

INTRODUCCIÓN A ROS

Héctor Mauricio Yepez Ponce

4 de febrero de 2018

La investigación académica e industrial en el desarrollo de diferentes tipos de robots, la exigencia y flexibilidad que éstos deben poseer a obligado al desarrollo de lo que hoy se conoce como **MRS** (Multi-Robot Systems). Los sistemas MRS son plataformas que permiten la implementación de varios robots en una tarea conjunta, existen diferentes plataformas que se dedican a cierto tipo de robots y una de ellas es ROS. Ros en pocos años se convirtió en un sistema pionero en MRS que ha crecido de manera exponencial, con mas de 3000 paquetes de aplicaciones y utilizado en casi 60 plataformas robóticas industriales y de investigación con una realimentación continua por múltiples investigadores y fanáticos de la robótica que buscan mejorar o crear nuevas opciones.

A pesar que su nombre es la sigla para sistema operativo, en realidad ROS es un meta-sistema operativo ya que es un sistema operativo que se instala sobre otro. Lo más usual es que sea un sistema UNIX (Ubuntu, Debian). Por lo tanto es de código abierto mantenido por la Open Source Robotics Foundation (OSRF). Incluyendo las abstracciones de hardware, control de dispositivos a bajo nivel, implementación de utilidades comunes, paso de mensajes entre procesos y gestión de paquetes. También proporciona herramientas y librerías para obtener, compilar, escribir y ejecutar código a través de múltiples ordenadores.

ROS fue originalmente desarrollado en 2007 por el Laboratorio de Inteligencia Artificial de Stanford (SAIL) con el soporte del proyecto Standfor AI Robot. En febrero de 2013, ROS se transfirió a la Open Source Robotics Foundation. ROS está liberada bajo los términos de la licencia BSD (Berkeley Software Distribution) y un software open source, gratuito para el uso comercial, investigación y promoviendo la reutilización de código. El objetivo de este sistema operativo es simplificar la tarea de crear cualquier robot imaginable, entregar facilidades para darles funcionalidades y hacerlos robustos y maleables mediante una serie de bibliotecas, drivers y herramientas que como parte de un MRS, le apuestan a la modularidad para ayudar a los desarrolladores.

0.1. Obejetivo

El objetivo de este sistema operativo es simplificar la tarea de crear cualquier robot imaginable, entregar facilidades para darles funcionalidades y hacerlos robustos y maleables mediante una serie de bibliotecas, drivers y herramientas que como parte de un MRS, le apuestan a la modularidad para ayudar a los desarrolladores.

0.2. Características

- Le apuesta por la modularidad (diferentes áreas que se conjugan en un proyecto).
- En ROS cada módulo es autónomo (como las computadoras en una red LAN).
- Cada módulo interactúa entre sí por medio de mensajes (protocolo XML-RCP) esto hace posible la programación en diferentes lenguajes C++ Python Java.

- Hace uso del protocolo TCP/IP para un esquema cliente-servidor principal servidor es el núcleo de ROS.

0.3. Componentes

0.3.1. Master

El ROS Master proporciona el registro de nombre y consulta el resto del grafo de computación. Sin el maestro, los nodos no serían capaces de encontrar al resto de nodos, intercambiar mensajes o invocar servicios.

0.3.2. Nodos

Los nodos son procesos que realizan la computación. ROS está diseñado para ser modular en una escala de grano fino; un sistema de control de robots comprende por lo general muchos nodos. Un nodo de ROS se escribe con el uso de las librerías cliente de ROS, roscpp (C++) y rospy (Phyton).

0.3.3. Servidor

El servidor de parámetros permite almacenar datos mediante un clave en una localización central, actualmente es parte de Master.

0.3.4. Mensajes

Los nodos se comunican mediante el paso de mensajes. Un mensaje es una estructura de datos compuestos.

0.3.5. Topicos

Un nodo envía mensajes publicando en un tópico. El tópico es un nombre que se usa para identificar el contenido del mensajes.

0.3.6. Servicios

La petición y respuesta se realiza a través de los servicios, que se definen a partir de una estructura de mensajes: una para la petición y otra para la respuesta. Un nodo proporciona un servicio con un nombre y un cliente utiliza dicho servicio mediante el envío del mensaje de petición y espera a la respuesta.

0.3.7. Bolsas

Las bolsas es un formato para guardar y reproducir de nuevo mensajes de ROS. Las bolsas son un mecanismo importante para el almacenamiento de datos, como lecturas de sensores, que pueden ser difícilmente adquiridas pero son necesarias para el desarrollo y testeo de algoritmos.