

Sistema de localización de un robot móvil usando técnicas de visión computacional

Rony Cruz Ramírez

Correo electrónico: rony.crr@gmail.com, rony.cr@ecmtz.cupet.cu
Empresa Comercializadora de Combustibles de Matanzas, Cuba

Artículo Original

Maikel O. Torres Piñero

Correo electrónico: maikel@electronica.cujae.edu.cu

Valery Moreno Vega

Correo electrónico: valery@electronica.cujae.edu.cu
Instituto Superior Politécnico José Antonio Echeverría, La Habana, Cuba

Resumen

La robótica móvil es una temática con múltiples campos de acción, por lo que los estudios en esta área son de vital importancia. En este documento se describe el desarrollo de un sistema de localización de un robot móvil, estimando su posición y orientación mediante visión computacional y el uso de una *webcam*, colocada a una altura en la que se visualiza el entorno de navegación. En la investigación se utilizó un Kit LEGO NXT, a partir del cual se construyó un robot móvil de ruedas de configuración diferencial. Se programó un software en lenguaje C++ y se usaron las librerías de funciones de OpenCV 2.0 para la atención a la *webcam*, el procesamiento de las imágenes capturadas, el cálculo de la localización del robot, el control y la comunicación mediante *Bluetooth*. Además, se implementó un control cinemático de posición y se realizaron varios experimentos con vistas a verificar la fiabilidad del sistema de localización; los resultados de uno de ellos son documentados aquí.

Palabras clave: robótica móvil, visión computacional, LEGO NXT.

Recibido: 10 de noviembre del 2011

Aprobado: 18 de diciembre del 2011

INTRODUCCIÓN

La influencia de los robots en la sociedad actual es cada vez más notable, tanto desde el punto de vista estrictamente social como a los aspectos económicos e industriales. Como consecuencia, la investigación y desarrollo de la robótica es de vital importancia. Dentro de este campo de estudio y aplicación se destaca el tema de la robótica móvil, debido a que los robots móviles cubren una gran variedad de tareas, como: los trabajos subterráneos y submarinos, las misiones espaciales, desactivación de explosivos, operación en zonas radioactivas y aplicaciones militares. En todos los casos la presencia de la robótica es justificada por la dificultad o imposibilidad de intervención humana, bien directa o teleoperada. [1]

El Departamento de Automática y Computación de la Cujae ha realizado varias investigaciones en el tema de la robótica móvil, específicamente sobre generación de trayectorias.

Para ello cuenta con un Kit LEGO NXT, el cual presenta *encoders* en sus motores, lo que permite utilizar la odometría para la estimación de la posición en aplicaciones de navegación. Esta técnica tiene como ventaja su simplicidad y bajo costo; sin embargo, es vulnerable a imprecisiones causadas por el deslizamiento de las ruedas e irregularidades en el suelo, lo que impide obtener con la exactitud requerida la posición del robot [1]. Debido a esto, hasta el momento no se han obtenido resultados reales de los estudios, por lo que es necesario un sistema de localización para el robot móvil que satisfaga los requisitos propios de la aplicación.

En el campo de la robótica móvil, el sistema de localización desempeña un papel primordial debido a la necesidad de estimar de forma fiable la posición de un robot móvil en un entorno de navegación, para su control por realimentación. El tema ha motivado el desarrollo de varias investigaciones [2], [3]; que proponen diversas técnicas para ello. Estas varían

significativamente en función del entorno de navegación, del conocimiento que de él se tenga, de la tarea a realizar y del tipo de sensor con que el robot va equipado o con el que se cuenta; pero una de los más populares por sus ventajas y facilidad es el uso de cámaras de video y algoritmos de visión computacional [1], [4]. Teniendo en cuenta esas características y el poder contar con una *webcam*, se seleccionó esta última técnica, para ser utilizada en la construcción de una plataforma para la navegación del robot.

SISTEMA DE LOCALIZACIÓN DEL ROBOT MÓVIL

Descripción de la plataforma de navegación construida

El desarrollo de esta investigación se basa en la construcción de una plataforma de navegación para el robot móvil LEGO NXT, con una *webcam* para la estimación de su posición y orientación (figura 1).

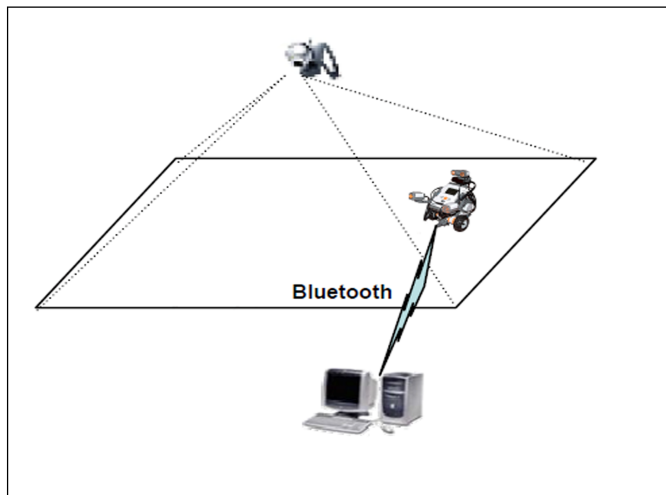


Fig. 1. Esquema de la plataforma de navegación.

Como se observa en la figura 1, el entorno de navegación es un área determinada por el alcance de visión de la *webcam*, instalada a una altura del suelo. La cámara es conectada a una PC mediante USB; de esta forma es posible captar imágenes del área de trabajo, realizar un tratamiento de ellas con algoritmos de visión computacional, obtener la posición y orientación del robot, calcular las señales de control y enviar por *Bluetooth* los comandos directos para la ejecución de las acciones sobre los motores de CD.

Para la construcción de la plataforma de navegación se colocó la *webcam* a una altura de 214 cm. Como resultado, el entorno de navegación del robot, limitado por el área de visión, tiene las siguientes dimensiones: 160 cm de largo x 120 cm de ancho, lo cual da una relación a esa altura de 2px/cm (1cm=2px), teniendo en cuenta que la resolución de la *webcam* es de 320 x 240 px. Esto permitió obtener una relación entre la altura y las dimensiones del área de visión. Así aumentando la primera es posible ampliar el entorno de navegación y conocer la relación px/cm para la nueva configuración, pero como consecuencia se pierde nitidez en

la imagen captada, debido a que la relación se hace menor y puede influir negativamente en el procesamiento de las imágenes.

La *webcam* utilizada es de tecnología CMOS, de reducidas cualidades y su interfaz con la PC es USB. En la conexión se utilizó un adaptador USB - USB de 120 cm que sumado a los 130 cm del cable propio de la *webcam*, da una longitud resultante de 250 cm, que no debe exceder los 500 cm, debido a las restricciones técnicas propias de USB. También es usado un adaptador USB-*Bluetooth*, que es conectado en uno de los puertos USB de la PC para la comunicación con el NXT.

Descripción del robot móvil

En la construcción del robot fue utilizado el kit LEGO MindStorms NXT. Esta herramienta educativa brinda una variada posibilidad de diseños de robots móviles, aunque el caso de interés es el de configuración diferencial, con un prototipo de tres ruedas proporcionado por LEGO: dos de tracción y una rueda de apoyo. [5] Al finalizar su construcción el robot presentó dimensiones de 19 cm de largo, 14 cm de ancho y 14 cm de altura. Las dos ruedas de tracción, de radio 2,8 cm, se montaron en los rotores de dos de los motores del kit.

Al prototipo se le agregaron tres bombillos propios del kit, mediante los adaptadores que este presenta. El primer bombillo se colocó en una extensión realizada con piezas del kit sobre la rueda de apoyo. Mientras que los otros dos se ubicaron en paralelo, en los anclajes, a los lados del ladrillo inteligente y sobre las ruedas de tracción. El objetivo de esta modificación es facilitar la estimación de la posición y orientación mediante el procesamiento de las imágenes tomadas con la cámara en el entorno.

En la configuración de los puertos solo se tomaron en cuenta los tres de salida, ya que no se usó ninguno de los sensores que presenta el kit, solo los actuadores, dos motores y los tres bombillos, que son alimentados por el ladrillo. Para ello se conectaron los motores derecho e izquierdo (mirando desde la rueda de apoyo hacia las de tracción) en los puertos de salida A y C respectivamente. En el puerto B se conectó un adaptador para la alimentación de los bombillos.

Descripción del software de manejo del robot móvil

En la arquitectura del sistema de localización para la navegación del robot móvil es necesario una interfaz de software en la PC para la atención a la cámara, la visión computacional, la implementación de los controladores de posición y la comunicación *Bluetooth* para el envío de comandos al *firmware* del ladrillo inteligente NXT y la ejecución por este de las órdenes de control.

Para la realización del software y de los módulos necesarios, se utilizó el IDE Qt Creator, basado en Qt4, y las librerías de funciones para la visión computacional de OpenCV 2.0. Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario. [6] OpenCV es una biblioteca de visión computacional originalmente desarrollada por Intel. Esta librería provee las funciones de

morfología y atención a la webcam, básicas en la captura y procesamiento de las imágenes. Ambos software son multiplataforma y distribuidos libremente.

Módulo de comunicación *Bluetooth* y control remoto

El ladrillo inteligente LEGO MINDSTORMS NXT se puede comunicar con dispositivos externos que utilizan *Bluetooth* Serial Port Profile (SPP). También es posible enviar comandos directamente al ladrillo NXT desde un dispositivo externo de *Bluetooth* o un ordenador. En el caso de este proyecto se hizo uso de este medio para ejecutar las acciones de control sobre el robot. El software implementado en la PC es el encargado de capturar las imágenes del flujo de video, calcular los valores de velocidad de cada motor y enviar los comandos con las órdenes al LEGO mediante *Bluetooth*. [7], [8], [9].

En el módulo elaborado se programaron funciones para establecer la comunicación *Bluetooth*, PC - LEGO y el control de los motores remotamente a partir del mecanismo de comandos directos. De esta forma es posible establecer la conexión, enviar y recibir mensajes. Para ellos fue necesario predefinir en el sistema un puerto serie (COM) fijo para la conexión *Bluetooth* con el dispositivo NXT y hacer uso de las APIs de Windows para el manejo de la conexión. [10]

Módulo de visión computacional y cálculo de la localización

Un sistema de visión computacional actúa sobre una representación de la realidad que le proporciona información sobre brillo, colores y formas [4], [11]. Estas representaciones suelen estar en forma de imágenes. Una imagen digital es una matriz cuyos índices ($f(x, y)$) identifican un punto en la imagen y la correspondiente matriz de valores identifica niveles de color y brillo. El objetivo del uso de la visión computacional en este caso, está orientado a encontrar las coordenadas espaciales bidimensionales de la posición del robot móvil y su orientación.

El proceso de estimación parte de la adquisición de las imágenes del flujo de video captado por la *webcam*. La captura de cada *frame* permite percibir el entorno de navegación del robot móvil, siempre y cuando este se encuentre dentro del área de visión. Para llevar a cabo este paso se usaron las funciones de la librería HighGUI de OpenCV debido a que esta contempla las funciones de I/O video. [12] Una imagen del robot en el entorno de navegación captada por la *webcam* se muestra en la figura 2.

En el desarrollo de la investigación se tuvieron en cuenta varios mecanismos y metodologías para obtener la posición de un objeto en una imagen, entre ellos se destacan de forma general dos: detección de objetos por segmentación de colores, y por segmentación de límites de región. [4]

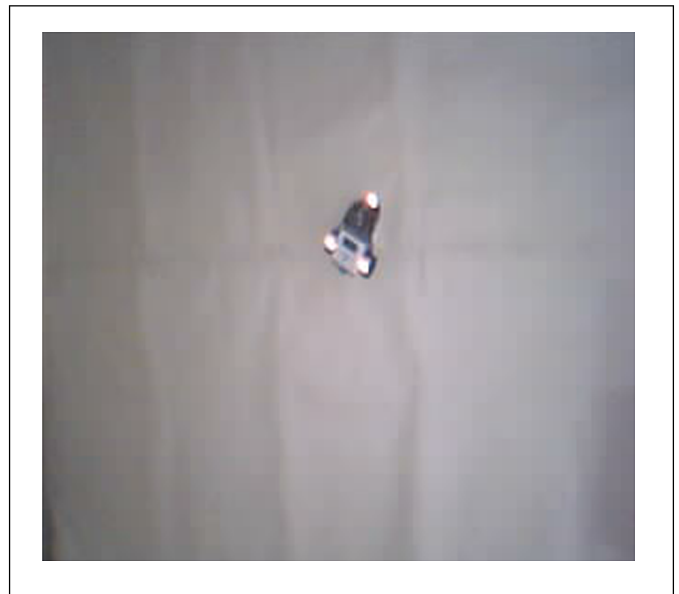


Fig. 2. Imagen del robot captada por la *webcam* en el entorno de navegación.

El primer método es muy utilizado en la robótica móvil, la técnica es idónea sobre todo en entornos donde navegan más de un robot. Esto se debe a que es posible colocar sobre cada uno de ellos señalizaciones de diferentes figuras geométricas o códigos de colores, que permitan identificarlos independientemente para calcular su posición y orientación. Como desventaja presenta la necesidad de un hardware de visión de alta definición. En este proyecto se desestimó esta técnica debido a las dificultades con la resolución y las características de la cámara. La definición obtenida de las imágenes captadas no permitió precisar correctamente los códigos de colores y señalizaciones colocadas en el robot.

Teniendo en cuenta estas dificultades se optó por el segundo método. Esta técnica es bastante usada en la detección y seguimiento de objetos. Consiste en convertir las imágenes a color, en RGB u otro espacio a escala de grises, donde la imagen se representa en niveles de intensidad luminosa. De esta manera es posible realizar transformaciones morfológicas que permitan segmentar el objeto en cuestión y separarlo del resto. [4] Al implementar esta técnica, como consecuencia de los ruidos y la iluminación no uniforme e irregular, se detectaron problemas en el proceso de segmentación, lo que impidió definir correctamente las fronteras del objeto en todas las regiones de la imagen. Como solución se decidió insertar a la arquitectura del robot LEGO NXT señalizadores activos, los bombillos descritos con anterioridad, con lo que se logran tres puntos de alta intensidad luminosa que, al analizarlos, representan los vértices de un triángulo isósceles. Teniendo esta información se pueden calcular la posición y orientación del robot en el entorno. [2]

La metodología diseñada para la segmentación de los bombillos en las imágenes capturadas de la *webcam* tiene el siguiente orden:

1. Convertir la imagen en RGB a escala de grises.
2. Binarizar la imagen en escala de grises.

3. Dilatar la imagen binaria.
4. Calcular los contornos en la imagen y comprobar que sean tres.
5. Separar los contornos en imágenes independientes.
6. Calcular el centro de cada contorno.

Las imágenes en escala de grises muestran niveles de intensidad que se visualizan como una variación de color desde el negro (valor 0) hasta el blanco (valor 255), como se puede ver en la figura 3.

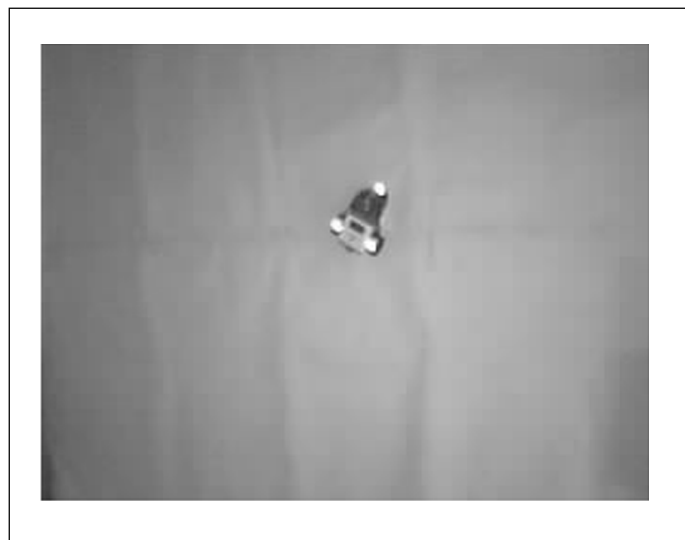


Fig. 3. Imagen en escala de grises.

Luego de obtener la imagen en escala de grises se procede a segmentar los bombillos. Para ello es necesario binarizar la imagen; este proceso consiste en transformar la matriz en escala de grises, donde cada índice puede tener 256 posibles valores a solo 2, blanco o negro. El objetivo en este paso es separar los bombillos del resto del fondo, así se obtiene una imagen con solo dos posibles valores de iluminación, donde los objetos segmentados se muestran en blanco y el resto en negro. [4]

El umbral de la binarización (*threshold*) debe ser tomado como un valor alto, debido a que los bombillos emiten luz y su valor de luminosidad en las imágenes en escala de grises es cercano al máximo; pero es posible que en algunas regiones, como consecuencia de los ruidos y la iluminación no uniforme, los valores de iluminación oscilen por lo que se utilizó un umbral sobre el valor 240. [4]

A la imagen binaria obtenida se le aplica dilatación, que es una de las operaciones de la morfología matemática utilizada en el procesamiento digital de imágenes. Con la dilatación se obtiene una representación de los contornos segmentados, con una mejor definición. [4] Lo que permite calcular los contornos y los centros de estos de forma segura. El elemento estructurante usado para la operación de dilatación fue una elipse con un diámetro de 3px.

Como se observa en la figura 4, solo quedan los tres contornos que ocupan los bombillos en blanco y el resto en negro. Lo siguiente es identificar cada uno de los contornos

y calcular las coordenadas de sus centros. Para llevar a cabo todas estas tareas de procesamiento de la imagen se utilizaron las funciones de las librerías de OpenCV. [12]



Fig. 4. Imagen binaria dilatada.

En el cálculo de los centros de los contornos se hace una transformación de coordenadas en el eje y o en el ancho de la imagen, con el objetivo de correr el centro de coordenadas (0,0), hacia el (0,240). De esta forma se establecen las coordenadas de acuerdo con el marco de referencia global determinado para el robot.

Teniendo las coordenadas del centro de cada contorno, que representan a los tres bombillos, se obtienen las coordenadas de los vértices del triángulo isósceles. Con los puntos obtenidos se calculan las longitudes entre cada uno de ellos de forma consecutiva, coincidiendo la menor con la base del triángulo. El punto medio de la base, que es el centro del eje de las ruedas, es la coordenada de posición buscada. Con esta y el tercer vértice, identificado por no conformar la base y que corresponde al bombillo sobre la rueda de apoyo, se determina el ángulo de orientación del robot, calculando la arcotangente de la mediana a la base formada por ambos puntos.

Módulo de control

El control de un robot móvil en la mayoría de los casos se realiza por realimentación. Uno de los controladores más sencillo y utilizado en el caso de un robot móvil de configuración diferencial, es el control basado en el modelo cinemático. La cinemática, se centra en el estudio del movimiento del robot en función de su geometría. [13] El control cinemático de robots móviles consiste en determinar las acciones del robot necesarias para llevarlo desde su posición actual a una posición final deseada, considerando las velocidades y la orientación.

Un robot móvil de ruedas de configuración diferencial presenta dos ruedas activas, paralelas e independientes y

una tercera rueda pasiva con funciones exclusivamente de equilibrio. El chasis del robot es considerado como un cuerpo rígido con movimiento en el plano horizontal, por la tracción de las ruedas activas, y se describe en términos de velocidad lineal $v(t)$ y la dirección $\theta(t)$, obteniéndose como resultado un desplazamiento lineal del punto medio del eje de las ruedas activas ($v(t)$) y rotacional ($\omega(t)$) del cuerpo del robot sobre este mismo punto. El modelo matemático de este problema cinemático considera las dos variables de control: $v(t)$, $\omega(t)$ y tres variables de estado: la posición y orientación del robot (x, y, θ) y está dado por [3], [13].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

El control de movimiento del robot se realiza proporcionando las velocidades de las ruedas $v_l(t)$ y $v_r(t)$ o equivalentemente $v(t)$ y $\omega(t)$, variables de control [3], [13].

La ley de control considerada [3], consiste en un controlador por realimentación de estados, que es adecuado para el cumplimiento de misiones de navegación como movimiento de un punto a otro. Para su diseño se considera una situación arbitraria de posición y orientación del robot y se predefine un punto objetivo a alcanzar, como se muestra en la figura 5.

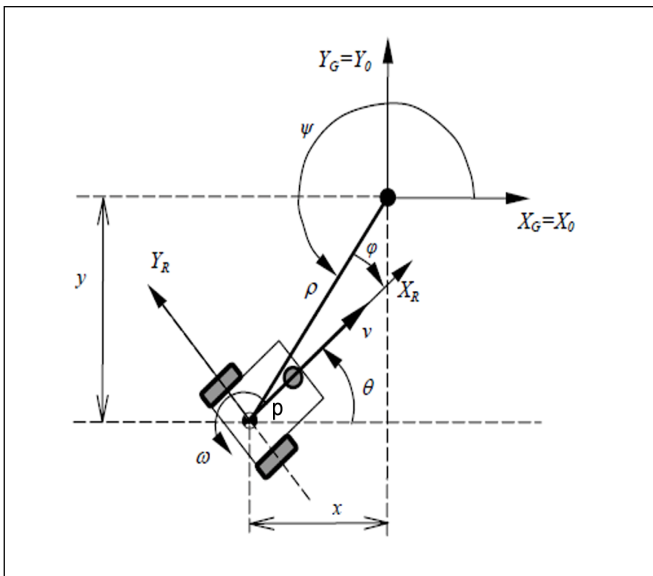


Fig. 5. Situación arbitraria considerada.

Donde se define:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (2)$$

$$\phi = 180 + \theta - \psi \quad (3)$$

En la figura 5, ϕ es el ángulo entre el eje X_R del marco de referencia del robot y el vector que conecta el punto medio en el eje de las ruedas P y el objetivo, este ángulo es definido entre $(\pi, -\pi)$ y determina el sentido de giro del robot en ω o $-\omega$. Mientras que ρ describe la distancia entre P y el objetivo, y ψ el ángulo entre la dirección en línea recta al objetivo y el eje X_G .

Así se establece la ley de control cuyo propósito es determinar las entradas del sistema v y ω [3]:

$$v = k_1 \rho \cos \phi \quad (4)$$

$$\omega = -k_1 \sin \phi \cos \phi - k_2 \phi \quad (5)$$

El movimiento del robot es definido por las velocidades imprimidas por los motores de CD a las ruedas, a partir de las referencias de las velocidades (v_r, v_l) recibidas por *Bluetooth* y que son determinadas mediante las señales de control (v, ω). El esquema del control implementado se puede observar en la figura 6.

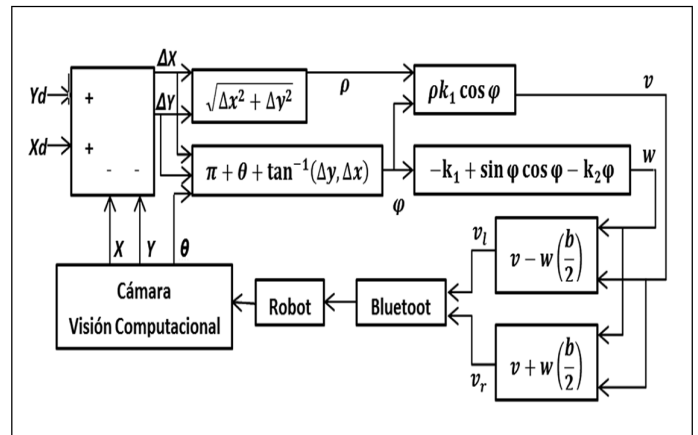


Fig. 6. Esquema del control implementado.

El software implementado es el encargado del muestreo para recibir las imágenes de la cámara, almacenarlas, procesarlas para obtener la posición y orientación, calcular las señales de control y transmitir los comandos al robot. En el software se utilizó una constante de tiempo de muestreo de 150 ms, este valor puede ser menor pero se presentaron retrasos con la cámara usada, debido a que el muestreo asincrónico para obtener los *frames*, impide dos capturas consecutivas en un tiempo menor al utilizado. Al establecer la constante de tiempo en ese valor, el tiempo total utilizado para las tareas de: captura, procesamiento de la imagen, cálculo de las señales de control y envío del comando de velocidad al NXT, es de 14 ms en una PC Intel Pentium Dual - Core. Mientras que el retardo del *Bluetooth* fue de 7 ms.

VERIFICACIÓN DEL SISTEMA DE LOCALIZACIÓN

Para la verificación del sistema de localización se realizaron varios experimentos uno de los cuales es documentado aquí. Los experimentos consistieron en la ejecución de tareas de navegación por el robot y el análisis de los puntos registrados en los recorridos realizados y de los valores de las señales de control calculadas por el controlador en cada muestreo. Luego se realizaron simulaciones del control en MATLAB considerando el modelo cinemático del robot. El ajuste del controlador estuvo dado por ($k_1 = 0,25$, $k_2 = 0,5$). Las gráficas de los experimentos son cotejadas con las obtenidas en las simulaciones.

Para el primer experimento, el robot tuvo una posición inicial en ($X_0 = 45\text{px}$, $Y_0 = 53\text{px}$, $\theta_0 = 99^\circ$) y el destino en ($X_f = 267\text{px}$, $Y_f = 186\text{px}$).

En la gráfica de la figura 7, con color azul y en guiones, el recorrido obtenido por la simulación. Mientras que en rojo los puntos del recorrido real, muestreados por el sistema de localización. Al analizar las dos curvas se observan las diferencias entre el recorrido simulado y el real. Para entender estas diferencias entre las curvas es necesario analizar las señales de control generadas (v , ω), en las figuras 8 y 9 respectivamente.

Como se observa en la figura 8, las curvas de velocidad lineal obtenidas no difieren mucho, sin embargo, resulta claro cómo los valores reales de velocidad superan a los simulados después de los 2 s, debido a esto la curvatura de la gráfica de los valores reales en la figura 7 es menor a la de la simulación, al igual que la distancia recorrida y con un movimiento más cercano a describir una recta. Como resultado se obtiene también el tiempo de movimiento del robot de 15,9 s, que es menor al de la simulación de 23 s.

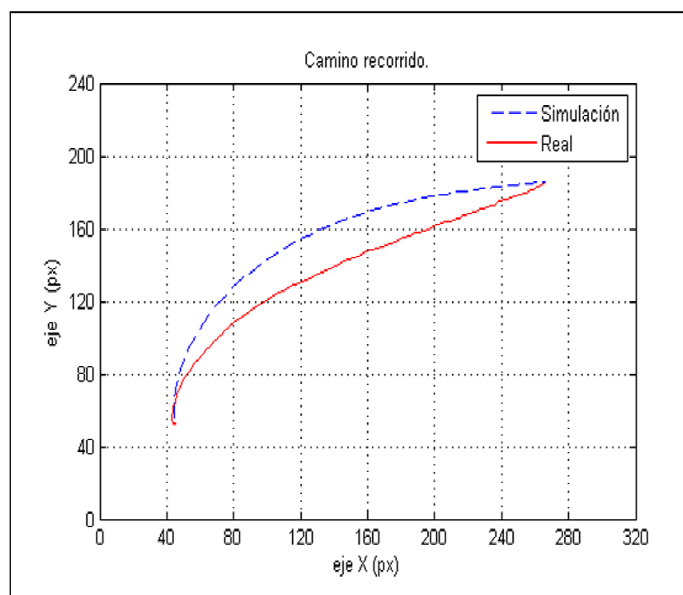


Fig. 7. Gráfica de x vs y , del experimento y la simulación.

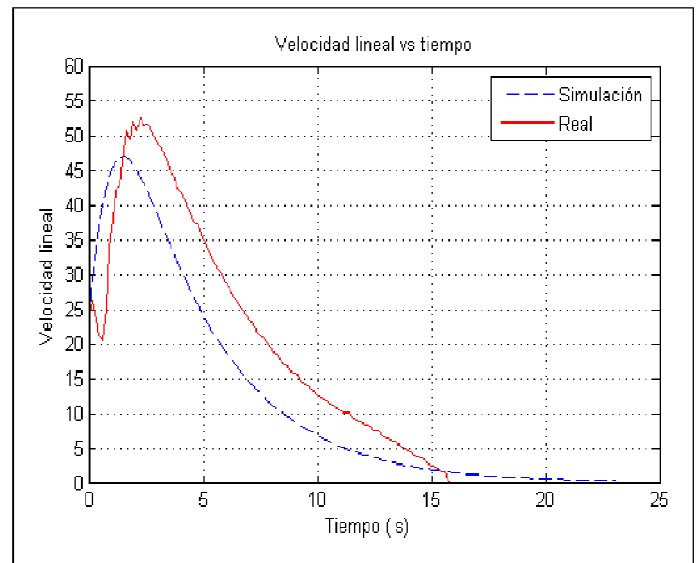


Fig. 8. Gráfica de v vs t , del experimento y la simulación.

En el caso de la velocidad angular ($\omega(t)$), la gráfica de los valores reales sigue más cercanamente a la de la simulación, y aunque sus valores siempre son superiores, el comportamiento del movimiento del robot es más afectado por los valores de la velocidad lineal que por los de la angular, debido a que su magnitud es mayor numéricamente y por lo tanto tiene más definición y protagonismo en el movimiento.

A pesar de las diferencias entre el comportamiento real y el simulado, la tarea de navegación es cumplida, el punto destino que es el objetivo, es alcanzado por el robot, lo que evidencia la fiabilidad de la estimación de la posición y la orientación, que constituyen la realimentación del sistema de control de la posición. Otros experimentos fueron realizados obteniéndose resultados similares.

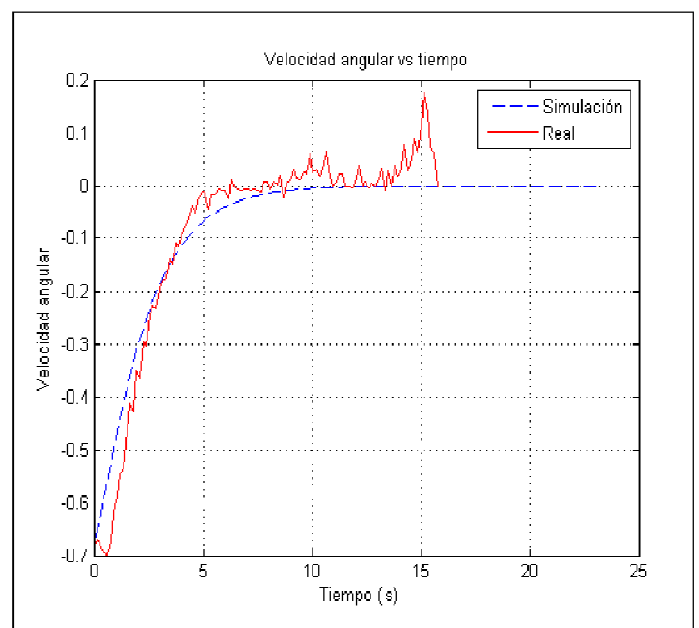


Fig. 9. Gráfica de w vs t , del experimento y la simulación.

CONCLUSIONES

Teniendo en cuenta los resultados obtenidos se pudo constatar que es posible controlar de forma precisa la posición de un robot móvil, y cumplir satisfactoriamente misiones de navegación con el uso del sistema de localización desarrollado. El software implementado es una herramienta necesaria y sirve de base para múltiples aplicaciones.

La baja calidad de la cámara provocó que el sistema siempre fuera perturbado por múltiples ruidos propios del entorno. Una de las mayores perturbaciones fue la iluminación, sus variaciones influyeron notablemente en la segmentación, principalmente en determinadas áreas en las que la incidencia de la luz directamente sobre la superficie del robot creó brillos de muy alta intensidad.

En los resultados, aunque hubo cierta semejanza entre las gráficas de las simulaciones y las reales, se observó cuanto difiere el modelo cinemático ideal de lo que representa realmente el robot, debido fundamentalmente a no tener en cuenta las características dinámicas de este.

Al contar con un sistema de localización fiable, es posible el uso del Kit LEGO en variadas aplicaciones de control, planificación y generación de trayectorias para un robot móvil. Esta investigación sienta las bases a múltiples estudios en el campo de la robótica móvil para el Departamento de Automática y Computación de la Cujae.

REFERENCIAS

1. **GONZÁLEZ, Javier y OLLERO, Anibal.** *Estimación de la Posición de un Robot Móvil* [en línea]. [ref. de 9 de febrero 2011]. Disponible en web: <http://www.robotica-urjc.es/doctorado/gonzalez-estimacion_de_la_posicion-96.pdf>.
2. **DE LUCA, A. et al.** *Visual- Based Planning and Control for Nonholonomic Mobile Robot. Proceedings of the 10th Mediterranean Conference on Control and Automation* [Lisbon, Portugal], 2002.
3. **MARTINS-FILHO, Luiz S. et al.** "Kinematic Control of Mobile Robots to produce Chaotic Trajectories". *ABCM Symposium Series in Mechatronics* 2, 2006, 258-264.
4. **FLORCZYK, Stefan.** *Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robots*. [Darmstadt]: Wiley-VCH Verlag GmbH & Co. KGaA, 2005. 198pp. ISBN: 3-527-40544-5.
5. **ASTOLFO, Dave; FERRARI, Mario and FERRARI, Giulio.** *Building Robots with LEGO MindStorms NXT*. Ed. Dave Astolfo. [Burlington]: Syngress Publishing. Inc., 2007. 447pp. ISBN: 13:9781597491525.

6. **BLANCHETTE, Jasmin and SUMMERFIELD. Mark.** *C++ GUI Programming with QT 4*. Prentice Hall, 2008. 752pp. ISBN-10: 0-13-714397-4.
7. **THE LEGO GROUP.** *LEGO MindStorms NXT Bluetooth Developer Kit*. 2006. Disponible en web: <<http://mindstorms.lego.com>>.
8. **THE LEGO GROUP.** *LEGO MindStorms NXT Direct Commands. LEGO MindStorms NXT Bluetooth Developer Kit*. 2006. Disponible en web: <<http://mindstorms.lego.com>>.
9. **THE LEGO GROUP.** *LEGO MindStorms NXT Communication Protocol. LEGO MindStorms NXT Bluetooth Developer Kit*. 2006. Disponible en web: <<http://mindstorms.lego.com>>.
10. **PETZOLD, Charles.** *Programming Windows*. Microsoft Press, 1998. 1479, pp. ISBN: 1-57231-995-X.
11. **ITO, Daiko.** *Robot Vision Strategies, Algorithms and Motion Planning*. Ed. Daiko Ito. [New York]: Nova Science Publishers, Inc., 2009. 475pp. ISBN: 978-1-61668-981-0.
12. **OpenCV Community.** *OpenCV Documentation*. 2009. Disponible en web: <<http://sourceforge.net/projects/opencv/>>.
13. **NOURBAKHSH, Illah R.; SIEGWART, Roland.** *Introduction to Autonomous Mobile Robots*. Ed. Ronald C. Arkin. [Cambridge, Massachusetts]: The MIT Press, 2004. 321pp. ISBN: 0-262-19502-X.

AUTORES

Rony Cruz Ramírez

Ingeniero en Automática, Empresa Comercializadora de Combustibles de Matanzas (ECCM), Unión Cupet, MINBAS, Matanzas, Cuba

Maikel O. Torres Piñeiro

Ingeniero en Automática, Máster en Informática Industrial y Automatización, Asistente, Departamento de Automática y Computación, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría Cujae, La Habana, Cuba

Valery Moreno Vega

Ingeniero en Máquinas Computadoras, Doctor en Ciencias Técnicas, Profesor Titular, Departamento de Automática y Computación, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría Cujae, La Habana, Cuba

Localization System for a Mobile Robot Using Computer Vision Techniques

Abstract

Mobile Robotics is a subject with multiple fields of action hence studies in this area are of vital importance. This paper describes the development of localization system for a mobile robot using Computer Vision. A webcam is placed at a height where the navigation environment can be seen. A LEGO NXT kit is used to build a wheeled mobile robot of differential drive configuration. The software is programmed in C++ using the functions library Open CV 2.0. this software then soft handles the webcam, does the processing of captured images, the calculation of the location, controls and communicates via Bluetooth. Also it implements a kinematic position control and performs several experiments to verify the reliability of the localization system. The results of one such experiment are described here.

Key words: mobile robotic, computer vision, LEGO NXT