

# Proyecto Final

Pamela Flores Iberri  
151750

Alessandro Mario Cassiel Lombardo Fabian  
156041

**Abstract**—Este proyecto final consistió en la elaboración de un robot, el cual, a través de una comunicación inalámbrica con XBee, un Arduino y ROS, pudiera ser controlado para lograr realizar una trayectoria cuadrada, una circular y una trayectoria cooperativa con otros robots. En esta práctica se implementaron todos los conocimientos obtenidos en el transcurso del semestre.

**Index Terms**—Arduino, Robot Operating System, Xbee, Motor DC

- Motor de corriente directa (x2)
- XBee S1
- Puente H
- Interruptor óptico (x2)
- Resistencias
- Protoboard
- Jumpers

## I. INTRODUCCIÓN

En el proyecto final se espera cumplir los siguientes objetivos:

- Integrar los conocimientos en programación de micro-controladores, adquisición de información por medio de sensores, control de actuadores, planificación de tareas y ROS.
- Generar un prototipo que permita atisbar potenciales aplicaciones industriales y académicas de alto impacto.
- Consolidar una metodología base de construcción y documentación de proyectos mecatrónicos.

Los siguientes conceptos son necesarios para entender el procedimiento y resultados de este proyecto:

**Robot Operating System (ROS)**, es un set de librerías y herramientas open source que ayudan al desarrollo e implementación de aplicaciones robóticas. Provee los servicios esperados de un sistema operativo como son: abstracción de hardware, control de dispositivos a bajo nivel, implementación de funcionalidad, envío de mensajes entre dispositivos y manejo de paquetes.

**Arduino** es una herramienta de desarrollo y prototipado basada en C/C++, la cual permite la conexión entre microcontroladores, sensores y actuadores de manera sencilla.

Los **XBee** son radios utilizados para lograr una interconexión y una comunicación inalámbrica entre otros dispositivos XBees, los cuales cuentan con entradas y salidas digitales y analógicas.

Un **motor de corriente directa** convierte energía eléctrica en mecánica, a través de la generación de un campo magnético. Los principales componentes de este tipo de motor son el estator, la parte fija, y el rotor, la parte móvil/giratoria.

## II. MARCO TEÓRICO

Los instrumentos o aplicaciones utilizadas para la realización de la práctica fueron:

- ROS Kinetic Kame
- Ubuntu Xenial (16.04 LTS)
- Microcontrolador AT MEGA 2560

## III. DESARROLLO

El proyecto se desarrolló en tres secciones:

### A. Robot móvil

Para realizar el proyecto debemos ensamblar un robot móvil que modela un pequeño coche.

Para esto a cada lado de la base fijamos un motorreductor acoplado a una rueda y un encoder con 20 ranuras. También colocamos un interruptor óptico en cada lado, ésto para poder medir la velocidad de giro, y así poder obtener la velocidad angular. Al frente del coche colocamos un tercer punto de apoyo: una rueda sin motor que gira libremente. Entre los motores y el arduino se encuentra un puente H, el cual se encarga del control de las corrientes eléctricas que se envían por medio del arduino.

También se colocó un módulo arduino a bordo, el cual recibe la lectura de los encoders para después ser procesados por medio del PID. En este módulo se encuentra también el dispositivo XBee para la comunicación inalámbrica con ROS, sin embargo, debido a que no se logró la comunicación exitosamente, fue necesario removerlo.

Por último agregamos una batería de 9v para alimentar al motor.

### B. Comunicación

Para la comunicación entre los XBees se siguió el procedimiento de la práctica 3, en el cual configuramos los dispositivos y los colocamos en el arduino pero al intentar realizar la comunicación nos da un error con el siguiente mensaje: *Unable to sync with device; possible link problem or link software version mismatch such as hydro rosserial\_python with groovy Arduino.*

Este fue un problema general de todo el grupo y no logramos hayar una solución.

### C. ROS

Para controlar el movimiento del robot se utilizó ROS, para esto descargamos e instalamos las librerías de ROS para Arduino.

En nuestro programa de Arduino, además del controlador PID, tenemos un subscritor a un nodo de ROS de tipo

*geometry\_msgs::Twist*, que está formado por los campos velocidad lineal en X y angular en Z. Con esto usaremos el modelo de velocidad lineal para calcular la velocidad de cada rueda y lograr el movimiento deseado de la siguiente manera:

$$V_L = 1/r(V - LW/2)$$

$$V_R = 1/r(V + LW/2)$$

Donde  $V_L$  y  $V_R$  son las velocidades respectivas de cada rueda,  $V$  y  $W$  son las velocidades lineal y angular objetivo,  $r$  es el radio de las ruedas y  $L$  es la distancia entre ellas.

Nuestro nodo de arduino también tiene un publicador de tipo *geometry\_msgs::Pose2D*, en el cuál reportará la pose del robot, calculada con las lecturas de los encoders ópticos y aplicando odometría invirtiendo el modelo dinámico usado antes.

En la computadora tenemos otro nodo de ROS escrito en C. Este nodo está suscrito al tópico de tipo *geometry\_msgs::Pose2D* que publica el arduino del coche. Este nodo también es el encargado de publicar en el tópico de tipo *geometry\_msgs::Twist* los comandos para mover el coche a una posición deseada, los cuales calcula basado en la pose reportada por el auto y una pose objetivo que puede ser hard codeada o recibida de otro nodo maestro. En el método principal definimos los comandos para realizar las trayectorias cuadrada, circular y en grupo para nuestro robot.

#### IV. RESULTADOS

Debido a que no logramos realizar la comunicación del Arduino con la PC a través de XBee no logramos probar el proyecto con ROS. Por lo tanto tuvimos que controlar el robot directame con el Arduino para probar los controladores. De esta manera logramos que el robot describiera las trayectorias pedidas de una manera satisfactoria.

#### V. CONCLUSIONES

Para realizar el proyecto aprendimos a modelar el movimiento de un pequeño robot con 2 ruedas, y a implementar los controladores necesarios en una tarjeta Arduino para mover exitosamente al robot. También reafirmamos nuestro manejo de ROS y aprendimos a usarlo en un sistema de lazo cerrado. No pudimos comunicar los nodos de ROS de la PC y el Arduino, así que no los pudimos probar debido al problema que tuvimos al utilizar los dispositivos XBee.

#### REFERENCES

- [1] "ROS". Internet: <https://www.ros.org/>
- [2] "ROS/Introduction". Internet: <http://wiki.ros.org/ROS/Introduction>
- [3] "¿Qué es XBee?". Internet: <https://xbee.cl/que-es-xbee/>
- [4] "Motor de corriente continua". Internet: [https://es.wikipedia.org/wiki/Motor\\_de\\_corriente\\_continua](https://es.wikipedia.org/wiki/Motor_de_corriente_continua)