

Práctica No. 5: Robot Operating System - ROS

Alberto Acosta López, Francisco Javier López Franco
Luis Fernando Peña Flores

Resumen—Durante la elaboración de esta práctica se implementó un pequeño sistema robótico, donde primero se logró una comunicación entre dos servicios dentro de una misma computadora, después una comunicación entre un componente físico y la computadora y, finalmente, una comunicación entre dos computadoras distintas. Se llevó a cabo un trabajo en paralelo donde cada integrante realizó su parte al mismo tiempo, con la finalidad de eficientar el tiempo disponible y se logró establecer una comunicación entre los distintos componentes previamente especificados.

1. INTRODUCCIÓN

En un sistema robótico existen distintos y diferentes componentes que deben estar en sincronía para que realicen el trabajo de manera eficiente, es por esto que se crearon lenguajes de programación que hacen uso de los diferentes componentes dentro del sistema, para que cada uno realice su trabajo o tarea de manera eficiente y exitosa. Por ello, en esta práctica se tuvo como objetivo comprender el funcionamiento de uno de esos lenguajes de programación, siendo ROS para esta práctica.

Para comprender el funcionamiento de lenguajes de programación orientados a sistemas robóticos, se tiene que aprender a mandar mensajes entre distintos componentes simulados dentro de una misma computadora; establecer una línea de comunicación con un componente físico, sin necesidad de una simulación para que al final, se tenga una comunicación entre distintas computadoras con un rol específico en cada una de ellas.

Para que se pueda escalar las dimensiones del sistema robótico, es necesario comenzar por pequeños componentes, entre ellos Arduino, siendo Arduino un componente previamente utilizado en este curso. Así, se logra enfocar en comprender la estructura de la compilación y ejecución de ROS.

Este documento está estructurado de la siguiente manera: marco teórico, para describir y dar información de los componentes utilizados; desarrollo, para dar a comprender los pasos seguidos durante esta práctica; resultados, donde se exponen y presentan los objetivos alcanzados al finalizar la práctica; conclusiones, donde se presentan las conclusiones individuales de los integrantes del equipo; rol o papel, donde se explica la división del trabajo entre los integrantes del equipo; y las fuentes consultadas.

2. MARCO TEÓRICO

ROS es un meta-sistema operativo *open-source* para robots, ya que provee los servicios que se esperan de un

sistema operativo, incluyendo abstracción de hardware, control de dispositivos a bajo nivel, implementación de funcionalidades comúnmente usadas, paso de mensajes entre procesos y manejo de paquetes. El principal objetivo de ROS es soportar reuso de código en investigación robótica. Es un *framework* de procesos que habilitan ejecutables para que se diseñen y acoplen individualmente al momento de ejecución. ROS está diseñado alrededor de manipulación de plataformas móviles complejas, haciendo más fácil tomar ventaja de un ambiente distribuido de computadoras. [1]

Los conceptos con los cuales ROS funciona son: [2]

1. **Nodos:** Son los procesos computables. ROS está diseñado para ser modular a pequeña escala, por lo que un sistema controlador de robots usualmente comprime varios nodos.
2. **Master:** Provee los nombres de registros y busca a través de la red de ROS a los demás nodos.
3. **Servidor de parámetros:** Permite que la información sea almacenada por una llave en una localidad central.
4. **Mensajes:** Es una estructura de datos que comprenden muchos tipos de campos. Es la forma mediante la cual se comunican los nodos.
5. **Tópicos:** Es el nombre utilizado para identificar el contenido del mensaje. Un nodo que esté interesado en cierto tipo de mensaje, se suscribe al tópico apropiado.
6. **Servicios:** El *request/reply* de los nodos es hecho a través de servicios, los cuales son un par de estructuras de mensajes: uno para request y otro para esperar el reply.
7. **Bolsas:** Son el formato para guardar y repetir mensajes de información de ROS.

El sistema funciona como se muestra en la figura

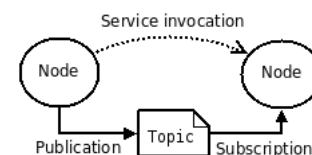


Figura 1. Conceptos básicos de ROS

Arduino es una plataforma *open-source* utilizado para construir proyectos electrónicos, ya que consiste de una parte física programable, referido como un microcontrolador, y

una parte de software, utilizado para escribir y mandar el código de la computadora al dispositivo. A diferencia de otros dispositivos programables, Arduino no necesita otro pedazo separado de hardware para mandarle nuevo código, sino que requiere un cable USB. [3]

3. DESARROLLO

Como se mencionó anteriormente, la práctica se dividió en cuatro secciones. Para cada una de ellas, exceptuando la última sección, se generó un código, en C++, capaz de resolver los problemas presentados en las mismas. Dicho código puede ser encontrado en el repositorio asociado al GitHub Classroom del curso.

La primera sección de la práctica consistió en implementar un programa en ROS capaz de enviar un mensaje de tipo String cuando un nuevo mensaje sea recibido. Una vez recibido el mensaje este se leía y se imprimía en la terminal correspondiente.

Para dicho problema fue necesario descargar los archivos iniciales del repositorio provisto por el profesor. Inmediatamente se compiló y se probó su correcto funcionamiento. Finalmente, bastó con la inserción de las líneas de código de envío de mensajes, tanto en el código del suscriptor como en el del publicador, para lograr el efecto deseado.

A lo largo de la segunda sección de la práctica se utilizó el simulador Turtlesim para aprender lo básico de ROS y sus paquetes. El objetivo fue implementar un programa capaz de mover la tortuga de Turtlesim dados distintas instrucciones del teclado. Por ejemplo, al introducir un '4' a través del teclado la tortuga debía de moverse hacia adelante.

En la solución de este problema, se utilizó como punto de partida el código generado en la primera sección de la práctica. Dicho código, capaz de enviar y recibir mensajes, fue clave para la comunicación entre el usuario y la tortuga de Turtlesim.

Durante la tercera sección de la práctica, se conjuntó la experiencia adquirida con ROS de las primeras dos secciones para comunicar un Arduino a través de ROS. El experimento consistió en, utilizando un Arduino, escuchar constantemente por nuevos mensajes, al escuchar un nuevo mensaje, prender un led durante 3 segundos y, finalmente, enviar un mensaje de respuesta.

Para atacar este problemática se alambrió en una protoboard el circuito habitual para prender el led. Además, se programó el código correspondiente al envío y la recepción de mensajes. Y, por último, se configuró el Arduino para prender el led y enviar la confirmación. Dicha configuración fue realizada como en las primeras prácticas de laboratorio.

Finalmente, en la última sección de la práctica, el objetivo fue ejecutar el programa de Turtlesim, realizado

en la segunda sección de la práctica, en otra computadora. Para dicho efecto, bastó con agregar las configuraciones pertinentes a las computadoras en cuestión.

Dichas configuraciones consistieron en identificar la dirección IP de cada computadora y asociarlas con las variables de ROS_HOSTNAME correspondiente. Además, se identificó con claridad cual computadora fue la 'maestra' y la 'esclava' al modificar la variable ROS_MASTER_URI.

4. RESULTADOS

En la primera sección de la práctica se tuvo un largo trabajo de investigación para entender la estructura de un proyecto de ROS. Una vez realizado dicha investigación, fuimos capaces de realizar el envío de mensajes sin más complicaciones.

Respecto a la segunda sección de la práctica se tuvo que hacer, de nueva cuenta, una investigación acerca de Turtlesim y su funcionamiento básico. Eso, y aunado a las sugerencias dadas en la práctica, permitió que completáramos la simulación exitosamente.

La tercera sección de la práctica fue la más complicada. Esto debido a las distintas instalaciones y configuraciones extra que debimos hacer tanto en Arduino como en ROS. Apoyados con el código realizado en la primera sección de esta práctica y en el de las primeras prácticas de laboratorio, pudimos realizar el prendido del led y el envío de los mensajes.

Por último, la cuarta sección de la práctica se completó de forma exitosa. Conjuntando conocimientos básico de la materia de Redes de Computadora con el programa realizado en la segunda sección de la práctica, fuimos capaces de ejecutar el programa remotamente en otra computadora.

5. CONCLUSIONES

5.1. Alberto Acosta

Esta práctica me brindó un conocimiento mayor acerca de la estructura de un lenguaje de programación desde una perspectiva más orientada a tener un control de cada proceso. Gracias a esto, aprendí a detectar errores y especificar rutas para que la compilación y el servicio funcionen de manera correcta. Fue crucial realizar los primeros pasos de forma ordenada para que las rutas de los archivos fueran las correctas y que el servicio funcionara adecuadamente. Fue satisfactorio ver la comunicación entre los distintos equipos y componentes y espero aprender más en el proyecto final.

5.2. Francisco López

Gracias a la práctica me di cuenta de la funcionalidad de ROS y de las facilidades que te brinda para realizar conexiones entre máquinas para así poder realizar sistemas más complejos. Sin este programa, el trabajo para lograr lo mencionado sería bastante tedioso. Además, el código utilizado para que funcione es bastante claro y muy simple de realizar, por lo que espero poder aprovecharlo para lograr realizar un proyecto final satisfactorio.

5.3. Fernando Peña

A lo largo de la presente práctica puede comprender lo básico de Robot Operating System, algunas de sus aplicaciones y de sus funcionalidades. Asimismo, fui capaz de realizar simulaciones de problemas a implementar físicamente, lo cual es de suma importancia antes de pasar a la construcción física del modelo propuesto. Una aplicación futura será en el proyecto final, que permitirá operar los motores del dispositivo a mover.

6. ROL O PAPEL

1. **Alberto:** Revisar puntos a checar de la práctica acerca del lenguaje C y de ROS e y realizar del reporte.
2. **Francisco:** Realizar el código en C para la funcionalidad de ROS.
3. **Fernando:** Realizar el reporte y apoyo en las conexiones de ROS desde distintas máquinas.

7. FUENTES CONSULTADAS

REFERENCIAS

- [1] O. source Robotics Foundation. Ros introducion. [Online]. Available: <http://wiki.ros.org/ROS/Introduction>
- [2] O. S. R. Foundation. Ros concepts. [Online]. Available: <http://wiki.ros.org/ROS/Concepts>
- [3] Arduino. Introduction to arduino. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>