

A MOBILE ROBOT AUTONOMO FOLLOWER LINE

UN ROBOT MÓVIL AUTÓNOMO SEGUIDOR DELÍNEA

Recibido: 26 de noviembre 2013- aceptado: 15 de diciembre 2013

Paola Andrea Galeano.¹

Universidad Nacional de Colombia

Ingrid Durley Torres.²

Institución Universitaria Salazar y Herrera

Juan Felipe Álvarez.³

Institución Universitaria Salazar y Herrera

Keywords:

Robot, mobile, autonomous, robot line follower and precision.

Abstract

This paper describes an application example of an autonomous mobile robot developed for the sole purpose of performing the task of following a path defined by a line plotted on the floor. The detailed construction schedule is based on a previous analysis made under the graphical development environment NXT-G 2.0, taking into consideration the time spent on the course, the precision achieved by following the path and the ability to act autonomously.

Palabras clave:

Robot, móvil, autónomo, seguidor de línea, precisión.

Resumen

Este artículo describe un ejemplo de aplicación de un robot móvil autónomo desarrollado con el fin único de realizar la tarea de seguir una trayectoria definida por una línea graficada en el suelo. El detalle de construcción y programación está basado en un análisis previo logrado bajo el entorno de desarrollo gráfico de Nxt-G2.0, considerando para ello el tiempo consumido en el recorrido, la precisión lograda al seguir la trayectoria y la capacidad de actuar autónomamente.

1. Estudiante Ing. sistemas. Universidad Nacional de Colombia. Medellín, Colombia. E-mail: pagaleanoh@unal.edu.co

2. Docente investigadora. Magister en Ing. Sistemas. Institución Universitaria Salazar y Herrera. Medellín, Colombia.

3. Docente Investigador. Magister en Filosofía. Institución Universitaria Salazar y Herrera. Medellín, Colombia. E-mail: j.alvarezv@iush.edu.co.

*El Modelo Semiautomático de Extracción de Información para el Marcado Semántico de Research Object

I. INTRODUCCION

Con los grandes avances tecnológicos que se presentan actualmente, se hace imperativo conocer el concepto y funcionamiento de la robótica, ya que su utilidad y eficiencia se ha visto reflejada en distintos ámbitos de la vida y diferentes áreas de la ciencia y la tecnología; tal es el de la medicina, la industria, el hogar, la educación, la seguridad, entre otros. Es por esta razón, que en el presente artículo se da a conocer algunas de las bases del funcionamiento de esta rama de la tecnología, a partir de un ejemplo simple, sencillo y de fácil aplicación.

En la actualidad existen varias plataformas de robots económicamente accesibles: AativMedia'sPionner Robot [1], MIT'sHandyBoard and Cricket controller cards [2], Lego MINDSTORM [3]. Por facilidad de acceso, se ha seleccionado el kit de Lego MINDSTORM, el cual se puede programar con el entorno de desarrollo Nxt-G2.0 [4], que viene incluido en el kit y proporciona herramientas básicas para la introducción a esta área mediante una interfaz muy gráfica, o también a través del propio lenguaje Java Lejos [5]. Con la interfaz gráfica sin embargo, es posible antes de construir un robot tangible, modelar un ejemplo previo, que una vez valorado puede ser trasladado al mundo real. Convertir el robot de un ejemplo previo (modelo abstracto) a un robot real, exige la definición de un método básico de tres etapas: la primera, que persigue la definición de la estrategia o técnica que se va a implementar; la segunda, que se ocupa propiamente del proceso de construcción y finalmente, la tercera, que establece el proceso de transcripción de las ordenes al lenguaje de programación comprensible por el robot. Cada etapa es desarrollada considerando bajo la premisa de generar un robot móvil autónomo, es decir un robot que actúe sin la intervención humana continúa. Aunque el ejemplo resulta ser una tarea sencilla, su principal valor radica en la aplicación de un método que no solo logra los objetivos propuestos, sino que puede ser implementado en casos más complejos.

Con el fin de brindar más detalle del caso de aplicación desarrollado, este artículo se organiza como sigue: la sección 2 detalla cada etapa del método propuesto; la sección 3 se ocupa de contextualizar los principales conceptos asociados con el tema de aplicación; la sección 4, implementa una a una las etapas del método pero esta las alternativas que pueden ser valoradas para lograr el objetivo de seguir una línea. Con el ánimo de seleccionar Evaluar alternativas a partir de modelos mentales, que posteriormente serán implementadas y funcionalmente

vez orientadas a la consecución del robot objetivo. La sección 5, recopila los resultados obtenidos y finalmente, en la sección6, se presentan las conclusiones y trabajos futuros.

II. LITERATURA ACTUAL

Antes de comenzar el proceso de desarrollo se hace necesario precisar los principales conceptos que se consideran importantes, para la clara comprensión del objetivo propuesto.

La palabra "robot" proviene de la palabra checa "robota", que significa trabajo. Utilizada en primera medida por el escritor y bioquímico ruso Isaac Asimov, prolífico autor de obras de ciencia ficción, historia y divulgación científica, en los relatos cortos reunidos en su libro I Robot (Yorobot) publicado en 1951 [6]. Desde esa fecha hasta ahora, sobre ese término se han acuñado varios otros, que se añadido como complementos aumentando la complejidad de la definición.

Tales el caso de lo que se ha definido como "robot móvil" y "robot autónomo", con el primero podemos identificar a aquellos robots que pueden desplazarse mientras completan una tarea encomendada, sin colisiones u otros percances. Mientras que por el segundo término, se definen aquellos robots que aceptan una descripción de alto nivel de la tarea a realizar y la ejecutan sin intervención humana [7].

Bajo los conceptos citados, este trabajo propone la construcción de un robot con móvil que realiza de manera autónoma la tarea de seguir una línea definida mediante un color específico en el piso.

III. METODOLOGÍA E INSTRUMENTOS

Una de las características básicas que debe seguir el robot móvil, autónomo, seguidor de línea, está representada en su capacidad de exploración y la habilidad de reconocer su entorno; con tal capacidad y habilidad, el robot no solo percibe los objetos del mundo, sino que a su vez, posibilita la interacción con el mismo. En la robótica lograr tal capacidad y habilidad, es posible a través de unos dispositivos denominados sensores, los cuales permiten advertir el entorno para dar una determinada reacción. En este sentido son variadas y múltiples

la más adecuada, se definen a continuación un conjunto de pasos que permiten, mediante un análisis previo, puestas en operación bajo la construcción de un robot móvil tangible.

Definición de Estrategias

En esta etapa, se elabora la idea y se visualizan las alternativas que permitan al robot seguir una línea lo más rápido, eficaz y preciso posible. Para ello, se valoran tantas propuestas como dispositivos se dispongan.

Zigzag

Esta idea implementa un sensor de luz [8,9], apuntado hacia el suelo y utiliza un condicional [10] muy simple, que dicta si el sensor de luz está sobre una superficie negra, para lo cual le exige que gire hacia la izquierda y posteriormente si se encuentra sobre una superficie blanca, gire hacia la derecha como se puede ver en la Fig. 1

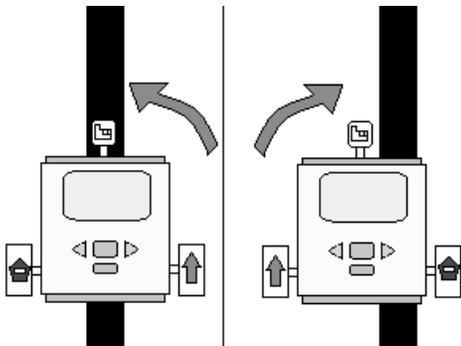


Figura 1. Imagen del método zigzag.

a fin de no detener la operación del robot, al verificar el condicional la primera vez, se debe indicar que el ciclo se repita infinitamente, para lo cual se define un ciclo repetitivo infinito, como se ve en [11].

Zigzag con 2 sensores

Este método utiliza un sensor de luz y otro de color distribuido por HiTechnic [12]. Ambos sensores están ubicados, uno al lado del otro y apuntan hacia abajo; su idea fundamental consiste, en que, cuando el sensor de la derecha (sensor de luz) esté una superficie blanca y el de la izquierda (sensor de color) esté sobre una superficie negra, se realice un giro hacia la izquierda; En caso que ocurra de forma contraria, el sensor de la derecha (sensor de luz) esté una superficie negra y el de la izquierda (sensor de color) esté sobre una superficie blanca, el giro se debe realizar, hacia la derecha como puede verse en la Fig. 2

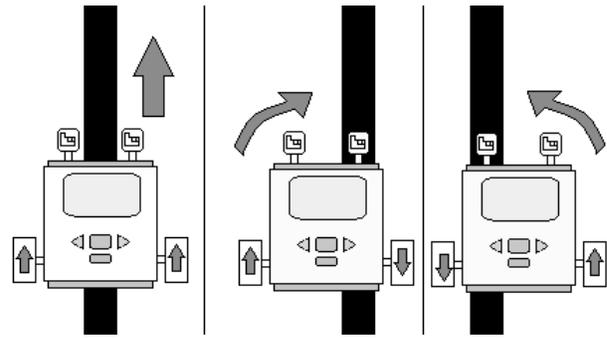


Figura 2. Imagen del método zigzag con dos sensores.

[13]; las anteriores condiciones se verifican contantemente, lo que asegura que estas instrucciones se ubican dentro de un ciclo infinito.

Seguidor de punto intermedio

Esta alternativa, utiliza un sensor de luz al cual se le daría un valor intermedio entre la superficie blanca y la negra [10], a la vez, que implementa una condición que indica, que si tal sensor detecta un valor mayor o igual al definido previamente, significa que el robot está más hacia el color blanco que hacia el color negro de la línea, razón por la cual le exige girar a la derecha; si al contrario, el valor es menor, es porque el robot, está más cerca de la superficie negra que en la blanca, por ende debe girar hacia la izquierda como se puede ver en la Fig. 3.

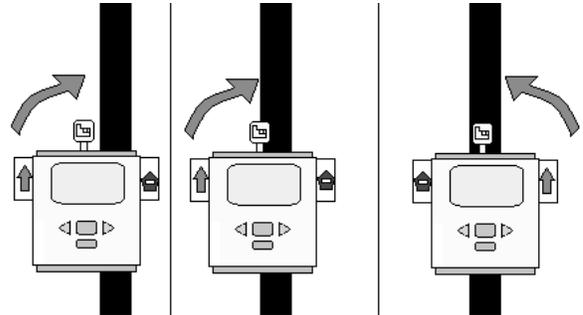


Figura 3. Imagen del método seguidor de punto intermedio.

Como en los casos anteriores, el conjunto de condicionales deben establecerse en un ciclo repetitivo infinito, que asegure la no detención del robot.

La alternativa elegida para el caso de aplicación, acá expuesto, corresponde al seguidor de punto intermedio, ya que con una idea simple, brinda mayor información al robot, el cual puede usar estos datos para facilitar la autonomía y la movilidad con una mayor precisión; además de que evidentemente ahorra piezas (sensores) al hacerlo con un solo sensor, lo que finalmente resulta una tarea más rápida de construir y programar.

Construcción

Teniendo en cuenta la etapa anterior, se construirá un robot que permita ejecutar la tarea de la mejor manera, para hacerlo tangible se usarán las piezas que contiene el kit de LEGO ya mencionado [14].

Mediante las piezas plásticas y el ladrillo inteligente que contiene el kit de Lego MINDTORM Nxt2.0 y gracias a su gran versatilidad se han ensamblado las diferentes partes que lo constituyen; por ejemplo, a cada motor, se le asoció un eje, sobre el que se instaló una rueda en cada punta; estas ruedas se ubicaron en la parte delantera a 8cm de separación entre sí, lo cual en conjunto con los ejes, que fueron unidos en la parte de atrás del motor con unas fichas largas como vigas, que llegan hasta el suelo, haciendo las veces de ruedas traseras, ofreciendo una mayor maniobrabilidad, estabilidad y velocidad que un modelo con 4 ruedas y más compacto en cuanto a diseño se refiere.

Los motores [15] en este diseño, se ubicaron directamente debajo del ladrillo inteligente sin ningún tipo de separación; estos, son alimentados con 9 V cada uno y con un consumo de 0,55A, 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, fue por su parte, 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 Fig. 4.

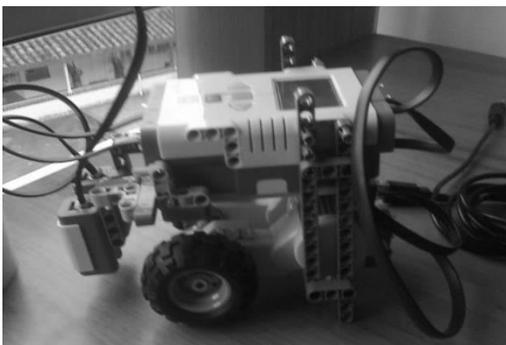


Figura 4. Robot terminado

Software

Como se citó anteriormente, para llevar a cabo los objetivos de este ejemplo, se utilizó el entorno de desarrollo gráfico Nxt-G 2.0, teniendo como base

principal la estrategia elegida (seguidor de punto intermedio). Sobre los condicionales, asociados a esta estrategia, se desarrolló un programa que sigue la línea correctamente; esto se logró añadiendo como instrucción principal una bifurcación o condición al que es una instrucción dada al ladrillo inteligente para que decida que hacer según el dato de entrada definido previamente; por ejemplo, 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 este 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.

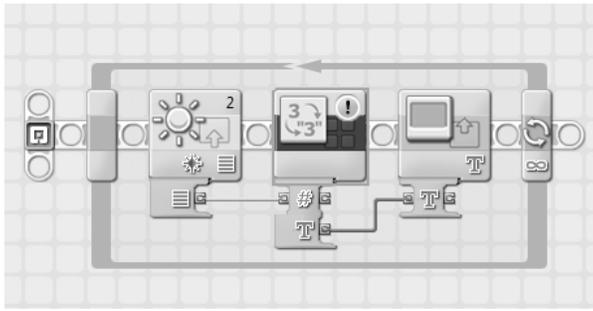
El color negro en su expresión mínima como ya se había dicho empieza en 40, puesto que entre mayor luminosidad más clara es la superficie, y la expresión mínima del blanco es 50 ya que entre menos luz haya se torna más oscura la superficie; por lo tanto si el sensor está sobre parte negra y parte blanca 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 (Craig, 2006) utilizando los datos definidos previamente:

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

Donde V_i 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:

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

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 Fig. 5.

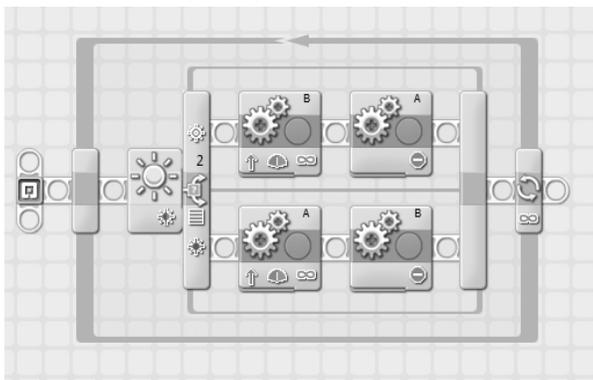


IV.

Figura 5. Ilustración Programa verificador de superficie.

Ahora, se crea una bifurcación en el NXT-G2.0 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 tanto 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 sino que la analice infinitamente, se insertan las instrucciones anteriores dentro de un ciclo infinito como se puede ver en la Fig. 6.



V.

Figura 6. Ilustración del programa seguidor de línea.

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 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.

EJEMPLO BÁSICO

Existen diversos ejemplos para el diseño y diagramación de una línea marcada en el suelo que servirá como ruta para un robot un seguidor [10]. Para este caso específico se utilizará una línea negra sobre un fondo blanco para mayor contraste, esta 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 Fig. 7.

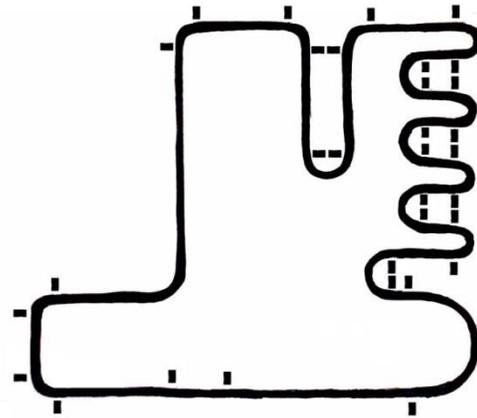


Figura 7. Pista detallada

El robot a utilizar será el Lego MINDSTORMNXT2.0, que será ensamblado teniendo en cuenta dos aspectos principales como lo son la estrategia y el software. Este robot puede ser programado bajo plataformas como: brickOS [11], LeJOS [13], RobotC [12], [15]; o incluso puede ser implementado, bajo el entorno de desarrollo gráfico NXT-G2.0 [5]. Este último elegido para este caso.

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:

$$\text{Velocidad Promedio} = \frac{\text{Distancia}}{\text{Tiempo}} \quad (3)$$

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

$$\text{Vel. Prom.} = \frac{11,2 \text{ m}}{143 \text{ s}} = 0,08 \text{ m/s} \quad (4)$$

Con la ecuación (4) se puede ver que la velocidad promedio que empleó el robot en cruzar la pista

completamente es de 0,08m/s (8cm/s). En el caso de las líneas distractoras, el sensor de luz del robot no las detecta porque hay una distancia de 1cm 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 por el de una rueda de la otra, lo que ocasiona que el radio de giro aumente, conllevando 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, también 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.

VI. TRABAJOS FUTUROS

En trabajos posteriores se podría maximizar la eficiencia del robot seguidor de línea haciéndole modificación es 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; y en la parte de construcción, se podría partir de leyes físicas, mecánicas y de otros componentes que no incluye el kit de Lego MINDSTORMNxt2.0, y que se adapten mejor a estas leyes para hacer un modelo más eficiente en cuanto a rendimiento.

VII. 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 y 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-G2.0 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 MINDSTORMNXT2.0 la construcción de cada nuevo diseño se torna realmente divertida e interesante; además es muy motivador verla interacción entre el diseño y el software 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 iniciar en el mundo de la robótica, hay que perseverar y no pretender que se solucionen fácilmente 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).

VIII. AGRADECIMIENTOS

Este trabajo muestra un resultado parcial asociado con la investigación titulada: "Metodología de Clasificación de Piezas Metálicas y No Metálicas según su forma y textura para selección mediante una banda transportadora"; investigación adscrita a la Institución Universitaria Salazar y Herrera.

REFERENCIAS

- [1] ActivMediaRobotics. 2011. Web site. Recuperado de: <http://www.activrobots.com/>
- [2] The Handy Board. 2009. Homepage. Recuperado de: <http://handyboard.com/>
- [3] LEGO Mindstorms. 2012. Lego Group, Homepage. Recuperado de: <http://mindstorms.lego.com/>
- [4] Floyd, J. 2009. "Lego Mind storms Nxt2.0: The King's Treasure", paperback, Apress; 1 edition, December 1, 2009.

[5] Koldo, O. 2012. "Robótica con LEGO MINDSTORMS", Hispabrick Magazine, Vol. 4, Recuperado de: <http://www.hispabrickmagazine.com/content/hbm004-iniciaci%C3%B3n-la-rob%C3%B3tica-con-lego%C2%AE-mindstorms>

[6] Isaac, A. 1951. I ROBOT. The New York Times. Book Review, ISBN84-270-0906-2. 1951.

[7] Brugali, D. 2007. "Software Engineering for Experimental Robotics". Sidebar-Programming Comercial Robots. Capítulo2.

[8] Bravo, A. 2011. "Actividades prácticas usando robótica: Condicionales en los diagramas de flujo". Recuperado de: <http://solorobotica.blogspot.com/2011/09/activiactiv-practicas-usando-robotica.html>

[9] Overmars, M. 1999. "Programación de Lego robots utilizando NQC". TB Utrecht he Netherlands.

[10] Ierache, J. Bruno, M. Dittler, M. Mazza, N. 2009. "Robots y Juguetes Autónomos una Oportunidad en el Contexto de las Nuevas Tecnologías en Educación" Universidad de Morón, Buenos Aires, Argentina.

[11] Paul, J. Deitel, H. 2008. "Cómo programar en Java", 7th ed. Luis Cruz, M. (Ed.) Pearson Educación, p.121.

[12] HiTechnic. 2001. HiTechnic Products is a division of Dataport Systems, Inc. Homepage. Recuperado de: <http://www.hitechnic.com/>

[13] Ferrari, G. Gombos, A. Hilmer, S. Stuber, J. Porter, M. Waldinger, J. Laverde. 2002. "Programming Lego Mindstorms with Java". Darío Laverde (Ed.) GroupWestand Jaguar Book

[14] Floyd, J. 2010. "LEGO MINDSTORMS NXT-G PrograminGguide", Capítulo 2. LaurinBeker (Ed.) ClayAndres

[15] Craig, J. 2006. "Robots". Tom Robbins (Ed.), Pearson Education pp.339-352, México