

APLICACIÓN DEL SISTEMA ROS EN DIFERENTES TIPOS DE ROBOTS

Héctor Mauricio Yepez Ponce

12 de febrero de 2018

El objetivo de ROS es la reutilización de código que se puedan implementar en robots con diferente hardware, con la finalidad de reducir el tiempo en la escritura de código y aportar con investigación en el mundo académico y la industria.

ROS puede resumirse en 4 objetivos filosóficos:

- **Topología punto a punto.** - Permite encontrar un proceso con otro en un determinado tiempo de ejecución, para esto se requiere de algún tipo de mecanismo de búsqueda que permita buscarlos y se denominan *servicios o master*.
- **Basado en herramientas.** - Para gestionar la complejidad de ROS, se ha optado por el diseño de un microkernel, donde se utilizan herramientas que permitan generar y ejecutar los diversos componentes de ROS, en lugar de la construcción de un entorno de desarrollo y ejecución monolítica.
Por ejemplo: Navegar por el árbol de código fuente, obtener y establecer parámetros de configuración, visualizar la topología de conexión, ancho de banda, dar forma gráfica a los datos, generar auto documentación, etc.
- **Multi lenguaje.** - La diversidad existente de código podría resultar un problema al momento de programar, es por eso que ROS actualmente soporta cuatro tipos de lenguaje diferentes: C++, Python, Octave y LISP.
- **Flexible.** - ROS reutiliza código de numerosos proyectos, esto se debe a que su sistema de construcción de ROS puede actualizar automáticamente el código fuente de los repositorios externos, aplicar parches y más.
- **Libre y código abierto.** - El código está disponible públicamente, bajo términos de licencia BSD, que permite el desarrollo de proyectos tanto comerciales y no comerciales.

HERRAMIENTAS DE ROS

Para obtener un sistema robótico en funcionamiento para los diferentes experimentos, debe existir un ecosistema de software mucho mas grande. Es por eso que ROS esta diseñado para minimizar la dificultad de depuración ya que tiene una estructura modular que permite que los nodos puedan tener un desarrollo activo y depurar los preexistentes.

Colaboración con el desarrollo. - La colaboración entre investigadores es necesaria con el fin de construir grandes sistemas. Con el objetivo de apoyar el desarrollo colaborativo, el sistema ROS está organizado en paquetes que contiene directorios y subdirectorios con diferentes tipos de funciones como: bibliotecas, ejecutables independientes, comandos para automatizar las demostraciones y pruebas etc. Esto permite al usuario la fácil manipulación y modificación de código.

ROS COMO PLATAFORMA PARA EXTENDER LAS CAPACIDADES DE LEGOS NXT

La gran cantidad de experimentos tiene como objetivo fomentar el uso de ROS, desarrollo de algoritmos por medio de lenguajes de programación como C++ y Python.

En este caso Laser Scanner es un proyecto sencillo tiene como objetivo demostrar el uso de ROS, sus partes y función es escanear un objeto el cual se encuentra en una posición para que un láser de línea lo ilumine de arriba hacia abajo para poder tomar una foto del mismo, una vez que tome la foto girará sucesivamente hasta realizar un giro completo y regrese a su posición inicial y posteriormente la figura ser representada en el software.

Mas que nada esta aplicación fue realizada con fines educativos que permiten mostrar la implementación de LEGO NXT con un sistema ROS, como base para la investigación y puesta en marcha en campos como la botánica, robótica y demás campos investigativos.

DESARROLLO DE UN ROBOT MOVIL COMPACTO INTEGRADO EN EL MIDDLEWARE ROS

Este es un proyecto que implica un grado mas de complejidad con el uso de un microcontrolador que permite recibir las señales para luego ser enviadas al ordenador con la finalidad de optimizar el tiempo de envío y recepción. TraxBot es un robot cuyo objetivo es reducir el tiempo de desarrollo, proporcionando abstracción de hardware y modos de operación intuitiva.

El sistema de control comprende un microcontrolador y un ordenador, el sistema de actuación lo componen principalmente los motores eléctricos, el sistema de soporte define la alimentación de las plataformas y el sistema de comunicación que se la realiza por medio de WIFI.

Independientemente de la arquitectura los robots requieren de un middleware que es un sistema embebido, concurrente, en tiempo real, distribuido y debe garantizar propiedades del sistema como seguridad, confiabilidad y tolerancia a fallos.

Un aspecto que hay que tener en cuenta es el uso de una placa Arduino UNO que también es de código libre y presenta una interfaz fácil de usar para la comunicación, permitiendo el intercambio de información entre el robot y la unidad remota. Si bien es cierto el microcontrolador Atmega328p puede ofrecer una flexibilidad limitada debido a sus limitaciones de hardware en comparación con otros sistemas puede resultar una muy buena opción, para compensar estas limitaciones, se usa un ordenador para procesamiento externo el cual esta dedicado a ejecutar ROS.

Para lograr esta comunicación entre Arduino y ROS es necesario utilizar la comunicación punto a punto por lo que se usa *stack rosserial* haciendo uso de un protocolo que permite intercambiar mensajes de ROS con Arduino UNO.

ROS permite monitorizar, depurar la información intercambiada y los nodos que se ejecutan en la red así como el uso de agentes virtuales simulados que usan el mismo código para simular robots reales o virtuales entre los simuladores más usados son *stage* o *gazebo*. Traxbot usa sensores sonares que le permite hacer un mapeo del escenario basado en la combinación de la información de odometría y las lecturas de los sonares. Se puede observar que ROS representa un sin número de ventajas trabajando como un sistema independiente así como asociado con demás dispositivos de control, la gran cantidad de herramientas y paquetes que se encuentran disponibles en los repositorios permiten reducir el tiempo de desarrollo a través de código, permitiendo partir de una base sencilla hacia lo más complejo.