

Práctica 5: Robot Operating System - ROS

Paola Del Hierro 157073
Luis Eduardo Delfín 155966
Rafael Zardain Bejar 158948

1. INTRODUCCIÓN

Programar software para crear el comportamiento de un robot puede ser muy complejo y difícil [1]. Tener protocolos y herramientas que faciliten y conviertan esta tarea más homogénea puede ser de gran ayuda para los equipos de desarrollo. La cooperación entre equipos que fomenta los proyectos de fuente libre ayuda al desarrollo de tecnologías ágiles.

En esta práctica se utilizó el sistema ROS para realizar distintas actividades. La principal de estas fue un programa que controlaba a una tortuga en una ventana de la computadora. Para esto fue necesario entender las funcionalidades principales de ROS, el paquete *Turtlesim* y realizar la conexión ROS-Arduino. El objetivo principal de este trabajo fue familiarizarse con estos conceptos, de los cuales el último tiene mayor relevancia para trabajos futuros.

El esquema de este documento es el siguiente. En la sección dos se dará una breve explicación de las tecnologías utilizadas durante la práctica. En la tercera sección se desplegará el desarrollo del ejercicio y los resultados obtenidos estarán en la sección 4. En la sección 5 estarán las conclusiones a las que se llegaron por integrante del equipo. Los papeles que desarrollaron los integrantes se presentarán en la sección 6 y finalmente se mostrarán las referencias bibliográficas.

2. MARCO TEÓRICO

ROS (Robot Operating System por sus siglas en inglés) es una herramienta flexible que facilita el desarrollo de software para robots. Se conforma de distintas librerías y herramientas que simplifican la tarea de programar el comportamiento complejo de un robot. ROS, además, es compatible con distintas plataformas robóticas. Este sistema permite el desarrollo de software de robots colaborativo pues es un proyecto *open source*. [1]

Turtlesim es un paquete de ROS que simula a una tortuga en una ventana de la computadora. El color del fondo se puede cambiar y se puede mover a la tortuga en el plano. El objetivo de este paquete es ser una forma de aprender las funciones básicas de ROS y como se utiliza este. [2]

Arduino es una excelente herramienta para programar hardware. Se puede conectar con ROS utilizando el paquete *roserial-arduino*. Este paquete vuelve al Arduino en un

nodo de ROS y permite que publique y suscriba a mensajes de ROS. [3]

3. DESARROLLO

Primero se preparó el sistema Linux para que corra ROS correctamente. Esto incluyó la instalación de varios paquetes. Después de una instalación correcta se hicieron dos experimentos, *TurtleSim* y Arduino. *TurtleSim* es una simulación de movimiento y posicionamiento que se expresa en una tortuga. El código para *TurtleSim* podía ser escrito en Python o en C++. Después de escogerse C++ se realizaron varias pruebas con varios comandos de teclado para poder operar la tortuga. La implementación en Arduino fue un poco diferente, ya que el Arduino tenía que servir como un receptor de mensajes enviados vía ROS y que actuara dependiendo del mensaje recibido.

4. RESULTADOS

Durante la primera parte de la práctica, se logró comprender la estructura básica de un proyecto de ROS. Después de comprender dicha estructura, se logró que el suscriptor le respondiera al publicador con un mensaje tipo String cuando un nuevo mensaje sea recibido y que el publicador lea e imprima a la terminal el mensaje que recibió.

Durante la segunda parte de la práctica se logró implementar un programa que mueva la tortuga utilizando el teclado en un proyecto de ROS. Primero, logramos correr el simulador de la tortuga que ya tenía programado dicha función de movimiento de la tortuga. Después, al ver como funcionaba, lo replicamos con nuestro propio código y funcionó. El código se encuentra dentro de la carpeta práctica 5.

Lo correspondiente a la tercera parte de la práctica, la sección de ROS-Arduino, no se pudo lograr debido a la falta de tiempo y a complicaciones en la instalación de ciertas librerías.

5. CONCLUSIONES

Paola Del Hierro En esta práctica se logró comprender los conceptos y funcionamiento básicos de ROS. Estos conceptos, así como la conexión ROS-Arduino serán útiles en proyectos futuros. Sin embargo no se logró completar totalmente debido a la falta de tiempo.

Luis Eduardo Delfin A pesar de que no se logró completar la practica en su totalidad, en esta práctica se logró comprender los conceptos y funcionamientos básicos de ROS. Me di cuenta del potencial y utilidad que tiene ROS.

Rafael Zardain La practica fue un buen comienzo de ver todos los componentes de ros para aplicarlo en nuestra practica final Fue buen descubrimiento de todo un entorno.

6. ROL O PAPEL

Paola Del Hierro: Programación de Turtlesim.

Luis Eduardo Delfin: Programacion en ROS para el protocolo de comunicación y Programación de Turtlesim.

Rafael Zardain: Programacion para Arduino-ROS y Programación de Turtlesim.

REFERENCIAS

- [1] ROS". *Ros.org* <https://www.ros.org>
- [2] T.L Harman, C. Fairchild, "Turtlesim, the first ROS robot simulation". *ROS Robotics By Example* <https://learning.oreilly.com/library/view/ros-robotics-by/9781782175193/ch01s07.html>
- [3] .Arduino IDE Setuop" *Ros.org* http://wiki.ros.org/rosterial_arduino/Tutorials/Arduino%20IDE%20Setup