



Lámpsakos

E-ISSN: 2145-4086

lampsakos@amigo.edu.co

Fundación Universitaria Luis Amigó
Colombia

Soto-Montoya, Jonathan Alexander; Gómez-Naranjo, Julio César; Torres-Pardo, Ingrid-Durley

**USO DE LOS CONCEPTOS BÁSICOS DE NXT-G 2.0 EN LA CONSTRUCCIÓN Y
DESARROLLO DE UN ROBOT SEGUIDOR DE LÍNEA**

Lámpsakos, núm. 9, enero-junio, 2013, pp. 51-58

Fundación Universitaria Luis Amigó
Medellín, Colombia

Disponible en: <https://www.redalyc.org/articulo.oa?id=613965330007>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

USO DE LOS CONCEPTOS BÁSICOS DE NXT-G 2.0 EN LA CONSTRUCCIÓN Y DESARROLLO DE UN ROBOT SEGUIDOR DE LÍNEA

USE OF BASIC CONCEPTS NXT -G 2.0 CONSTRUCTION AND DEVELOPMENT OF A ROBOT LINE FOLLOWER

Jonathan Alexander Soto-Montoya, Est.

*Grupo de Investigación SISCO,
Fundación Universitaria Luis Amigó
Medellín, Colombia
jonathan.sotomo@amigo.edu.co*

Julio César Gómez-Naranjo, Est.

*Grupo de Investigación SISCO,
Fundación Universitaria Luis Amigó
Medellín, Colombia
julio.gomezna@amigo.edu.co*

Ingrid-Durley Torres-Pardo, MSc

*Grupo de Investigación GEA.
Institución Universitaria Salazar y Herrera.
Medellín, Colombia.
i.torres@iush.edu.co*

(Recibido el 10-10-2012. Aprobado el 20-12-2012)

Resumen. En este artículo se muestra la estrategia de programación de un seguidor de línea, a partir de la aplicación de fórmulas matemáticas básicas. Las fórmulas son traducidas en un lenguaje gráfico elemental, que hace posible la experimentación de la estrategia, en un robot LEGO MINDSTORM.

Palabras clave: Robótica, programación, fórmulas matemáticas, seguidor de línea.

Abstract. In this paper the strategy of programming a line follower shown, from the application of basic mathematical formulas. The formulas are translated into an elementary graphical language, which enables testing of the strategy, a MINDSTORM LEGO robot.

Keywords: Robotics, programming, mathematical formulas, line follow.

1. INTRODUCCIÓN

Con los grandes avances tecnológicos que se presentan actualmente, se torna importante conocer el concepto y funcionamiento de la robótica, ya que su utilidad y eficiencia se ha visto reflejada en las diferentes áreas de la ciencia y la tecnología, como la medicina, el área industrial, el hogar, educación, seguridad, entre otros. Es por esto que en este artículo se pretenden dar a conocer bases del funcionamiento de esta rama de la tecnología a partir de un ejemplo básico y simple.

En la actualidad existen varias plataformas de robots económicamente accesibles (AvtivMedia's Pionner robot [1], MIT's HandyBoard and Cricket controller cards [2], The LEGO Group's LEGO MINDSTORM [3] son algunos ejemplos. Aunque para el alcance de esta propuesta y por la facilidad de contar con este recurso, se usa el kit de LEGO MINDSTORM Nxt 2.0 [4]. Este kit, como muchos otros, cuenta con su propio entorno de programación, en este caso el entorno de desarrollo Nxt-G 2.0 [5]. El Nxt-G, consta de un lenguaje gráfico de programación [6] [7] [8], que resulta amigable al usuario, razón que justifica, su selección en esta propuesta.

La estrategia de programación que se va a utilizar en este artículo comprende tres etapas principales:

1.1. Una formulación técnica

El objeto principal de esta etapa, es “imaginar, diagramar y/o dibujar” lo que debe ser el robot y lo que deberá hacer el seguidor de línea, para que ambos converjan en una estrategia rápida y eficaz que le permita a un robot seguir un camino demarcado con una línea. Es decir, con la formulación técnica se espera construir el plan o estrategia concebida para obtener el seguidor que se desea y el robot que se requiere.

1.2. Construcción del robot

Teniendo en cuenta la etapa anterior, se construirá el robot requerido, a partir del acoplamiento de las diferentes piezas que contiene el kit de LEGO ya mencionado [9].

1.3. Construcción del software

Utilizando la idea planteada en la formulación técnica definida y del seguidor que se desea, en esta etapa se desarrollará el proceso de traducción de

las fórmulas matemáticas básicas al lenguaje de programación gráfico [10], el cual finalmente será interpretado y ejecutado por el robot anteriormente construido [11].

2. MARCO REFERENCIAL

Aunque ya se ha citado de manera muy breve el objetivo que se persigue en esta propuesta se considera necesario resaltar algunos de los conceptos formales comúnmente aceptados en el área de la robótica, que resultan recurrentes en este trabajo y que permiten dar mayor claridad a este tema.

2.1. Robótica

Es una rama de la tecnología que estudia el diseño y construcción de máquinas capaces de desarrollar tareas cotidianas o no, en las que se necesita el uso de la inteligencia o, simplemente, para eliminar el factor humano. Actualmente, expertos de varios campos trabajan en la investigación de distintos temas de la robótica, esta generalmente tiende a subdividirse, lo que garantiza un desarrollo más veloz [12] [13] [14].

2.2. Robot

Es un ente virtual y mecánico que desempeña tareas controlado automáticamente por una computadora. En la actualidad, existe una gran cantidad de tipos de robots; el tipo que se utilizó para este trabajo es de tipo autónomo, que se caracteriza por su habilidad para producir y ejecutar su propio plan y estrategia de movimiento [15]. Los robots generalmente cuentan con actuadores para mover la estructura mecánica y con sensores para percibir su entorno [16]; para este trabajo, se pretende usar un robot que consta de un sensor de luz o fotoeléctrico [17], el cual cuenta con un componente que genera luz, y otro que responde al cambio de la intensidad de la misma; además, se usarán dos servomotores [18] para que la locomoción [19] sea correcta.

3. ESTRATEGIA DE PROGRAMACIÓN

Cuando se habla de robot seguidores de línea [20], se están citando robots cuyo objetivo consiste en recorrer con la mayor precisión posible una línea marcada en el suelo, sin embargo, la precisión depende de varios factores, entre ellos, de la distinción de colores que realice el robot del ambiente a través de sus sensores [21]. Con el fin de facilitar esta labor, comúnmente se usa una línea negra sobre un fon-

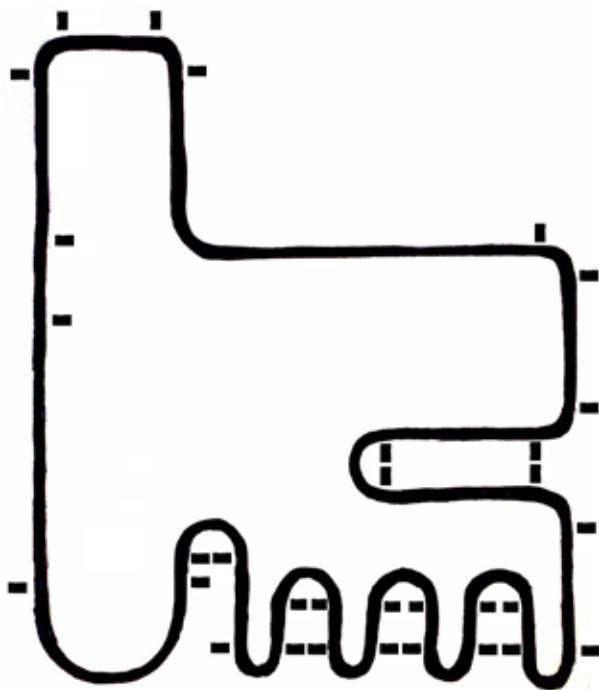


Fig. 1. Pista detallada

do blanco para que exista un mayor contraste. En el presente caso, la línea consta de 1,5 cm de grosor; 11,2 m de longitud aproximadamente y 32 líneas distractoras con el mismo grosor y 4 cm de largo, ubicadas a 1 cm del borde de la línea principal (2 a la izquierda y 30 a la derecha) como puede verse en la figura 1.

Una vez detallado el ambiente de ejecución, se procede a seguir paso a paso, las tres etapas de desarrollo formuladas en la primera sección de este artículo.

3.1 Una formulación técnica

Esta etapa inicialmente, se ocupó de evaluar algunas de las técnicas existentes para seguir una línea. Para ello se consideraron tres técnicas diferentes:

3.1.1 Zigzag

esta idea implementa un sensor de luz [22] apuntando hacia el suelo y utiliza un condicional [23] simple, que dicta que si el sensor de luz está sobre una superficie negra, el robot debe girar hacia la izquierda y posteriormente si se encuentra sobre una superficie blanca, girar hacia la derecha como se puede ver en la figura 2. Con el fin de evitar que el robot se de-

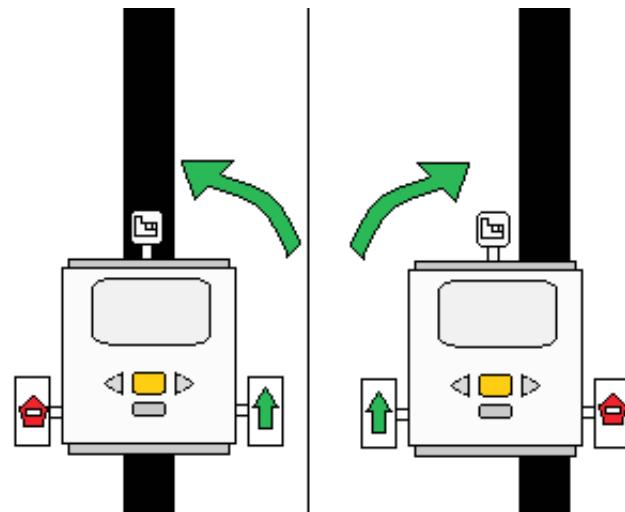


Fig. 2. Imagen del método zigzag

tenga y verifique el proceso de giros anteriormente citado, se han puesto las instrucciones dentro de un ciclo repetitivo infinito [24].

Para la consecución de este método, el robot debe ser diseñado como lo muestra la figura 2, de manera que tenga, como ya se dijo, un sensor (de color o de luz) en la parte de adelante apuntando hacia el suelo, a no más de un centímetro del mismo. El prototipo también debe poseer tres llantas: dos instaladas paralelamente por debajo de la parte trasera del ladrillo y una tercera instalada en la parte inferior delantera del mismo.

3.1.2 Zigzag con 2 sensores

Este método utiliza un sensor de luz y otro de color distribuido por HiTechnic [25], los cuales están uno al lado del otro y apuntan hacia abajo. Se fundamenta en que cuando el sensor de la derecha (sensor de luz) esté en una superficie blanca y el de la izquierda (sensor de color) esté sobre una negra, haga un giro hacia la izquierda; y si ocurre de forma contraria, haga un giro hacia la derecha como puede verse en la figura 3; luego, se vuelve a verificar la condición siempre porque estas instrucciones estarían dentro de un ciclo infinito.

Para la consecución de este método, el robot debe estar diseñado como lo muestra la figura 3, de manera que tenga 2 sensores (de color, luz o su combinación) en la parte de adelante apuntando hacia el suelo, a no más de un centímetro del mismo. Para su

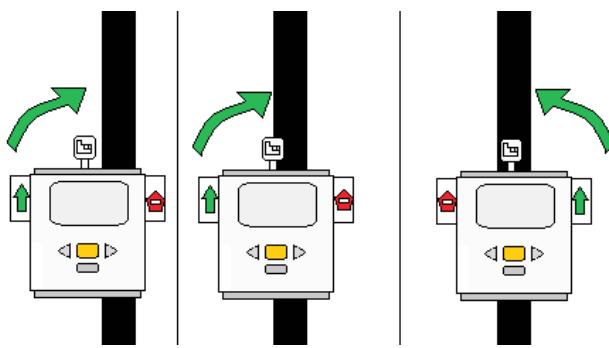


Fig. 3. Imagen del método zigzag con dos sensores

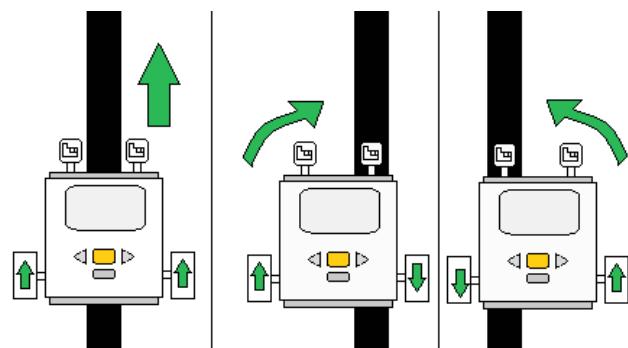


Fig. 4. Imagen del método seguidor de punto intermedio

correcta locomoción, el ladrillo inteligente debe estar equipado con 2 llantas en la parte trasera y moderadamente separadas del mismo y además, debe tener una en el centro de la parte delantera.

3.1.3 Seguidor de punto intermedio

esta forma de interpretar el problema utiliza un sensor de luz al cual se le daría un valor intermedio entre la superficie blanca y la negra, que implementa una condición que dice que si se detecta un valor mayor o igual al definido previamente, significa que está más hacia el blanco que hacia el negro y que debe girar a la derecha y, si al contrario, el valor es menor, es porque está más cerca de la superficie negra que de la blanca, por ende, debe girar hacia la izquierda como se puede ver en la figura 4, comprendiéndose esta condición dentro de un ciclo repetitivo infinito.

Para la consecución de este método, el robot debe ser diseñado como lo muestra la figura 4 de manera que tenga un sensor (de color o de luz) en la parte de adelante apuntando hacia el suelo, a no más de un centímetro del mismo. El prototipo también debe poseer tres llantas: dos instaladas paralelamente por debajo de la parte delantera del ladrillo y una tercera unida en la parte inferior trasera del mismo.

El método lógico escogido fue el seguidor de punto intermedio, puesto que es la idea más simple, versátil y rápida para llevar a cabo la tarea, además utiliza un solo sensor haciendo la tarea mucho más fácil.

3.2 Construcción

Mediante las piezas plásticas y el ladrillo inteligente que contiene el kit de lego MINDTORM Nxt 2.0 y

gracias a su gran versatilidad se han ensamblado las diferentes partes que lo constituyen, como los motores, que con ejes se les instaló una rueda a cada uno de ellos. Estas están en la parte delantera a 8 cm de separación entre sí, lo cual, en conjunto con las vigas que fueron unidas a la parte de atrás del motor y llegan hasta el suelo, hacen las veces de ruedas traseras ofreciendo una mayor maniobrabilidad, estabilidad y velocidad que un modelo con cuatro ruedas y más compacto en cuanto a diseño se refiere.

Los motores [26] en este diseño están directamente debajo del ladrillo inteligente sin ningún tipo de separación; estos, alimentados con 9 V cada uno y con un consumo de 0,55 A, pueden llegar hasta 117 rpm y las ruedas empleadas constan de unas dimensiones de 5,6 x 2,6 cm (modelo 55976). Estas especificaciones se acomodan a las condiciones del medio, gracias a la tracción que brindan.

El sensor de luz está ensamblado en la parte frontal del robot a 1 cm del suelo para así garantizar una buena lectura de la superficie en la que se encuentra, que en este caso entregará la información necesaria para programar correctamente los motores. Este sensor se encuentra a 2 cm de cada rueda, lo que garantiza una reacción más eficiente de los motores como puede verse en la figura 5.

3.3 Software

Como ya se ha dicho, para llevar a cabo los objetivos de este ejemplo se va a utilizar el entorno de desarrollo gráfico Nxt-G 2.0. Teniendo en cuenta como la base principal a la estrategia elegida (seguidor de punto intermedio), se ha desarrollado un programa que sigue la línea correctamente. Esto se ha logrado añadiendo como instrucción principal una bifurcación o condicional que es una instrucción dada al ladrillo



Fig. 5. Robot terminado

inteligente para que decida qué hacer según el dato de entrada definido previamente que, por ejemplo, puede ser el de una variable, sensor de luz, de sonido, de ultrasonido, etcétera.

El sensor de luz consta de un rango de 0 a 100 para expresar la luminosidad de la superficie que percibe. Por pruebas hechas con dicho sensor en un ambiente medianamente iluminado, se determinó que el color blanco está dado por los valores entre 50 y 65, y el color negro entre 25 y 40; en este caso se necesita que cuando el sensor esté en negro vaya hacia la izquierda y cuando esté en blanco o en el punto intermedio vaya hacia la derecha; por lo tanto, se requiere de un valor intermedio entre el blanco y el negro.

El color negro en su expresión mínima como ya se había dicho empieza en 40 puesto que entre mayor luminosidad, más clara es la superficie, y la expresión mínima del blanco es 50 ya que entre menos luz haya se torna más oscura la superficie; consecuentemente, si el sensor está sobre una parte negra y una parte blanca retornará un valor entre 40 y 50, el sensor lo recibiría como punto intermedio entre los dos. Para determinar exactamente ese punto intermedio se calcula con la fórmula del promedio [27] utilizando los datos definidos previamente:

$$Vi = \frac{N + B}{2} \quad (1)$$

Donde Vi es el valor intermedio, N el color negro en su mínima expresión y B el color blanco en su mínima expresión; ahora, remplazando su valor en (1) nos daría:

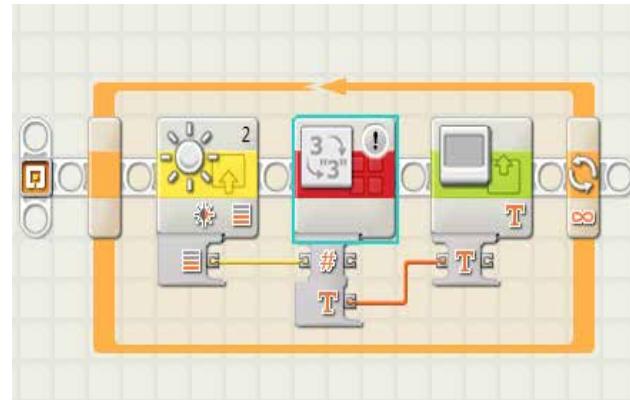


Fig. 6. Ilustración programa verificador de superficie

$$Vi = \frac{40 + 50}{2} = 45 \quad (2)$$

Como se puede ver en (2), el valor intermedio vale 45; para aseverar este dato, se ha verificado y validado mediante un programa que retorna en pantalla el valor de la superficie en la que se encuentra como se ve en la figura 6.

Ahora se crea una bifurcación en el NXT-G 2.0 que comprueba el valor que recibe del sensor de luz; si es mayor o igual que el del punto intermedio, significa que está en una superficie blanca, por ende, debe girar hacia la derecha; para lograrlo, se añaden dos bloques de movimiento, uno que define que se debe mover el motor de la izquierda ilimitadamente a una potencia del 70% (se pone en 70% en vez del 100% para que no se tope con las líneas distractoras), y otro que dicte que el motor de la derecha se detenga. Si, por el contrario, el valor recibido es menor que el del punto intermedio, quiere decir que está en una superficie negra, por lo que debe girar hacia la izquierda; para llevarlo a cabo, se establece con bloques de movimiento de nuevo, que el motor de la izquierda se detenga y el de la derecha avance ilimitadamente con una potencia del 70%.

Para que este proceso no se detenga la primera vez que verifica la condición y, en su lugar, la analice infinitamente, se insertan las instrucciones anteriores dentro de un ciclo infinito como se puede ver en la figura 7.

Cuando el valor verificado es mayor, igual o menor al punto intermedio, mueve infinitamente un motor y detiene el otro según el valor de entrada, pero como

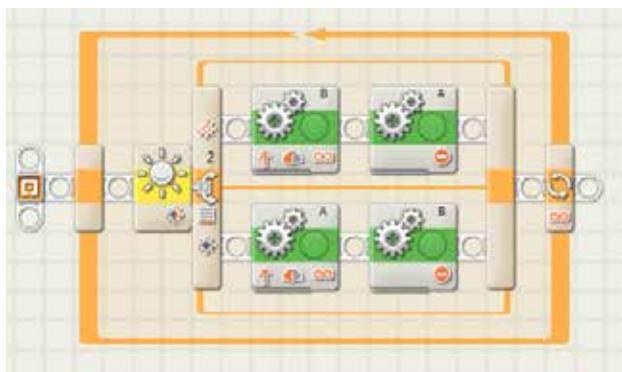


Fig. 7. Ilustración del programa seguidor de línea

está dentro de un ciclo infinito, verificará en fracciones de segundo una y otra vez la condición, y si toma la decisión contraria a la que estaba ejecutando, detiene al motor que estaba en movimiento infinito y mueve el otro; por esto es que es tan necesario el ciclo infinito.

4. RESULTADOS

La línea de 11,2 m fue recorrida por el robot en un tiempo de 143 s. Ahora, con la distancia y el tiempo definidos, se puede calcular la velocidad promedio en la que recorrió la pista con la siguiente fórmula [28]:

$$\text{Velocidad promedio} = \frac{\text{distancia}}{\text{tiempo}} \quad (3)$$

Remplazando los datos ya mencionados (distancia y tiempo) en (3) nos quedaría así:

$$\text{Velocidad promedio} = \frac{11.2 \text{ m}}{143 \text{ s}} = 0,08 \text{ m/s} \quad (4)$$

Con la ecuación (4) se puede ver que la velocidad promedio que empleó el robot en cruzar la pista completamente es de 0,08 m/s (8 cm/s). En el caso de las líneas distractoras, el sensor de luz del robot no las detecta porque hay una distancia de 1 cm entre esta y la línea principal, lo que deja un espacio de color blanco por el que pasa el sensor, sin percibir siquiera algo del negro; además, gracias al punto intermedio definido mediante la operación matemática de promedio y la potencia que se le imprimió (70%), el robot siempre busca estar en la frontera entre la línea negra y la superficie blanca.

El error más común en cuanto a diseño es separar demasiado los motores y, en consecuencia, una rueda de la otra, lo que ocasiona que el radio de giro aumente llevando a que tome la curva mucho más lento y pierda su versatilidad. O del mismo modo, alejarlos demasiado del ladrillo inteligente y del sensor de luz, lo que podría ocasionar una deficiencia en el cumplimiento de los objetivos.

Normalmente se pensaría que implementar una llanta al lado contrario de las ruedas unidas al motor (donde en este caso están las vigas), brindaría una mayor velocidad; lo que es cierto, pero al momento de girar lo hace mucho más ineficiente, pues la misma tracción que brinda para ir más rápido en una línea recta, es la misma que no deja ejecutar el giro apropiadamente en una curva.

Otro error común de diseño es adaptar el sensor de luz a una distancia inadecuada del suelo, ocasionando una lectura incorrecta de los datos. Entre muchos errores de programación que se presentan los más comunes son imprimirle demasiada velocidad a los motores, lo que hará que se tope con las líneas distractoras o que tome la línea en otra posición de la pista. Igualmente a la hora de programar comúnmente no se tiene en cuenta la iluminación del ambiente, lo que puede generar que el robot no actúe de la manera deseada.

5. TRABAJOS FUTUROS

En trabajos posteriores se podría maximizar la eficiencia del robot seguidor de línea haciendo modificaciones en las tres etapas principales mencionadas anteriormente (la estrategia, el software y la construcción). En cuanto a software, se podría remplazar el entorno de desarrollo gráfico Nxt-G 2.0 por uno más completo, como Java LeJOS [29]; y en la parte de construcción, se podría partir de leyes físicas, mecánicas y de otros componentes que no incluye el kit de Lego MINDSTORM Nxt 2.0, y que se adapten mejor a estas leyes para hacer un modelo más eficiente en cuanto a rendimiento.

6. CONCLUSIONES

La introducción o acercamiento a la robótica básica ayuda a desarrollar actitudes y aptitudes referentes a la investigación, la lógica, la física y las matemáticas; lo cual lleva a comprender la importancia de las ciencias básicas, la robótica y su aplicabilidad en la vida cotidiana y procesos que requieren de su uso.

A pesar de las carencias del entorno de desarrollo, Nxt-G 2.0 ofrece una manera de desarrollar aplicaciones de una manera fácil mediante una interfaz gráfica muy intuitiva e interactiva.

Gracias a las facilidades de ensamblaje que ofrece el kit de Lego MINDSTORM Nxt 2.0, la construcción de cada nuevo diseño se torna realmente divertida e interesante; además es muy motivador ver la interacción entre el diseño y el software [30] para cumplir con los objetivos previstos.

El método implementado en este robot seguidor de línea, serviría de base para ser usado en un robot estructuralmente más conciso; el cual podría llevar a cabo tareas de transporte de objetos de un lugar a otro, lo que lleva a optimizar procesos y a economizar costos y tiempo.

Al iniciarse en el mundo de la robótica, es preciso perseverar y no pretender que se solucionen los problemas que aparecen en el camino, puesto que existen demasiadas variables que complican las cosas mucho más (como la cambiante intensidad luminosa del ambiente).

AGRADECIMIENTOS

Brindamos nuestros agradecimientos a todas las personas e instituciones que hicieron posible la realización y culminación de este trabajo. En primer lugar a la Fundación Universitaria Luis Amigó (FUNLAM), sede Medellín, por su apoyo en el desarrollo de este trabajo; a la Universidad Nacional de Colombia, sede Medellín, por permitirnos usar sus instalaciones y herramientas; al director del grupo SINTELWEB, el doctor Jaime Alberto Guzmán Luna por su apoyo y orientación en el desarrollo de este trabajo; al profesor Mauricio López de la Facultad de Ingeniería de la FUNLAM, sede Medellín por su orientación y gran ayuda en el proceso; finalmente a Jhonatan Quejada Mena, estudiante de Ingeniería de Sistemas de la FUNLAM por la compañía en el proceso de desarrollo de este trabajo.

REFERENCIAS

- [1] V. G. Cerf, "What's a Robot," *Communications of the ACM*, vol. 56, No. 1, p. 7, Jan. 2013.
- [2] *The Handy Board* home page [Online]. Available: <http://handyboard.com/>
- [3] *LEGO Mindstorms* home page [Online]. Available: <http://mindstorms.lego.com/>
- [4] J. Floyd Kelly, "Introduction," *Lego Mindstorms Nxt 2.0: The King's Treasure*, 2009.
- [5] J. Floyd Kelly, "Chapter 2," *Lego Mindstorms Nxt 2.0: The King's Treasure*, 2009.
- [6] M. Overmars. (1999). "Programación de Lego robots utilizando NQC," [Online]: Available: <http://www.unlu.edu.ar/~ogarcia/ia/doc/NQCOVSP.pdf>
- [7] M. Boogaarts, Jo. A. Daudelin, B. L. Davis, J. Kelly, D. Levy, L. Morris, F. Rhodes, R. Rodes, M. P. Scholz, C. R. Smith and Rob Torok, "The LEGO Mindstorms NXT Idea Book: Design, Invent, and Build," *Magazine Ubiquity*, vol. 2007.
- [8] M. Boogaarts, Jo. A. Daudelin, B. L. Davis, J. Kelly, D. Levy, L. Morris, F. Rhodes, R. Rodes, M. P. Scholz, C. R. Smith and Rob Torok, "The LEGO Mindstorms NXT Idea Book: Design, Invent, and Build" [Online]. Available: <http://dl.acm.org/citation.cfm?id=1331943&coll=portal&dl=ACM>
- [9] O. Koldo, "Robótica con LEGO MINDSTORMS", *Hispabrick Magazine*, vol. 4, Apr., 2009. [Online] Available: <http://www.hispabrickmagazine.com/content/hbm004-iniciaci%C3%B3n-la-rob%C3%A9tica-con-lego%C2%AE-mindstorms>
- [10] L. Sommerville, "Ingeniería del software", 7th ed. En: M. Martín Romo, ¿Qué es software?, parte I. Madrid: Pearson Educación, 2005.
- [11] C. de Pablos, J. J. López Hermoso, S. Martín Romo and S. Medina, "Capítulo 3," En: Pozuelo de Alarcón (Ed.), *Informática y comunicaciones en la empresa*, Madrid: ESIC, pp. 54-55, 1, abril, 2004.
- [12] J. J. Craig, "Prefacio," En: P. M. Guerrero Rosas (Ed.), *Robótica*, México: Pearson Educación, 2006.
- [13] M. Mataric, "What is a Robot?: Defining robotics". *MIT Press*, pp. 1-6, 2007.
- [14] M. Mataric, "What is a Robot?: Defining robotics" pp. 1-6, 2007. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6291131&queryText%3Drobotics>.
- [15] T. Bajd, M. Mihelj, J. Lenarcic, A. Stanovnik and M. Munih, "Robotics", En: S. G. Tzafestas (Ed.), *Robot vocabulary*, New York: Springer, 2010.

- [16] A. Ollero Baturone, "Capítulo 1: Introducción: Robótica: Manipuladores y robots móviles". En *Robots manipuladores*, España: Marcombo, 2001.
- [17] Consejo Superior de Investigaciones Científicas. "Introducción a los sensores". El Museo Universal, 1987.
- [18] M. Cortés Cherta, "Curso moderno de máquinas eléctricas rotativas", En: *Servomotores de corriente continua*, España: Reverté, 1994.
- [19] T. Jhon, "Principles of Ant Locomotion Could help future robots team work underground," *ACM TechNews* May, 2013.
- [20] J. Floyd Kelly, "LEGO MINDSTORMS NXT-G Program guide", Chapter 2. Laurin Beker (Ed.), 2010.
- [21] Institute of Electrical and Electronics Engineers, Inc., "Robots, visión and signal processing (RVSP)", IEEE Tainan Section International forum, Kaohsiung City, Taiwan, Nov 2011.
- [22] J. Ierache, M. Bruno, M. Dittler and N. Mazza, *Robots y juguetes autónomos una oportunidad en el contexto de las nuevas tecnologías en educación*, Buenos Aires: Universidad de Morón, p. 2, 2008.
- [23] A. Bravo, "Actividades prácticas usando robótica: Condicionales en los diagramas de flujo" [Online]. Available: <http://solorobotica.blogspot.com/2011/09/actividades-practicas-usando-robotica.html>
- [24] J. Paul y H. M. Deitel, *Cómo programar en Java*, 7th ed. México: Pearson Educación, 2008. p. 121.
- [25] *HiTechnic* Home page [Online]. Available: <http://www.hitechnic.com>
- [26] J. Ierache, M. Bruno, M. Dittler and N. Mazza, *Robots y juguetes autónomos una oportunidad en el contexto de las nuevas tecnologías en educación*, Buenos Aires: Universidad de Morón, 2008, p. 2.
- [27] L. Moya de Madrigal, *Introducción a la estadística de la salud*, San José: Editorial de la Universidad de Costa Rica Ciudad Universitaria "Rodrigo Facio", 1986, p. 157.
- [28] R. A. Serway and J. S. Faughn, *Física para bachillerato general*, vol. 1, México: International Thomson Editores, 2006, p. 50.
- [29] G. Ferrari, A. Gombos, S. Hilmer, J. Stuber, M. Porter, J. Waldinger and D. Laverde. "Programming Lego Mindstorms with Java". Group West and Jaguar Book, may 2002.
- [30] L. Sancho Chavarría, "La computadora: recurso para aprender y enseñar", En *Programas de uso común aplicados a la educación*, Costa Rica: EUNED, 1997.