Práctica No.5 Robot Operating System - ROS. Principios de Mecatrónica

Arcadio Alexis Calvillo Madrid 159702
Arlet Díaz Méndez 154840

Resumen—Durante esta práctica se realizaron programas para conocer el Sistema Operativo Robótico. Se implementó la conexión de un nodo publicador y un suscriptor para ver la funcionalidad de comunicación con la que opera ROS. Así mismo, se implemento un programa para el simulador Turtlesim que permite mover a la tortuga a través de botones del teclado. Además, se implemento un programa utilizando Arduino que muestra la interacción de ROS con actuadores. Finalmente, se conectaron dos computadoras para lograr que el controlador de Turtlesim fuera ejecutado en una máquina distinta a la que se encargó de correr el simulador.

1. INTRODUCCIÓN

En la actualidad, la presencia de robots es cada vez más frecuente en muchos aspectos de la vida cotidiana y de las diferentes industrias. Existen robots comerciales e industriales que se encargan de llevar a cabo procesos que requieren de alta precisión o que pueden ser más eficaces al ser relizados por una máquina en lugar de un humano. Algunas de sus aplicaciones se encuentran en plantas de manufactura, montaje y embalaje, en transporte, en exploraciones en la Tierra y en el espacio, cirugía, armamento, investigación en laboratorios y en la producción en masa de bienes industriales o de consumo.

Los objetivos planteados para esta práctica son repasar la programación y compilación en C/C++, aprender el funcionamiento de ROS (Sistema Operativo Robótico) y cómo funciona en conjunto con Arduino. Para ello, primero, se implementará un programa de comunicación sencilla en la terminal de la computadora. Después, se implementará una simulación en Turtlesim que después se complementará ejecutando el controlador en una computadora distinta que el simulador. Finalmente, se implementará un programa para que utilizando Arduino y Ros se cree un controlador para la comunicación entre dos dispositivos y la implementación de una alerta.

Este documento tiene la siguiente organización: un Marco Teórico en el que se explican las tecnologías utilizadas, así como los conocimientos teóricos que las respaldan, un Desarrollo en el que se explica cómo contribuyen los componentes a la solución, una sección de Resultados en la que se habla de los valores obtenidos durante la práctica, luego se presentan las Conclusiones de la práctica, los Roles de cada integrante del equipo y las Referencias utilizadas durante el proceso.

2. MARCO TEÓRICO

ROS Robotic Operating System (Sistema Operativo Robótico) es un framework para el desarrollo de softare especializado en robots. Provee los servicios estándar de un sistema operativo: abtracción de hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes. Funciona bajo una estructura de gráfos en la cual los distintos nodos pueden recibir, mandar y multiplexar mensajes. Las aplicaciones de Ros son las siguientes:

- Un nodo principal de coordinación.
- Publicación o subscripción de flujos de datos: imágenes, estéreo, láser, control, actuador, contacto, etc.
- Multiplexación de la información.
- Creación y destrucción de nodos.
- Los nodos están perfectamente distribuidos, permitiendo procesamiento distribuido en múltiples núcleos, multiprocesamiento, GPUs y clústeres.
- Login.
- Parámetros de servidor.
- Testeo de sistemas.

Estas aplicaciones se utilizan para el correcto funcionamiento de los programas, pues ROS funciona principalmente por la comunicación de mensajes entre distintos dispositivos. Los programas pueden ser escritos en distintos lenguajes de alto nivel. Para esta práctica utilizaremos C/C++. [1]

Arduino Mega Es una placa de Arduino (compañía de desarrollo abierto de hardware)que dispone de un microcontrolador AVR Atmel de 8 bits. Se considera la placa más potente de Arduino debido a que es el que más pines i/o tiene, es apto para trabajos más complejos que el que soportan otras placas. El Arduino Mega usa el microcontrolador Atmega2560. [2]

IDE Arduino El entorno de desarrollo integrado de Arduino es una aplicación disponible para varios sistemas operativos desarrollado en Java que se utiliza para escribir y cargar programas en las placas Arduino. Este IDE permite el uso de los lenguajes C y C++ pero se utilizan reglas especiales de estructura de códigos. [3]

Lenguaje C Es un lenguaje de programación imperativo, de propósito general, que admite programación estructurada, alcance de variable léxica y recursión, lo que se define como lenguaje de alto nivel. Sin embargo, cuenta con un

sistema de tipo estático que evita muchas operaciones no intencionadas en bajo nivel. Lo anterior se debe a que es un lenguaje orientado a la implementación del sistema operativo Unix. [4]

3. DESARROLLO

Para alcanzar los objetivos de la práctica se llevaron a cabo cuatro actividades. Para empezar, se creó una comunicación entre nodos para mostrar el funcionamiento básico de ROS. Se modificarn los archivos dados por el profesor para crear dos nodos, un suscriptor y un publicador, en los cuales a través de dos instancias de la terminal se podían enviar mensajes de tipo string que eran replicados en la otra terminal por el publicador.

La siguiente actividad realizada fue la implementación de un controlador que controla el movimiento de la tortuga del simulador Turtlesim. A traves de los botones 2, 4, 6 y 8 del teclado se dirige el movimiento para indicarle la dirección en la que se debe mover.

Luego, se creó un programa en Arduino para conectar nodos publicadores y suscriptores con dispositivos externos a la computadora, en este caso LEDs. Se trató de lograr un envío y recepción de mensajes. Los LEDs sirven como alarma para indicar la llegada de un nuevo mensaje. Además, despues de 3 segundos de haber recibido el primer mensaje, y haber accionado la alarma, se envía una respuesta.

Por último, se habilitó el controlador hecho en la segunda tarea para que pudiera ejecutarse en una computadora distinta a la del simulador. Para esto, Se tiene que indicar a ROS en qué computadoras se ejecuta cada proceso. La identificación de las computadoras para este proceso se hizo mediante la dirección IP. Además, es necesario especificar cual es el puerto por el que ROS va a establecer la comunicación entre los nodos.

4. RESULTADOS

Los resultados obtenidos durante la práctica fueron satisfactorios para las primeras dos tareas. Sin embargo, por falta de tiempo no logramos desarrollar las últimas dos tareas. Para la actividad de la comunicación de dos nodos en una misma computadora se logró que al escribir un mensaje en la primera terminal ese se replicara en la otra terminal. En la segunda actividad, en TurtleSim, logramos hacer que la tortuga se moviera siguiendo las instrucciones que se le daban a trvés de las teclas al comunicar el simulador con el controlador de ROS.

5. CONCLUSIONES

Arlet: El funcionamiento de ROS es un tanto complicado pues es necesario declarar los nodos en diversos archivos. La localización de los programas y los nodos está declarada en muchos archivos, por lo que el principal problema para el equipo fue encontrar la falla en errores de ruta, pues tomaba tiempo encontrar en qué archivo se encontraba el error. Sin embargo, haciendo los cambios de forma ordenada no se producían errores y la ejecución de los programas era simple.

Alexis: En ROS el modo de ejecución es bastante más complicado que lo que usamos comúnmente. Uno de los principales problemas es identificar el nombre del nodo que se va a manejar. Por otro lado, permite el funcionamiento de clusters de trabajo que facilitan mucho cualquier tarea que queramos implementar en ROS.

6. ROL O PAPEL

Arlet: Modificación de los archivos para el establecimiento de la comunicación entre nodos ROS. Programación para el controlador de Turtlesim. Programación para el uso de ROS con Arduino. Implementación del hardware para la alarma de mensajes. Modificación de archivos para la comunicación de nodos ROS en dos computadoras distintas.

Alexis: Modificación de los archivos para el establecimiento de la comunicación entre nodos ROS. Programación para el controlador de Turtlesim. Programación para el uso de ROS con Arduino. Implementación del hardware para la alarma de mensajes. Modificación de archivos para la comunicación de nodos ROS en dos computadoras distintas.

7. FUENTES CONSULTADAS

[1] Wikipedia. Sistema Operativo Robótico. [En línea]. Disponible en: https://es.wikipedia.org/wiki/SistemaOperativoRobótico

[2] Arduino. Arduino Mega 2560. [En línea]. Disponible en: http://arduino.cl/arduino-mega-2560/

[3] Sin autor. Aprendiendo Arduino. [En línea]. Disponible en: https://aprendiendoarduino.wordpress.com/category/ide/

[4] Bonet, E. Lenguaje C. [En línea]. Disponible en: https://informatica.uv.es/estguia/ATD/apuntes/laboratorio/LenguajC.pdf