

# Aplicación de técnicas evolutivas y visión por computadora para navegación autónoma de robots utilizando un TurtleBot 2

Yadira Quiñonez <sup>1</sup>, Iván Tostado <sup>1</sup>, Carlos Burgueño <sup>1</sup>

yadiraqui@uas.edu.mx, itostado@uas.edu.mx, cburguenog@gmail.com

<sup>1</sup> Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, Av. Leonismo Internacional S/N y Av. de los Deportes, Ciudad Universitaria, Mazatlán, Sinaloa, México.

DOI: 10.17013/risti.e3.93-105

**Resumen:** La navegación autónoma de robots es uno de los principales problemas entre los robots debido a su complejidad y dinamismo, ya que depende de las condiciones ambientales como la interacción entre ellos mismo, personas o cualquier cambio sin previo aviso en el entorno. En este trabajo, hemos propuesto la aplicación de técnicas evolutivas para la navegación autónoma de robots. El objetivo de este artículo es desarrollar el control de movimiento de un robot para alcanzar una meta en un entorno desconocido, donde los comportamientos de bajo nivel son obtenidos a través de redes de neuronas artificiales y posteriormente han sido evolucionados mediante algoritmos evolutivos. Para este propósito ROS y el TurtleBot 2 han sido utilizados. El documento finaliza con una discusión crítica de los resultados experimentales.

**Palabras-clave:** Robots autónomos; redes de neuronas artificiales; algoritmos evolutivos; neuro-controladores; ROS.

## *Application of evolutionary techniques and computer vision for the autonomous robots navigation using a TurtleBot 2*

**Abstract:** The autonomous navigation of robots is one of the main problems among the robots due to its complexity and dynamism as it depends on environmental conditions as the interaction between themselves, persons or any unannounced change in the environment. In this work, we have proposed the application of evolutionary techniques for autonomous navigation of robots. The objective of this work is to develop the motion control for a robot to reach a goal in an unknown environment, where the low level behaviors are obtained through artificial neural networks and subsequently have been evolved by evolutionary algorithms. For this purpose ROS and robot TurtleBot 2 are used. The paper ends with a critical discussion of the experimental results.

**Keywords:** Autonomous Robots; Artificial Neural Network; Evolutionary Algorithms; Neuro-controllers; ROS.

## 1. Introducción

La robótica, una de las áreas más características de la Inteligencia Artificial, ha tenido un crecimiento sorprendente desde sus inicios hasta hoy en día, se han desarrollado progresos muy significativos en diversos campos de aplicación. Los robots se han convertido en una herramienta fundamental para producir, trabajar y realizar labores peligrosas en la tierra y fuera de ella. Tradicionalmente, las aplicaciones de la robótica según (Parker, 2008) estaban enfocadas principalmente en el sector industrial (por ejemplo: la soldadura, el montaje, el procesamiento, manipulación de piezas, el corte de materiales mediante robot), donde el principal objetivo era la automatización masiva en los servicios para aumentar la productividad, la flexibilidad, la calidad, pero sobre todo, la mejora de la seguridad con el fin de disminuir el riesgo de las personas en la realización de tareas peligrosas.

En las últimas dos décadas, de acuerdo con (Bräunl, 2008) el campo de aplicación de la robótica se ha ido ampliado a otros sectores, algunos ejemplos son: robots para la construcción (Baeksuk, 2009; Hanjong, 2007), robots domésticos (De Almeida, 2011; Sahin, 2007), robots asistenciales (Linder, 2010; Nagatani, 2009; Santana, 2005), robots en la medicina (Guglielmelli, 2009; Low, 2011), robots de defensa, rescate y seguridad (Okamura, 2010; Reed, 2011), entre otros.

La robótica ha experimentado un rápido desarrollo en el último siglo, ha habido una gran cantidad de investigación realizada con respecto a los robots móviles autónomos (Burgard, 2005; Chaimowicz, 2004; Howard, 2006). Estas investigaciones, se han dirigido hacia la búsqueda de métodos eficientes y robustos para el control de robots móviles. Hoy en día se puede considerar una disciplina científica plenamente establecida, de la que están surgiendo nuevas áreas de conocimiento. Una de estas nuevas áreas se centra en el estudio de sistemas compuestos por varios robots, conocidos como sistemas multi-robot. En general, estos sistemas están formados por un conjunto de robots físicamente independientes, equipados con sofisticados sensores y actuadores que, en un mismo entorno, interactúan entre sí y con el medio para lograr un objetivo en común (Farinelli, 2004), tratando así de mejorar la eficacia, eficiencia, rendimiento y robustez de un sistema robótico unitario.

La visión por computadora tiene una gran cantidad de aplicaciones en diferentes áreas, sin duda, la robótica es una de las más beneficiadas, ya que, hay muchos enfoques para resolver la navegación de robots móviles, una de las técnicas más utilizadas es la de navegación basado en el análisis de imágenes. En el campo de visión por computadora se utilizan diversas técnicas, que contantemente se combinan con algoritmos de aprendizaje automático para obtener mejores resultados. La navegación autónoma de robots basada en la visión por computadora es un recurso maravilloso, porque cualquier información que se pueda extraer de una cámara puede proporcionar una gran ayuda para lograr el movimiento del robot.

En este artículo, proponemos la utilización de técnicas evolutivas y visión por computadora para la navegación autónoma de un robot usando un TurtleBot 2. El objetivo de este trabajo consiste en desarrollar el control de movimiento para que un robot alcance un objetivo en un entorno desconocido. Las variables de entrada y salida del robot se han modelado en una red de neuronas artificiales, y posteriormente, han sido evolucionadas mediante algoritmos evolutivos.

Este artículo está estructurado de la siguiente manera: en la sección II se describen las herramientas de implementación que se utilizarán para llevar a cabo los experimentos.

En la sección III se hace una descripción de los comportamientos básicos implementados tanto en simulación como en el robot. La sección IV presenta una explicación detallada de la evolución de neuro-controladores, se definen los parámetros a considerar para la evolución de dichos comportamientos. En la sección V se muestran los resultados obtenidos de los experimentos que se han realizado a partir de la evolución de un conjunto comportamientos básicos. Finalmente, las conclusiones de este trabajo se describen en la sección VI.

## 2. Herramientas de implementación

ROS (Robot Operating System) es una plataforma de código abierto para el desarrollo de sistemas robóticos. Proporciona toda una serie de servicios y librerías que simplifican considerablemente la creación de aplicaciones complejas para robots. ROS permite el uso de distintos lenguajes de programación. De forma oficial soportan Python, C++ y Lisp además de muchas otras. Actualmente, la librería está dedicada para el sistema operativo Ubuntu (Linux) que es completamente estable, aunque se está adaptando a otros sistemas operativos como Fedora, Mac OS X, Arch, OpenSUSE, Slackware, Debian y Microsoft Windows.

ROS está diseñado para promover la compartición de código y habilitar el desarrollo de aplicaciones de robótica, que permita a la comunidad científica progresar y replicar los resultados de otros grupos de investigación de forma más rápida. En la actualidad, hay diversos grupos que utilizan ROS para manipular a sus robots (Cousins, 2010; Araujo, 2014), algunos ejemplos son: Care-O-bot 3, iRobot Create, Aldebaran Nao y turtlebot 2.

TurtleBot 2 es un robot móvil de cinemática diferencial que está programado con ROS (ver Figura 1) y puede utilizarse para múltiples aplicaciones, es una plataforma robótica abierta que está diseñado específicamente para la educación e investigación.



Figura 1 - TurtleBot 2 equipado con un sensor en 3D, bumpers y cámara Kinect Xbox 360, para navegar en ambientes interiores y exteriores

### 3. Descripción de los comportamientos básicos

Los comportamientos de navegación cubren las habilidades básicas que tiene un robot, utilizando los datos de las características del entorno para llevar a cabo la navegación autónoma. A continuación se describen cada uno de los comportamientos de bajo nivel definidos para lograr el objetivo:

- *Buscando*: este comportamiento hace que el robot navegue por el entorno en busca de un objetivo.
- *Evitando obstáculos*: este comportamiento genera una trayectoria en donde el robot de forma dinámica evita el contacto físico con otros elementos del entorno como los obstáculos, las paredes u otros robots. Utiliza la información de los sensores con el fin de determinar las posiciones de los elementos y generar una velocidad de giro para evitar el posible contacto.
- *Desbloqueando*: mientras que los robots navegan por el entorno, pueden ser bloqueados por obstáculos. El comportamiento de desbloqueo se ha definido para dar solución a esta situación. Básicamente, el comportamiento hace que el robot vaya en dirección opuesta al obstáculo que se ha detectado por el sensor.
- *Siguiendo*: Este comportamiento se encarga de hacer un seguimiento de personas o incluso a otros robots cuando son percibidos a través de la cámara. Un proceso de seguimiento basado en el color del robot se lleva a cabo y en algunos momentos son calculados. El comportamiento trata de mantener la coordenada de la posición horizontal del robot a seguir en el centro de la imagen.

Las redes de neuronas artificiales (RNA) se han estudiado desde los años 60's hasta la actualidad por diversos investigadores de la comunidad científica para resolver problemas en diferentes áreas de aplicación. Básicamente, las RNA como su nombre lo indica, están compuestas por un número de neuronas interconectadas, donde sus parámetros de entrada son el conjunto de señales recibidas del entorno, dichas señales, son calculadas usando una función de activación, y finalmente, se obtiene la señal de salida como se muestra en la Figura 2.

Cada uno de los comportamientos básicos han sido modelados mediante una RNA, donde los parámetros de entrada para el comportamiento de evitando obstáculos es la lectura de los sensores del robot. El parámetro de entrada para comportamiento de seguimiento es la componente  $x$  que retorna la cámara. Finalmente, los parámetros de salida de la red corresponden a la velocidad de rotación del robot, considerando una velocidad de avance constante para cada uno de los comportamientos.

### 4. Evolución de los neuro-controladores

Los algoritmos evolutivos (AE) son técnicas de búsqueda y optimización basadas en procesos naturales de evolución, con el fin de resolver problemas complejos usando modelos computacionales. En la actualidad existen diversos algoritmos evolutivos, los más usados en robótica evolutiva de acuerdo con (Nolfi, 2002) son: Estrategias Evolutivas, Algoritmos Genéticos, Programación Evolutiva y Programación Genética. A pesar de que todos estos fueron desarrollados de forma independiente, comparten el objetivo de imitar los procesos naturales evolutivos. En los últimos años, la

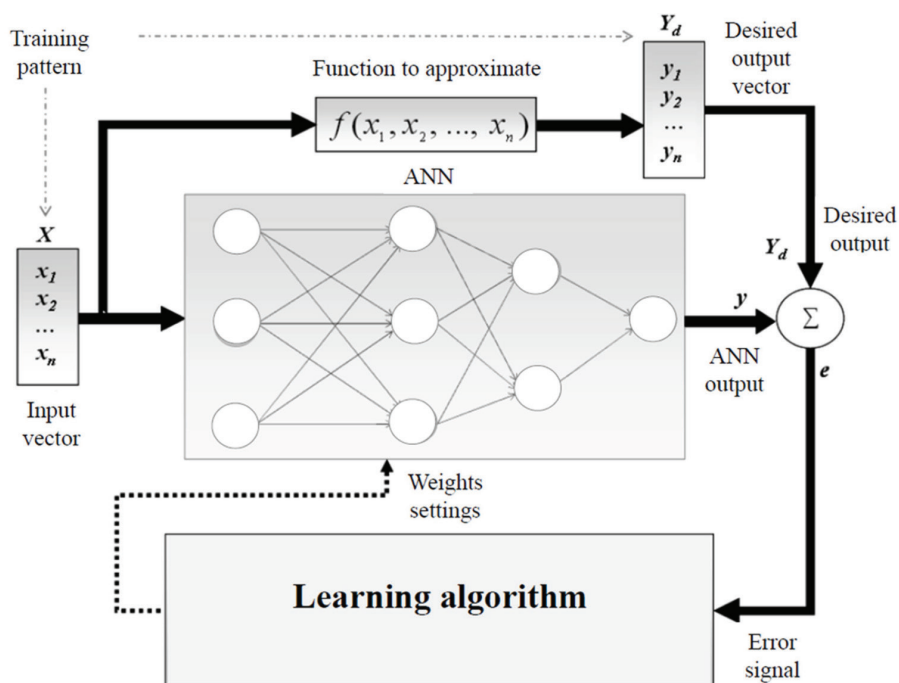


Figura 2 - Esquema general de una red neuronal y diagrama representativo de entrenamiento

aplicación de la evolución artificial en los robots móviles se ha desarrollado muy rápido, específicamente, la evolución de técnicas que les permita adaptar sus comportamientos ante cambios imprevistos, siendo este un desafío clave en la robótica (Sukhatme, 2002).

Las RNAs son evolucionadas a través de algoritmos genéticos, en La Figura 3 se muestra un diagrama de flujo que describe la estructura general del AE. En primer lugar, se inicia con una población que se genera aleatoriamente, después de esto, evaluamos y ordenamos la adaptabilidad (fitness) de cada individuo de la población, se seleccionan los mejores individuos al azar, donde los individuos mejor adaptados tendrán más probabilidad de ser seleccionados. Luego, se aplica el operador de mutación a cada uno de los individuos seleccionados para generar un nuevo individuo y renovar la población. El algoritmo converge cuando el mejor individuo de la población satisface la solución al problema (Quiñonez, 2009).

En las simulaciones hemos considerado algunos parámetros para la evolución de comportamientos: el número de generaciones, el número de individuos, el número de pasos y la probabilidad de mutación. De acuerdo con los resultados obtenidos se requieren alrededor de 40 a 60 generaciones para obtener un controlador que cumpliera con los requisitos; las poblaciones estaban compuestas por 10-40 individuos. Los mejores resultados fueron obtenidos a partir de 40 generaciones con una población de 20 individuos; se obtenía mejor rendimiento usando elitismo porque mantiene la calidad de la población y con un factor de mutación de 0.3 el algoritmo converge más rápido (Quiñonez, 2009).

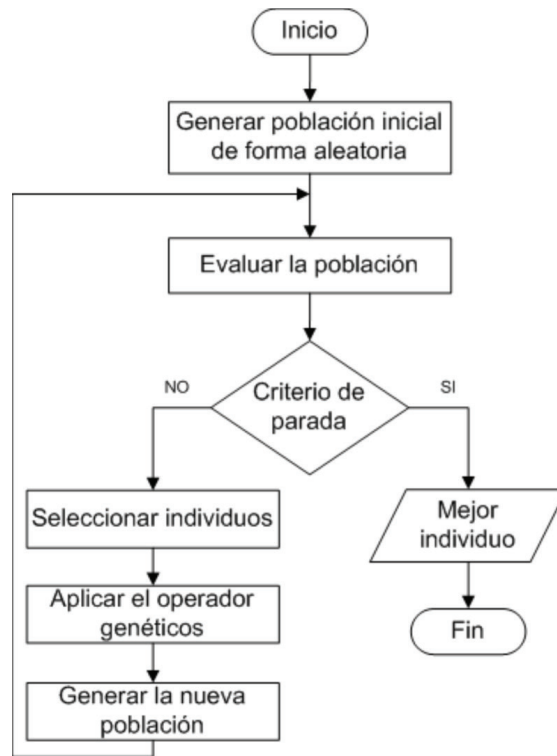


Figura 3 - Algoritmo evolutivo diseñado para evolucionar los comportamientos de bajo nivel

## 5. Resultados experimentales

Hemos llevado a cabo diversos experimentos para evaluar los algoritmos establecidos. La implementación ha sido realizada en Python, la simulación en ROS, y posteriormente, se realizaron las pruebas físicas en el TurtleBot 2.

El primer experimento se ha realizado con poblaciones pequeñas de 20 individuos, y con un número pequeño de generaciones de 50, sin utilizar elitismo y un factor de mutación relativamente bajo (0, 1). En la evaluación de cada individuo se ubicó al robot siempre con la misma orientación y en la misma posición, de tal forma que cada individuo sea evaluado siempre en las mismas condiciones que los otros. La adaptabilidad (fitness) de cada individuo se midió contando el número de paso (steps) que el individuo tomaba en llegar al objetivo, los mejores adaptados tenían un menor número de pasos, de tal forma que la adaptabilidad ( $f$ ) fue calculada de la siguiente manera:

Se realizaron tres ejecuciones, todas ellas con los mismos parámetros, en la figura 4 podemos observar un resumen de los resultados obtenidos. El eje  $x$  corresponde a la generación, y el eje  $y$  al valor del fitness normalizado del mejor individuo de dicha población. Se utiliza el fitness normalizado con el objetivo de visualizar mejor los

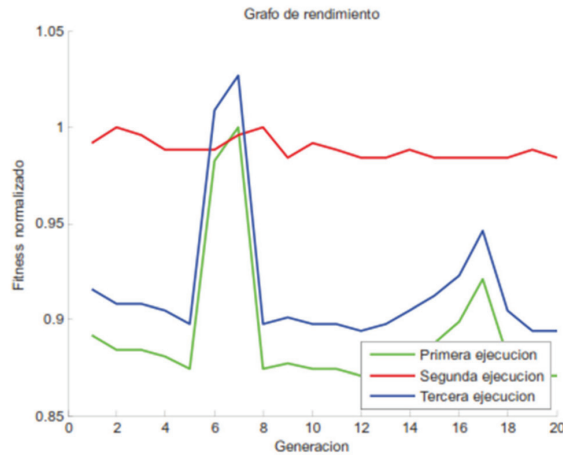


Figura 4 – Índice de rendimento de los resultados del primer experimento

resultados, de tal forma que podamos comparar el mejor individuo de una población con respecto al mejor individuo de entre todas las poblaciones.

Como se observa en la Figura 4 los resultados no son nada prometedores, es una búsqueda totalmente aleatoria. En ciertas poblaciones hemos tenido individuos muy buenos que a pesar de que hayan sido seleccionados para mutar y formar parte de una nueva generación, esta nueva no haya mejorado con respecto a la anterior. Para mejorar estos resultados utilizamos elitismo, esto es, se elige al mejor individuo de una población para colocarlo íntegramente en la nueva población generada a partir de la anterior.

En el siguiente experimento, asignamos el mismo valor colocado a los parámetros del experimento anterior con la diferencia que se ha utilizado elitismo. En la figura 5 se puede observar un resumen de los resultados obtenidos. Analizando los resultados, se puede observar claramente como el elitismo mantiene la calidad de la población. Además, se observa como el mejor individuo de la primera generación fue el mejor de todas las generaciones, es muy probable que en este caso hayamos caído en un óptimo local. Para mejorar esto, hemos aumentado el factor de mutación a 0.3 para ampliar el espacio de la búsqueda.

Para mejorar los resultados utilizamos una función de evaluación de adaptabilidad logarítmica que reaccione mejor ante estos pequeños cambios. En la evaluación de adaptabilidad el robot inicia en una posición y orientación completamente aleatorias, al hacer esta consideración fue necesario realizar un ajuste en la función de adaptabilidad siendo calculada de la siguiente manera:

$$f = \log \left( 1 + \frac{d}{steps} \right)$$

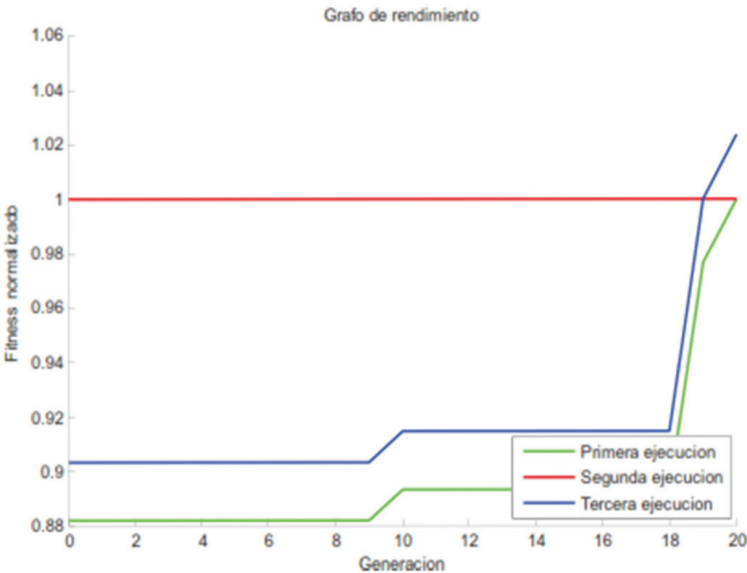


Figura 5 – Índice de rendimiento de los resultados del segundo experimento

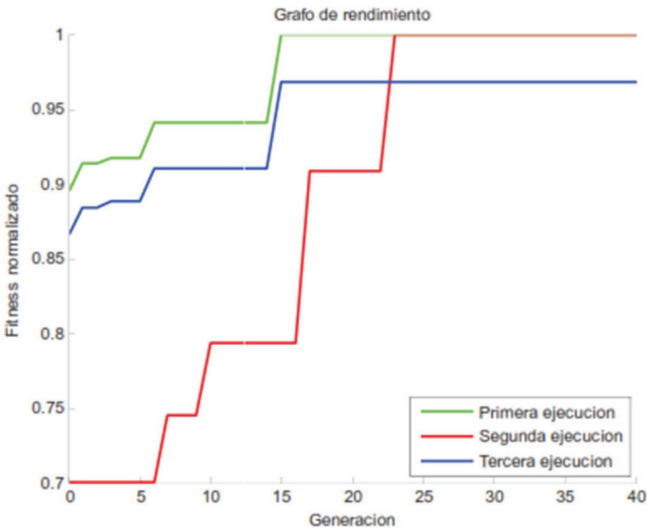


Figura 6 – Índice de rendimiento de los resultados del tercer experimento

Adicionalmente, aumentamos el número de generaciones a 40 para observar mejor la evaluación de las poblaciones utilizando estas mejoras. En la figura 6 se puede observar que la convergencia resulta ser un poco más rápida. En este caso, procedemos a probar los resultados obtenidos seleccionando al mejor individuo de la población final, e introduciendo sus parámetros en el robot para analizar su comportamiento.



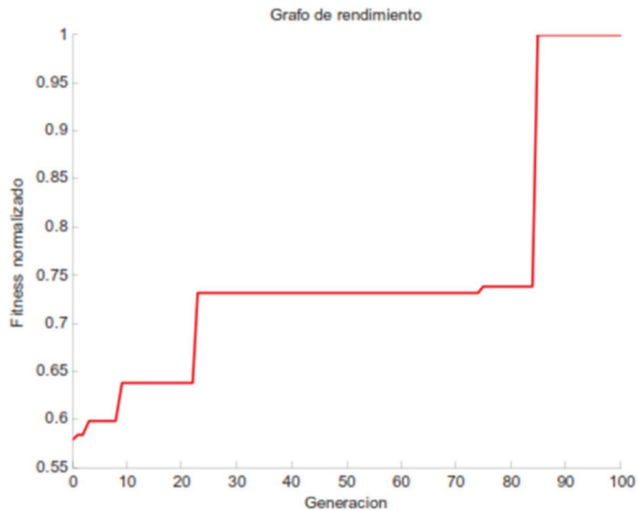


Figura 7 – Índice de rendimiento de los resultados del cuarto experimento

El último experimento realizado considera todas estas mejoras. En la figura 7 se muestran los resultados obtenidos, se puede observar que la convergencia resulta ser lenta, esto es normal debido a que estamos buscando una solución en un espacio de búsqueda mucho más amplio, el cual está considerando la posición y la orientación inicial del robot.

A continuación, en las siguientes figuras se muestran las implementaciones en un robot físico, para probar los resultados de los experimentos con los algoritmos establecidos. En la Figura 8 se observa el entorno en tiempo real en donde el robot realiza la navegación autónoma y la forma en como lo visualiza.

En la Figura 9 y 10 se muestra una secuencia de imágenes con la implementación en el robot, se puede observar claramente como el robot utiliza los datos y características del entorno para llevar a cabo una navegación autónoma con éxito, haciendo uso de los comportamientos básicos establecidos.

## 6. Conclusiones

Los algoritmos evolutivos son técnicas muy potentes para encontrar una solución a un problema de optimización. Aquí se ha demostrado que antes de definir la función de evaluación, es fundamental realizar un análisis exhaustivo del problema a resolver, debido a que, una mala definición de la función de evaluación puede llevar a resultados no satisfactorios. Para conseguir que el algoritmo converja hacia una buena solución en el menor tiempo posible, es imprescindible definir una muy buena función de evaluación que permita reconocer dentro de una población cuales son los mejores individuos.

Se ha verificado que el uso de técnicas evolutivas para evolucionar los neuro-controladores son apropiadas para el desarrollo de comportamientos básicos de los robots. Estas técnicas al ser utilizadas de manera combinada, son un mecanismo potente que permiten



Figura 8 – Visualización del TurtleBot 2 en un entorno físico

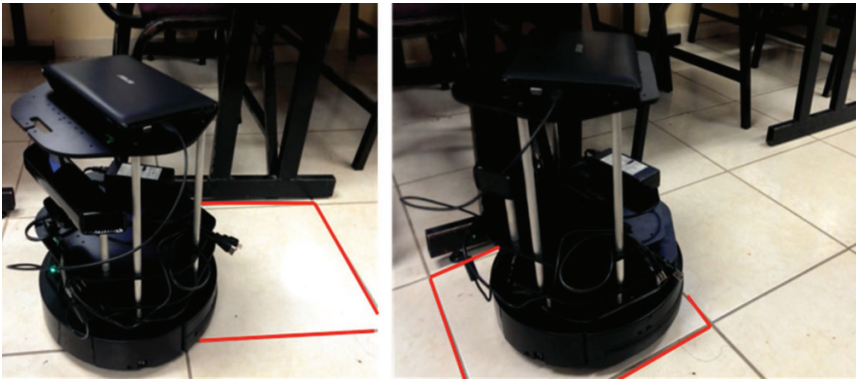


Figura 9 – Implementación en el robot en un entorno físico

suavizar de manera considerable los movimientos del robot, y de esta forma, encontrar no solo comportamientos que optimicen los movimientos de un robot sino también para realizar tareas complejas.

Con respecto a las herramientas de implementación, podemos decir que son herramientas muy completas para el control de dispositivos robóticos, debido a que, soportan una gran variedad de hardware robótico comercial.



Figura 10 – Implementación en el robot en un entorno físico

## Agradecimientos

Los autores quieren dar las gracias a la Universidad Autónoma de Sinaloa por la financiación de este proyecto de investigación con el apoyo otorgado mediante la convocatoria de PROFAPI 2013 con clave PROFAPI/146.

## Referencias

- Araujo, A., Portugal, D., Couceiro M., & Rocha, R. (2014). Integrating Arduino-Based Educational Mobile Robots in ROS. *Journal of Intelligent & Robotic Systems*, 77(2), 281-298.
- Baeksuk , C., Kyungmo , J., Youngsu , C., Daehie , H., Myo-Taeg, L., Shinsuk , P., Yongkwun, L., Sung-Uk, L., Min, C.K. & Kang , H. K. (2009). Robotic automation system for steel beam assembly in building construction. *4th International Conference onAutonomous Robots and Agents* (págs. 38 - 43). IEEE.
- Bräunl, T. (2008). *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*. Springer-Verlag Berlin Heidelberg.
- Burgard, W., Moors, M., Stachniss, C., & Schneider, F. (2005) Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3), 376-386.

- Chaimowicz, L., Grocholsky, B., Keller, J.F., Kumar, V., & Taylor, C.J. (2004) Experiments in multirobot air-ground coordination. *IEEE International Conference on Robotics and Automation*, (4), 4053–4058.
- Cousins, S., Gerkey, B., Conley, K., & Garage, W. (2010). Sharing software with ROS. *IEEE Robotics & Automation Magazine*, 17(2), 12-14.
- De Almeida, A. T., & Fong, J. (2011). Domestic service robots. *IEEE Robotics and Automaton Magazine*, 18(3), 18-20.
- Farinelli, R., Iocchi, L., & Nardi, D. (2004). Multirobot systems: A classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5), 2015-2028.
- Guglielmelli, E., Johnson, M.J., & Shibata, T. (2009). Guest editorial special issue on rehabilitation robotics. *IEEE Transactions on Robotics*, (25), 447-480.
- Hanjong , J., ChiSu , S., Kyunghun , K., Kyunghwan , K., & Jaejun , K. (2007). A study on the advantages on high-rise building construction which the application of construction robots take. *IEEE International Conference on Control, Automation and Systems* (págs. 1933 - 1936).
- Howard, A., Parker, L.E., & Sukhatme, G.S. (2006) Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6), 431–447.
- Linder, T., Tretyakov, V., Blumenthal, S., Molitor, P., Holz, D., Murphy, R., Tadokoro, S., & Surmann, H. (2010). Rescue robots at the collapse of the municipal archive of cologne city: a field report. In *International Workshop on Safety Security and Rescue Robotics*, págs. 1-6.
- Low, K.H. (2011). Robot-assisted gait rehabilitation: from exoskeletons to gait systems. *Defense Science Research Conference and Expo (DSR)*, págs. 1-10.
- Nagatani, K., Okada, Y., Tokunaga, N., Yoshida, K., Kiribayashi, S., Ohno, K., Takeuchi, E., Tadokoro, S., Akiyama, H., Noda, I., Yoshida, T. and Koyanagi, E. (2009). Multi-robot exploration for search and rescue missions: a report of map building in RoboCupRescue 2009. In *International Workshop on Safety Security and Rescue Robotics*, págs 1-6.
- Nolfi, S., & Floreano, D. (2002). Evolutionary robotics: The biology, intelligence and technology of self-organizing machines, *Journal Artificial Life*, MIT Press, Cambridge, MA, USA, pp. 419-424.
- Okamura, A.M., Mataric, M.J., & Christensen, H.I. (2010). Medical and health-care robotics. *IEEE Robotics and Automation Magazine*, 17(3):26-37.
- Parker, L. E. (2008). Multiple Mobile Robot Systems. En B. Siciliano, & O. Khatib, *Springer Handbook of Robotics* (págs. 921-941). Springer-Verlag Berlin Heidelberg.
- Quiñonez, Y., De Lope, J., & Maravall, D. (2009). Cooperative and competitive behaviors in a multi-robot system for surveillance tasks, *Computer Aided Systems Theory, EUROCAST*, pp. 437-444.

- Reed, K.B., Majewicz, A., Kallem, V., Alterovitz, R., Goldberg, K., Cowan, N.J., & Okamura, A.M. (2011). Robot-assisted needle steering. *IEEE Robotics and Automation Magazine*, 18(4):35-46.
- Sahin, H., & Guvenc, L. (2007). Household robotics: autonomous devices for vacuuming and lawn mowing. *IEEE Control Systems Magazine*, 27(2), 20-96.
- Santana, P., Barata, J., Cruz, H., Mestre, A., Lisboa, J. & Flores, L. (2005). A multi-robot system for landmine detection. *IEEE Conference on Emerging Technologies and Factory Automation*, (1), 721-728.
- Sukhatme, G.S. & Mataric, M.J. (2002). Robots: intelligence, versatility, adaptivity. *Communications of the ACM*, (45), 30-32.