



DESAFÍOS Y TENDENCIAS DE LA PROGRAMACIÓN DE ROBOTS EN EL CONTEXTO EDUCATIVO

Flor Ángela Bravo Sánchez, Alejandra María González Correal, Enrique González Guerrero

**Pontificia Universidad Javeriana
Bogotá, Colombia**

Resumen

La robótica en la educación ha mostrado un gran potencial como herramienta de enseñanza y aprendizaje en todos los niveles educativos; desde preescolar hasta el nivel universitario. El uso de robots en el aula de clase permite la generación de ambientes de aprendizaje activos y colaborativos donde los estudiantes pueden construir nuevo conocimiento, desarrollar habilidades y competencias del siglo XXI, y mejorar su experiencia de su proceso aprendizaje. Este uso de la robótica se conoce a menudo con el término de robótica educativa.

Generalmente una actividad de robótica educativa involucra la programación de robots. Este tipo de actividades crea contextos para que los estudiantes, a través de la programación de robots, experimenten y den solución a problemas de la vida real. Los robots le ofrecen al estudiante una realimentación instantánea que los lleva a preguntarse e indagar sobre cómo resolver los problemas que enfrentan con el control de los robots. Esto permite que los estudiantes sean protagonistas de su propio proceso de aprendizaje y alcancen sus objetivos de aprendizaje.

Sin embargo, la programación de robots también se ha convertido en uno de los principales obstáculos en la introducción de la robótica en la educación formal. Programar un robot puede resultar un desafío para estudiantes y profesores no programadores y sin conocimientos de conceptos como sensores, motores, entradas analógicas y digitales, entre otros.

El presente artículo ofrece una corta introducción a la programación de robots y a su potencial como herramienta de enseñanza y aprendizaje. Posteriormente, se presenta

una revisión de las estrategias de programación de robots y de las existentes herramientas para programar robots. Luego, se discute los desafíos y tendencias de la programación con robots en el ámbito educativo. Finalmente, se presentan las conclusiones.

Palabras clave: robótica educativa; programación de robots; software de programación

Abstract

Robotics in education has shown great potential as a teaching and learning tool at all educational levels; from preschool to university level. The use of robots in the classroom allows the generation of active and collaborative learning environments that support knowledge building, develop 21st-century skills and competencies, and enhance the students' learning experience. This use of robotics is often referred to as educational robotics.

Usually, a robot-based activity involves the programming of robots. This type of activities creates contexts that enable students to experiment and solve real-life problems through robot programming. Robots provide the student with instant feedback that leads them to wonder and inquire about how to solve the problems they face with robot control. It allows students to be protagonists of their own learning process and achieve meaningful learning.

However, robot programming has also become one of the main obstacles in the integration of educational robotics into the formal education. Programming a robot could be a challenge for students and teachers without programming skills and prior knowledge of engineering concepts such as sensors, motors, analog and digital inputs, among others.

This article offers a brief introduction to robot programming and its potential as a teaching and learning tool. After that, a review of robot programming strategies and the existing robot programming environments are presented. Then, we discuss the challenges and trends of programming with robots in the educational field. Finally, the conclusions are presented.

Keywords: educational robotics; robot programming; robot programming software

1. Introducción

Los robots proporcionan a los estudiantes de un medio tangible y concreto que facilita la comprensión de una variedad de conceptos e ideas abstractas (Keengwe, 2015). Las actividades de robótica educativa generan ambientes de aprendizaje activos y colaborativos donde los estudiantes pueden adquirir y aplicar habilidades del siglo XXI, tales como la colaboración, la comunicación, la creatividad y el pensamiento

crítico. Adicionalmente, los robots atraen, motivan y mejoran la experiencia de los estudiantes en su proceso de aprendizaje.

Existen actividades de robótica educativa que generan interesantes contextos de aprendizaje a través de la programación con robots. En estas actividades, los estudiantes programan comportamientos en los robots que dan solución a un problema o desafío propuesto. La realimentación instantánea que brindan los robots hace que los estudiantes aprendan de sus errores y refuercen sus acciones correctas. En este artículo se analizan los beneficios que le brinda a los estudiantes crear programas para robots. También presenta una descripción de las estrategias de programación empleadas en la programación de robots educativos. Finalmente, se identifica los retos y tendencias en la programación de robots para propósitos educativos.

La estructura de este artículo es la siguiente: se inicia con una introducción al concepto de programación de robots y a su potencial en los procesos de enseñanza y aprendizaje. Luego, se presenta una revisión de las principales estrategias de programación en el contexto educativo. Posteriormente, se discute los desafíos y tendencias de la programación de robots en el contexto educativo. Finalmente, se mencionan las conclusiones.

2. Introducción a la programación de robots

La programación de robots consiste en el proceso en que el usuario, a través de una interfaz, define los comportamientos que un robot o más robots deben ejecutar sin la intervención humana. El usuario puede crear el programa usando una interfaz gráfica, por comandos de texto, gestos, comandos de voz, entre otros. Luego, el robot usa algoritmos de procesamiento para entender y procesar la entrada del usuario (Crandall, 2014).

Seymour Papert, quien fue uno de los pioneros en promover el aprendizaje a través de la programación, creó el lenguaje de programación llamado *Logo* para el aprendizaje intuitivo de conceptos geométricos (Berg, 2002; Isaías, 2014). *Logo* daba la oportunidad a los niños de crear un programa al mover una tortuga virtual en diferentes direcciones en una pantalla. Luego, este lenguaje de programación fue usado en conjunto con un robot tortuga que tenía montado un lápiz. El robot dibujaba en un papel siguiendo los comandos definidos por los niños en el programa de Logo (Druin, 2000). Papert consideraba que la programación de robots permite a los estudiantes aprender mientras experimentan y reflexionan sobre su experiencia de aprendizaje (Kazakoff, 2013; Papert, 1991).

Otra ventaja de la programación de robots es que permite a los estudiantes la exteriorización de ideas y sentimientos a través de los comportamientos del robot (Keengwe, 2015). Adicionalmente, la programación de robots permite explorar conceptos de causa y efecto en una forma concreta (Bers, 2008).

Una actividad de robótica que envuelve programación facilita el desarrollo de estrategias de resolución de problemas. La programación de robots crea contextos enriquecidos, en los cuales los estudiantes pueden aprender a identificar un problema, encontrar diferentes opciones de solución, seleccionar la mejor solución, generar acciones secuenciales que lleven a la solución del problema. Adicionalmente, la posibilidad de automatizar y hacer tareas simultáneas les permite a los estudiantes proponer soluciones de una forma eficaz (Valero,2016).

A pesar del potencial que tiene la programación de los robots en la educación, se ha convertido en uno de los principales obstáculos para la integración de la robótica en los planes de estudios de los colegios. Programar un robot puede resultar complejo para profesores y estudiantes que carecen de habilidades en programación y no están familiarizados con conceptos de robótica como actuadores y sensores. Existen varias alternativas que buscan facilitar este proceso de programación; por ejemplo, la programación visual, la programación tangible y la programación por demostración. A continuación, detallamos las principales estrategias para crear programas para robots.

3. Estrategias de programación de robots

En el contexto educativo, los robots son programados principalmente a través de técnicas de programación manual tales como programación textual, visual y tangible. Sin embargo, en los últimos años se ha incrementado el interés en el uso técnicas de programación automática tales como la programación por demostración, como una alternativa intuitiva para que usuarios no programadores puedan crear fácilmente programas para robots. A continuación, detallamos cada una de estas estrategias.

Lenguajes de programación textual

En esta aproximación, los usuarios crean el programa del robot en forma de texto. Existen lenguajes de programación textual de propósito general tales como Java, C/C++ y Python que brindan las facilidades para crear códigos para robots. También se han desarrollado lenguajes de programación textual específicos para una plataforma de robótica o aplicación. Estos programas proporcionan simplificaciones que facilitan la programación de robots. Por ejemplo, ROBOTC (<http://www.robotc.net/>) y LeJOS (<http://www.lejos.org/>) son ambientes de programación que facilitan la programación de los robots LEGO Mindstorm. La desventaja de esta opción de programación en el contexto educativo es que requiere que el profesor y los estudiantes de colegio posean experiencia en programación y conocimientos en la sintaxis del lenguaje.

Lenguajes de programación visual

Este tipo de lenguaje permite a los usuarios usar iconos para crear el programa del robot. Se aprovecha de las formas, los colores y los símbolos de los iconos para ayudar a los usuarios a entender la funcionalidad del bloque y a encontrarlos fácilmente.

La mayoría de ambientes de programación que emplean lenguaje de programación visual usan la funcionalidad de arrastrar y soltar (*drag and drop*) los iconos dentro de

un área de programación. Estos iconos pueden ser conectados unos con otros como piezas de rompecabezas. Esta opción es empleada en programas como Scratch (<https://scratch.mit.edu/>), Blockly (<https://developers.google.com/blockly/>) y Mindstorms EV3 (<https://www.lego.com/en-us/mindstorms>). Otra alternativa de conectar iconos es a través de cable de datos, como en el caso de programas como Microsoft VPL (<https://msdn.microsoft.com/>) y Choregraphe (<http://doc.aldebaran.com/1-14/software>).

Existen ambientes de programación de robots que integran la programación visual y la programación textual, tales como Enchanting (<http://enchanted.robotclub.ab.ca/>), Ardublock (<http://blog.ardublock.com/>), MiniBloq (<http://blog.minibloq.org/>), Modkit (<http://www.modkit.com/>). En esta aproximación, los usuarios crean un programa en forma de texto conectado iconos que tienen escrito comandos de texto. Esta opción de programación ayuda a los usuarios a crear programas sin errores de sintaxis.

Programación tangible

Este tipo de programación permite que el usuario crear programas de robots a través de la manipulación de objetos físicos; por ejemplo, bloques o cubos. Cada objeto físico tiene una función específica (por ejemplo, ordenes de movimiento del robot o lectura de los sensores) y la combinación de estos objetos crean el programa del robot. La ventaja de la programación tangible es que permite la programación colaborativa de los robots, a diferencia de los lenguajes de programación que usan una pantalla, en que solo un estudiante puede controlar el teclado o el ratón (Wang, 2016). Cubetto (<https://www.primotoys.com/>), Kibo (<http://kinderlabrobotics.com/kibo/>), y Project Bloks (<https://projectbloks.withgoogle.com/>) son ejemplos de ambientes de programación tangibles.

Programación por demostración

En esta aproximación el robot aprende nuevos comportamientos a partir de una o más demostraciones del comportamiento deseado por parte de un humano u otro robot. El robot deduce automáticamente el programa que le permite reproducir el comportamiento demostrado (Argall, 2009). El robot puede aprender un nuevo comportamiento a través de las siguientes opciones:

- Manipulación física del robot aprendiz.
- Teleoperación del robot aprendiz.
- Manipulación de una representación virtual del robot aprendiz.
- Manipulación de una representación física del robot aprendiz.
- Demostraciones a través del movimiento de un profesor humano o un robot.
- Demostraciones a través del comportamiento de un objeto.

Este tipo de programación provee al usuario de una intuitiva forma de crear comportamientos en el robot, debido a que se asemeja a la forma en que los humanos y animales les enseñan a otros (Fischer, 2016). Un inconveniente de la programación

por demostración es la dificultad de generar automáticamente un programa que el usuario pueda fácilmente entender y modificar.

La programación por demostración ha sido principalmente explorada en el campo de la robótica industrial. Sin embargo, en la actualidad se explora esta estrategia para programación de robots educativos. Software de programación de robots educativos que usan esta aproximación son RoboPlus Motion 2.0 (<http://www.robotis.us/software/>) y Choregraphe.

4. Desafíos y tendencias en la programación con robots en el ámbito educativo

Actualmente las actividades de robótica en el contexto educativo se han enfocado principalmente en el apoyo a los procesos de enseñanza y aprendizaje de áreas tecnológicas como matemáticas, física y programación. Sin embargo, los robots también pueden ofrecer grandes beneficios a las áreas no tecnológicas tales como biología, historia, literatura, entre otras. Uno de los principales obstáculos que tiene el uso de robots en la educación es la programación de robots. Un gran número de profesores de colegio no tienen experiencia en programación, especialmente los profesores de áreas no tecnológicas. Este obstáculo puede ser superado a través de herramientas que permitan a usuarios no programadores crear programas de forma intuitiva para robots.

Existen ambientes de programación que usan los lenguajes visuales, tales como Scratch, Blockly, Lego Mindstorm NXT, que facilitan a profesores y estudiantes la programación de robots. Sin embargo, en la mayoría de casos el uso de estos programas requiere que los usuarios tengan conocimientos básicos en conceptos de programación (por ejemplo, bucles y declaraciones condicionales) y que estén familiarizados con el funcionamiento de motores y sensores. En consecuencia, el proceso de programación puede resultar complejo para usuarios que no cumplan estos requerimientos. Una solución a esta problemática es contar con un ambiente de programación intuitivo que permita crear programas para robots usando un lenguaje de alto nivel como por ejemplo usando verbos; "bailar", "caminar" o "sentarse", en vez de especificar las acciones de adquisición y control (sensores y actuadores). La ventaja de programar con un lenguaje de alto nivel es que el estudiante puede centrarse en identificar los comportamientos que debe tener el robot para dar solución a un problema o reto propuesto.

Una de las dificultades de las herramientas existentes de programación visual es la creación de nuevos comportamientos por parte de usuarios no programadores. Sin embargo, algunas estrategias de programación automática, como la programación por demostración, permiten crear nuevos comportamientos en los robots de una forma intuitiva. La desventaja de la programación automática es la generación de un programa que pueda ser entendido y editado por usuarios en general. Una solución para esta problemática, es contar con un ambiente de programación que integre las estrategias de programación visual y programación por demostración.

Los ambientes de programación también pueden contribuir al logro de los objetivos de aprendizaje de la asignatura. Por ejemplo, el ambiente de programación puede introducir conceptos de música, mientras el estudiante crea los comportamientos de un robot para tocar una melodía. Adicionalmente, el software de programación de robots también puede contribuir a la integración de la robótica en los procesos de enseñanza y aprendizaje de áreas no tecnológicas. Por ejemplo, puede facilitar la implementación de estrategias de enseñanza que son ampliamente usadas por los profesores de áreas no tecnológicas, como *storytelling* robótico o juego de roles con robots actores.

Generalmente las actividades de robótica involucran solo un robot. Sin embargo, el potencial de la robótica educativa se puede incrementar al desarrollar actividades que integren múltiples robots, debido a que existen una gran variedad de temas que son posibles de explicar con más de un robot; por ejemplo, los movimientos de los planetas en el sistema solar, el comportamiento social de las hormigas o los viajes de Cristóbal Colón para el descubrimiento de América. El principal inconveniente en la implementación de actividades que incluyen múltiples robots es la programación simultánea. La mayoría de los ambientes existentes de programación soportan la programación de solo un robot (Jost, 2014). En consecuencia, es importante brindar ambientes de programación que faciliten la programación de múltiples robots, la sincronización de acciones y la gestión de recursos compartidos.

Otro factor importante es el desarrollo de estrategias de programación que soporten el desarrollo de habilidades del siglo XXI como la comunicación y la colaboración. El problema de muchos de los ambientes de programación existentes es que limitan a los estudiantes a un trabajo individual; por ejemplo, solo un estudiante puede controlar el teclado o el ratón del computador. Es importante explorar estrategias que permitan involucrar a más de un estudiante en el proceso de programación.

En los últimos años se ha incrementado el uso de robots sociales en el contexto educativo (Mubin, 2013). Sin embargo, existen pocos ambientes de programación que permitan personalizar fácilmente las características de los robots sociales. Por ejemplo, permitir al usuario crear comportamientos que expresen una personalidad o emociones fácilmente. Es importante ofrecerles a los usuarios herramientas de programación que permitan fácilmente personalizar las expresiones verbales y no verbales del robot social.

Finalmente, un aspecto clave de los ambientes de programación es el tiempo para el desarrollo del programa del robot. Los profesores usualmente tienen tiempo limitado para desarrollar los temas de sus asignaturas. Por lo tanto, es importante ofrecerles alternativas que reduzcan el tiempo de programación de robots.

5. Conclusiones

La exploración de trabajos en robótica educativa, permitió concluir que la programación de robots juega un papel importante en las actividades de enseñanza y

aprendizaje, pues se configura como una herramienta que permite desarrollar diversas competencias y habilidades. En particular, la creación de comportamientos en los robots permite a los estudiantes exteriorizar sus ideas en una forma tangible y explorar comportamientos de causa y efecto en una forma concreta. Adicionalmente, permite el desarrollo de estrategias de resolución de problemas y fomenta el desarrollo del razonamiento secuencial y los pensamientos lógico-matemático, computacional y crítico.

Para incrementar el potencial del uso de los robots en los procesos de enseñanza y aprendizaje se requiere que los ambientes de programación soporten la programación simultánea de múltiples robots, faciliten el trabajo colaborativo, dispongan de alternativas de programación intuitivas, reduzcan el tiempo de programación de un robot y faciliten la implementación de estrategias de enseñanza como drama educativo. La conjugación de estos requerimientos permitiría desarrollar una herramienta versátil que apoyara el proceso de instrucción por parte de los profesores, enfocado a garantizar un proceso adecuado de aprendizaje integral, que articule habilidades sociales y disciplinares.

Actualmente, estamos investigando estrategias de programación de robots que permitan al usuario la creación y puesta en escena de dramatizaciones con múltiples robots actores de forma intuitiva. El desafío de este entorno de programación es la definición de acciones simultáneas de un personaje, por ejemplo, cantar mientras se está caminando. Otro desafío es la programación de varios personajes simultáneamente; por ejemplo, una conversación entre dos personajes y su sincronización.

La búsqueda de una estrategia de programación intuitiva incluye identificar, cuales son los modelos mentales que una persona construye para describir eventos secuenciales y la lógica que utiliza para representar dichos eventos. En este sentido, la aproximación seleccionada es la descripción usando un lenguaje de alto nivel, usando elementos básicos tales como verbos, sujetos, objetos, adjetivos y adverbios. Desde esta perspectiva, programar es contar una historia, definida como una secuencia de eventos y su desenlace. Sin embargo, recurrir solo a palabras puede ser complejo, es por esto que articular esta estrategia con el modo gráfico facilitaría la descripción.

6. Referencias

Artículos de revista

- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5), 469-483.
- Crandall, J. W., Manohar, V., Gezehe, I., Atherton, A., & Goodrich, M. A. (2014). *Principles of Robot Programming Systems*.

- Fischer, K., Kirstein, F., Jensen, L. C., Krüger, N., Kukliński, K., aus der Wieschen, M. V., & Savarimuthu, T. R. (2016, March). A comparison of types of robot control for programming by Demonstration. In Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on (pp. 213-220). IEEE.
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245-255.
- Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A., & Dong, J. J. (2013). A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1, 209-0015.
- Papert, S. and Harel, I. (1991). *Constructionism*. Norwood.
- Valero, A. & Ortega, B. (2016) Programación y Robótica en el aula, ¿cómo y por qué? Revista Ruta Maestra Edición No. 17 ISSN: 2322-7036. Editorial Santillana, Colombia. Pg.117. Disponible en: <http://www.santillana.com.co/rutamaestra/edicion-17/pdf/22.pdf>
- Wang, D., Zhang, L., Xu, C., Hu, H., & Qi, Y. (2016). A Tangible Embedded Programming System to Convey Event-Handling Concept. In Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction (pp. 133-140). ACM.

Libros

- Berg, G. A. (Ed.). (2002). *The knowledge medium: Designing effective computer-based learning environments*. IGI Global.
- Bers, M. U. (2008). *Blocks, robots and computers: Learning about technology in early childhood*. Teacher's College Press, NY, NY.
- Druin, A., & Hendler, J. A. (2000). *Robots for kids: exploring new technologies for learning*. Morgan Kaufmann.
- Keengwe, J. (Ed.). (2015). *Handbook of research on educational technology integration and active learning*. IGI Global.
- Isaias, P. (Ed.). (2014). *Human-Computer Interfaces and Interactivity: Emergent Research and Applications: Emergent Research and Applications*. IGI Global.

Sobre los autores

- **Flor Ángela Bravo Sánchez:** Candidata al doctorado en ingeniería. Ingeniera electrónica con maestría en ingeniería electrónica. Pontificia Universidad Javeriana. bravof@javeriana.edu.co
- **Alejandra María González Correal:** Doctora en ingeniería. Ingeniera Electrónica con maestría en ingeniería electrónica. Profesora asistente de la Facultad de Ingeniería de la Pontificia Universidad Javeriana. agonzalez@javeriana.edu.co
- **Enrique González Guerrero:** Doctor en informática. Ingeniero eléctrico con maestría en ingeniería eléctrica. Profesor titular de la Facultad de Ingeniería de la Pontificia Universidad Javeriana. egonzal@javeriana.edu.co

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2017 Asociación Colombiana de Facultades de Ingeniería (ACOFI)