

# ROS una mano para el desarrollo de robots. ROS a hand for the development of robots.

Porfirio. Cortez-Solis <sup>a,\*</sup>, Luis-Ángel. Ávila-Vázquez <sup>a,\*</sup>, Daniel-Eduardo, Palafox-Rodríguez <sup>a,\*</sup>.

<sup>a</sup> Ingeniería en Sistemas Computacionales, Tecnológico de Estudios Superiores de Ecatepec, 15EIT0001B, Ecatepec de Morelos, Edo. México, México.

## Resumen

El presente trabajo se analizará y detallará acerca del sistema operativo ROS, el cual nos proporciona bibliotecas y herramientas que sirven a los desarrolladores de software para crear aplicaciones robóticas. ¿Alguna vez han escuchado de este sistema operativo? Para que se utiliza, que es, cuáles son sus aplicaciones, en donde se utiliza, que se necesita, requerimientos de software, hardware, todos estos puntos son los que vamos a analizar para que podamos conocer más acerca de este sistema operativo que nos permite tener puerta abierta a esta tecnología que es el futuro. Podemos ver que la mayoría de los brazos robóticos se aplican en la industria que permiten que las tareas puedan ser automatizadas, que sean realizadas en menor tiempo, que sean eficaces y mejor terminadas. Ya que la demanda hoy en día en las aplicaciones que se les da uso son mayores tendremos que hacer el uso de la robótica para poder satisfacer estas necesidades de hoy en día.

**Palabras Clave:** Fuente abierta, robótica, herramientas, bibliotecas, software, Sistema operativo

## Abstract

The present work will analyze and detail about the ROS operating system, which provides us with libraries and tools that help software developers create robotic applications. Have you ever heard of this operating system? What is it used for, what is it, what are its applications, where is it used, what is needed, software requirements, hardware, all these points are what we are going to analyze so that we can learn more about this operating system that allows us to have an open door to this technology that is the future. We can see that most of the robotic arms are applied in the industry that allow tasks to be automated, to be carried out in less time, to be effective and better finished. Since the demand today in the applications that are used are greater, we will have to make use of robotics to be able to satisfy these needs today.

**Keywords:** open source, robotics, tools, libraries, software, operating system

## 1. Introducción

Los sistemas operativos hoy en día son la primera necesidad de todo trabajo, de toda actividad, de toda tarea para las empresas, para los negocios, para la mayor parte de las actividades cotidianas, hace algunos años atrás se empezaban a implementar El primer Sistema Operativo de la historia fue creado en 1956 para un ordenador IBM 704, y básicamente lo único que hacía era comenzar la ejecución de un programa cuando el anterior terminaba. Es importante no realizar cambios en el formato de esta plantilla, por lo que no se deberá cambiar tamaño, tipo de fuente o espaciados. Para énfasis utilizar la herramienta cursiva; no subrayar. En los años 60 se produce una revolución en el campo de los Sistemas Operativos. Después de ello la necesidad de utilizarlos para desarrollar varias actividades que nos permiten automatizar

toda la información fue de primera necesidad. En la industria anteriormente no se implementaban brazos robóticos hasta la actualidad haciendo que los procesos eran más lentos, pero al día de hoy, la implementación de ellos permite que estos procesos sean muchos más rápidos, pero para que se pudieran utilizar, desarrollar e implementar necesitan de un sistema operativo ROS en donde cualquiera lo puede aprender y poner en práctica, ya que es un sistema operativo libre.

La planilla conserva el formato, espaciado y demás elementos necesarios para realizar el manuscrito del artículo que se pretende publicar.

## 2. Antecedentes.

\*Autor para la correspondencia: [cortezsolisporfirio@gmail.com](mailto:cortezsolisporfirio@gmail.com)

Correo electrónico: 201521839@tese.edu.mx (Porfirio. Cortez-Solis), 201710002@tese.edu.mx (Luis-Ángel Ávila-Vázquez), 201510708@tese.edu.mx (Eduardo-Daniel Palafox-Rodríguez).

Dos estudiantes de doctorado de la Universidad de Stanford, Keenan Wyronek y Eric Berger estaban empezando sus doctorados y se enfrentaron a esa situación de reinención de la rueda: quienes se proponían innovar en el software de robótica dedicaban el 90% de su tiempo a reescribir el código que otros habían escrito antes para construir un prototipo de banco de pruebas, y solo el 10% de sus esfuerzos se dedicaba a la innovación. ROS fue su propuesta para hacer frente a esa enorme cantidad de tiempo perdido. Después en 2006 plantearon un plan para encontrar donantes que financiaran la construcción de 10 robots idénticos, llevarlos a 10 universidades y junto a un equipo de ingenieros de software construir el software base común y las herramientas para desarrolladores que permitieran compartir y reutilizar los avances de los demás. Utilizando aportaciones de la propia universidad y aportaciones individuales (como por ejemplo la recibida de Joanna Hoffman y Alain Rossmann, miembros del primer equipo de Apple Macintosh) construyeron el robot PR1, que sirvió para conseguir el apoyo al proyecto de los principales equipos de I+D de software de robótica de todo el mundo.

Mientras Keenan y Eric luchaban por hacer realidad su ambiciosa visión de hacer de ROS el “Linux de la robótica” conocieron a Scott Hassan, inversor y fundador de Willow Garage, un centro de investigación centrado en productos de robótica. Scott había creado empresas revolucionarias de Internet (Google y eGroups) utilizando software de código de abierto y quería que los futuros empresarios de la industria de la robótica tuvieran una base similar de código abierto. A Scott le pareció tan interesante la idea que en 2008 decidió financiarla y empezar con ellos un Programa de Robótica Personal dentro de Willow Garage. Así, en 2009 nació el robot PR2 y un año más tarde, en 2010, la primera versión de ROS. El proyecto ROS llegó a ser tan importante que todos los demás proyectos de Willow Garage fueron descartados y Willow Garage se concentró únicamente en el desarrollo y la difusión de ROS.

### 3. ROS (ROBOT OPERATING SYSTEM)

“El Sistema Operativo para Robots (ROS) es un marco de trabajo (framework) flexible para escribir software para robots. Se trata de un conjunto de herramientas, bibliotecas y convenciones cuyo objetivo es simplificar la tarea de crear un comportamiento robótico complejo y robusto en una amplia variedad de plataformas robóticas.” [1]



Fig. 1 Logotipo de ROS

#### Que ofrece ROS:

- 1.- ROS es un software de código abierto. ROS es gratuito y puede usarlo con fines comerciales.
- 2.- Todas las transformaciones del robot se rastrean en ROS.

Ya no necesitamos encontrar las ecuaciones de cinemática directa para encontrar la posición del efector final. Solo tenemos que proporcionar la descripción del robot en formato URDF. Eso es todo, el módulo tf o tf2 realiza un seguimiento de todas las transformaciones del robot desde la base hasta el efector final.

3.- Hay solvers de IK para resolver tus problemas de Ik.

El problema de la cinemática inversa ya no es un problema. Hay muchos complementos que resuelven el problema Ik para manipuladores robóticos simples a complejos. Por lo tanto, se ahorra tiempo para encontrar las ecuaciones de forma cerrada para Ik o encontrar la inversa de Jacobian. Pocos solucionadores de Ik son IKfast (Analytical Solver) y kdl (Numerical Solver).

4.- Sorprendido con la planificación de rutas y la evitación de obstáculos, no se preocupe, Moveit se encargará.

¡Moveit! es un conjunto de paquetes y herramientas para realizar planificación de movimiento, manipulación, percepción 3D, cinemática, comprobación de colisiones, control y navegación en ROS.

5.- Quiere trabajar con robots de accionamiento diferencial.

Luego use la pila de navegación ROS.

La pila de navegación contiene algoritmos de navegación listos para usar que se pueden usar en robots móviles, especialmente en robots con ruedas diferenciales. Usando la navegación ROS podemos hacer que el robot se mueva de forma autónoma en un entorno determinado.

6.- ROS proporciona una manera fácil de controlar los actuadores de los robots mediante el paquete de control ROS.

El control ROS proporciona un bucle de control PID incorporado con límites conjuntos para que el movimiento del robot sea rápido y suave.

7.- Abra CV con interfaz a ROS.

Podemos usar bibliotecas OpenCv con ROS para tareas de visión y percepción.

8.- ROS tiene herramientas útiles para visualizar su robot. Ellos son Rviz y Gazebo

Podemos hacer una réplica exacta de su robot en Gazebo con todas las propiedades físicas (como peso, inercia, colisiones, etc.) y probar su código en simulación antes de probarlo en hardware real. Rviz es una herramienta de depuración que visualiza lo que está experimentando su robot real/simulado (como el movimiento de las articulaciones, la visión, etc.).

9.- ROS es ligero.

ROS puede ejecutarse en placas integradas como Raspberry pi.

#### Comandos de ROS

**roscd:** cambia a un directorio de paquete o pila (ej. roscd stage).

**roscore:** ejecuta todo lo necesario para que dar soporte de ejecución al sistema completo de ROS. Siempre tiene que estar ejecutándose para permitir que se comuniquen los nodos.

Permite ejecutarse en un determinado puerto (ej. Roscore o roscore -p 1234)

**roscrcate-pkg:** crea e inicializa un paquete. Se tiene que ejecutar desde uno de los directorios válidos para que contengan paquetes. El formato de ejecución es: roscrcate-pkg paquete [depen1 ...] donde depen1 es una dependencia. Por ejemplo, si el paquete que estamos creando va a usar los mensajes estándar y va a usar código c++, debemos indicar las dependencias std\_msgs y roscpp.

**roscrcode:** nos proporciona información sobre un nodo. Disponemos de las siguientes opciones:

- **roscrcode info nodo** (muestra información sobre el nodo)
- **roscrcode kill nodo** (mata ese proceso)
- **roscrcode list** (muestra los nodos ejecutándose)
- **roscrcode machine maquina** (muestra los nodos que se están ejecutando en la máquina).
- **roscrcode ping nodo** (comprueba la conectividad del nodo).

**roscrcrun:** permite ejecutar cualquier aplicación de un paquete sin necesidad de cambiar a su directorio. Podemos pasarle parámetros con \_my\_param:=value (ej. roscrcrun stage stageros) stage es el paquete y stageros es la aplicación que ejecutamos.

**roscrcrostopic:** permite obtener información sobre un tópico.

- roscrcrostopic bw (muestra el ancho de banda consumido por un tópico)
- roscrcrostopic echo (imprime datos del tópico por la salida estándar)
- roscrcrostopic find (encuentra un tópico)
- roscrcrostopic info (imprime información de un tópico)
- roscrcrostopic list (imprime información sobre los tópicos activos)
- roscrcrostopic pub (publica datos a un tópico activo)
- roscrcrostopic type (imprime el tipo de información de un tópico)

**roscrcroswtf:** permite chequear si algo va mal. Ejecutamos roscrcd y después roscrcroswtf.[2]

#### 4. Aplicaciones.

##### Robotframework

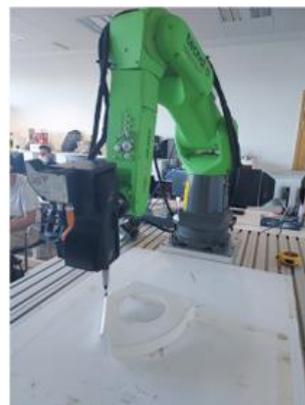
Se ha utilizado para el desarrollo de un sistema robótico capaz de realizar una monitorización autónoma y continua en invernaderos que permite la detección, identificación y control de plagas (en el contexto del proyecto europeo GreenPatrol), y también para el desarrollo de una solución de robótica colaborativa para la inspección mediante ultrasonidos de componentes aeronáuticos (en el contexto del proyecto Cro-Inspect. el uso de Robotframework puede reducir significativamente el tiempo de preparación de nuevas aplicaciones robóticas con manipuladores móviles en tareas relacionadas con la agricultura y la industria.



Inspección avanzada de piezas compuestas complejas.

##### Célula robótica para acabado superficial de piezas.

Basado en el sistema ScanNPlan, TECNALIA ha desarrollado una célula robótica capaz de realizar procesos de pulido y rebardado sobre piezas de distinta geometría. CR-7iAL, un sistema de visión 3D de luz blanca estructurada, un perfilómetro láser y una herramienta seleccionada en función del proceso a realizar. Gracias al sistema modular de ROS y el soporte a distinto hardware (tanto los controladores de distintos robots como sensores), es posible intercambiar el equipamiento en función a los requerimientos de la aplicación sin que tenga un gran impacto sobre el grueso del desarrollo de la aplicación. [3]



Detalle de la célula robótica: a) real,

Entre algunos otros de los robots que ya utilizan ROS se pueden encontrar:

- **PR1:** robot personal desarrollado por el laboratorio Ken Salisbury en Stanford.
- **PR2:** robot personal que está siendo desarrollado por Willow Garage.
- **Baxter** de Rethink Robotics, Inc.
- **Robot de Shadow:** mano robótica diestra motorizada desarrollada por la empresa Shadow y la cual se está desarrollando mediante el consorcio de un proyecto europeo dentro del marco europeo. Entre los participantes de este proyecto se pueden encontrar la empresa Shadow Robot, la Université Pierre et Marie Curie-Paris (Francia) o la Universidad Carlos III de Madrid (España).

**HERB:** desarrollado en CMU dentro del programa de robótica personal de Intel.[4]

##### Celda robotizada de Inspección Óptica

Se han desarrollado nuevos paquetes para aquellas funcionalidades para las que no había paquetes ROS disponibles, entre los que se encuentran, por ejemplo, un paquete para el control de la máquina a alto nivel mediante un enfoque de máquina de estados, dotado de características específicas en el control de maquinaria industrial: gestión de eventos relativos a la seguridad (cruce de barreras, paros de emergencia, etc.), gestión de alarmas, mensajes al operador, además de la gestión del proceso, y paquetes para la interacción con otros dispositivos, como un lector de código de barras, interfaz con el PLC y un paquete encargado de la parametrización y control del sistema de visión Cognex. Todo ello se complementa con una interfaz de visualización y control (HMI) basada en el software PASVisu 111 de PILZ, que hace uso de OPC-UA para el intercambio de datos con el PLC de seguridad y el sistema ROS.

## 5. Conclusiones.

La difusión de ROS en el entorno académico es notable y marcadamente creciente en el entorno industrial, lo que deja entrever que en un futuro cercano puede ser un elemento común e indispensable en muchas aplicaciones de la robótica, cada vez más variadas y cubriendo nuevos sectores productivos y de servicios. Evidentemente, la velocidad con que se producen cambios tecnológicos puede provocar la aparición de nuevos estándares en el campo de la robótica, pero, dada su difusión actual, es previsible que ROS marque el camino a seguir, ya sea con sus actuales características o con posibles variantes derivadas de ellas.

## Agradecimientos

Agradezco a mis padres por el apoyo de cada día hasta el término de mi carrera, aquellos que me han impulsado, que me han guiado por el camino del estudio, gracias al esfuerzo que han empeñado en mi puedo concluir mis estudios con gran alegría, honor y orgullo para poder emprender en el mundo laboral.

Gracias a cada uno de mis compañeros por ayudarme, apoyarme cuando más lo necesite, cuando más no podía ellos estuvieron incondicionalmente viendo cada uno de mis logros y nuestros logros, de los cuales me llevo de cada uno de ellos un conocimiento, una anécdota y un aprendizaje.

## Referencias

- “ROS: Home.” <https://www.ros.org/> (accessed Nov. 28, 2022).
- A. Martínez, “Manual de ROS | PDF | Robot | Robótica,” Oct. 31, 31AD. <https://es.scribd.com/document/433260767/Manual-de-ROS> (accessed Nov. 28, 2022).
- R. Suárez, J. Rosella, and M. Vinagreb, “Robot Operating System (ROS),” *Este trabajo se ha llevado a cabo por iniciativa y en el seno del Grupo de Trabajo de Innovación de la Asociación Española de Robótica y Automatización (AER)*. . Accessed: Nov. 28, 2022. [Online]. Available: [chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.aer-automation.com/wp-content/uploads/2022/03/ROS\\_articuloAER.pdf](chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.aer-automation.com/wp-content/uploads/2022/03/ROS_articuloAER.pdf)
- D. Ortega Delgado, “Qué es ROS (Robot Operating System) | OpenWebinars,” Sep. 21, 2017. <https://openwebinars.net/blog/que-es-ros/> (accessed Nov. 28, 2022).

[1]

## Apéndice A. Primer Apéndice

- ROS permite el desarrollo de aplicaciones para robot más fácil y con los requerimientos que se necesita para ejecutar
- Se pueden crear infinidad de proyectos con una gran facilidad gracias a las librerías que contienen.