K., y Harsha, N. (2014). Automatic motion control of powered wheel chair by the movements of eye blink. En *International conference on advanced communication control and computing technologies (icaccct 2014)* (p. 1-6).

Crawley, P. (2012). *The brains behind the electric wheelchair, one of canada's 'great artifacts'*. On line. Descargado de `http://www.theglobeandmail.com/report-on-business/small-business/sb-managing/the-brains-behind-the-electric-wheelchair/article4502631/`

Florczyk, S. (2005). *Robot vision: Video-based indoor exploration with autonomous and mobile robots*. Wiley-Vch Verlag.

García, E. (2012). *Visión artificial*. FUOC Fundación para la Universitat Oberta de Catalunya.

Ollero, A. (2001). *Robótica manipuladores y robots móviles*. Marcombo Boixareu Editores.

OMS, y BM. (2011). *Informe mundial sobre la discapacidad.* On line. (Organización Mundial de la Salud y el Grupo del Banco Mundial)

Pololu, P. (2008). *Datasheet ultrasonic distance sensor.* On line. (Parallax Inc.)

Rasbridge, J. (2016). *The power chair: A history.* On line. Descargado de `http://www.powerchairsdirect.co.uk/powerchair%20articles/powerchair-history.html`

Reddy, S., Bhatia, D., y Venkateswara, B. (2016). Design of low-cost manual cum electric-powered wheelchair for disabled person's to use in indoor. En *2nd international conference on next generation computing technologies (ngct 2016)* (p. 1-6).