

Lectura y Monitoreo Inalámbrico de Sensores y Controladores de un Cuadricóptero utilizando ROS

Gabriel Noya Doval¹, Carlos Serrano Barreto¹

Resumen— Monitorear correctamente los datos provistos por los sensores y el valor de los controladores de los motores de un robot es imprescindible para detectar y corregir posibles fallos. Debido a lo peligroso que puede llegar a ser un cuadricóptero cuando ocurre un error, es necesario contar con un sistema de monitoreo a distancia, para así garantizar la seguridad del desarrollador. Dicho esto, se propone realizar un sistema de monitoreo comunicado por WiFi utilizando *Robotic Operating System* (ROS), que disponga de una interfaz gráfica que permita ver a tiempo real los datos de los sensores y el valor de los controladores de los motores, con el objetivo de asegurar la correcta lectura de los datos y que sea posible ajustar las ganancias de los controladores de una manera segura.

Palabras Clave— Cuadricóptero, controladores, sensores, ROS

I. DESCRIPCIÓN DEL PROYECTO

Se desea desarrollar e implementar un sistema de lectura y monitoreo inalámbrico de sensores en un cuadricóptero a través de una interfaz gráfica, con el objetivo de verificar su correcto funcionamiento a una distancia prudente para evitar cualquier percance con el robot. Adicionalmente, se desea implementar los controladores PID necesarios para volar de manera estable, además de implementar un sistema para monitorear inalámbricamente los valores de los mismos, con el fin de graficarlos y poder ajustar las ganancias de los mismos, mejorando la calidad del vuelo. Este sistema e interfaz serán realizado utilizando ROS y un cuadricóptero real que se construirá durante la asignatura.

II. DESCRIPCIÓN DEL CUADRICÓPTERO

A. Cuerpo y motores

El cuadricóptero tiene un cuerpo de fibra de vidrio, modelo S500 de 480mm con PCB de potencia integrado. Contiene cuatro motores EMax 2212 de 980KV, cada uno equipado con una hélice HobbyKing Slowfly 10x4.5. Con esta combinación de motor y hélice, cada motor logra alrededor de 900 gramos de empuje. Para controlar los motores se dispone de cuatro ESC Racerstar de 30A cada uno.

B. Potencia

El cuadricóptero es alimentado con una batería LiPo Turnigy de 5000 mAh, contiene tres celdas y posee una tasa de descarga de 20C, es decir, puede otorgar hasta 100 amperios. Esta potencia es distribuida utilizando el *Printed Circuit Board* (PCB) que incluye el cuerpo del dron. Para alimentar las unidades de procesamiento, se cuenta con dos reguladores de voltaje de 5V.

¹Grupo de Investigación de Mecatrónica, Universidad Simón Bolívar, Caracas, Venezuela. {13-10982, 13-11349}@usb.ve

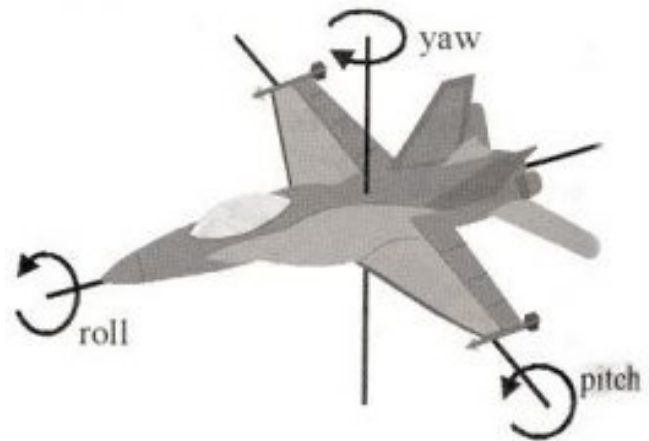


Fig. 1. Ángulos medidos y controlados por el giroscopio y por los controladores PID, respectivamente. Imagen tomada de [1].

C. Sensores

Para conocer los ángulos *yaw*, *pitch* y *roll* del cuadricóptero mostrados en la Figura 1, se dispone de un giroscopio y acelerómetro MPU6050. Para medir la altura, se dispone del medidor de presión atmosférica MS5611 GY-63 con el que haciendo un simple cálculo, se puede transformar la presión atmosférica en altura sobre el nivel del mar. Adicionalmente, se cuenta con un GPS NEO-6M, una cámara de Raspberry Pi, y un sensor de nivel de batería.

D. Unidades de procesamiento

Se cuenta con una Raspberry Pi Zero W como unidad de procesamiento principal y con dos Arduino Nano que se encargan del control y lectura de sensores.

E. Comunicación inalámbrica

El cuadricóptero cuenta con un receptor de radio frecuencia de cinco canales. Además, el Raspberry Pi Zero W posee WiFi integrado, por lo que puede ser conectado a internet, haciendo posible controlar y monitorear al robot a través de un computador ajeno.

III. PROCEDIMIENTO

El primer paso es instalar un sistema operativo compatible con ROS en el Raspberry Pi. El modelo disponible no es compatible con Ubuntu, por lo que se instalará el sistema operativo Raspbian Stretch [2] y ROS Kinetic. Después de esto, se instalarán los paquetes mencionados en la Sección III-B y se probará que ROS funcione de manera remota

al comunicarse con un computador en la misma red. A continuación, se programará la lectura de los sensores en Arduino, se probarán y se publicarán sus datos como tópicos en ROS. Una vez terminada la programación de la lectura de todos los sensores esenciales para el vuelo, se procederá a programar la interfaz gráfica utilizando una aplicación web o una aplicación creada con Node.js y Electron conectada con ROS.

Una vez sea posible monitorear los sensores de una manera segura, se procederá a implementar los controladores PID para controlar los ángulos *yaw*, *pitch* y *roll*. Luego, se publicarán las salidas de los controladores para poder ser graficadas y ajustadas.

A. DIAGRAMA DE BLOQUES

El diagrama de bloques propuesto enseña cómo se comunicarán los datos de los sensores y de dónde son adquiridos. Este diagrama se observa en la Figura 2.

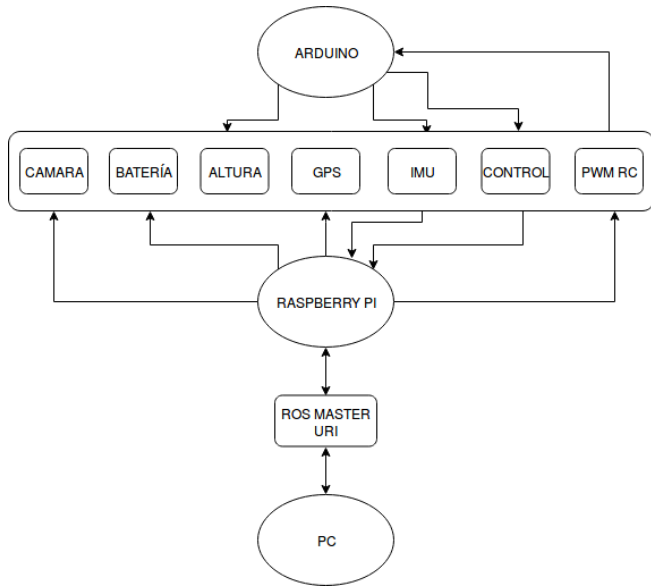


Fig. 2. Diagrama de bloques propuesto para el desarrollo del proyecto.

B. PAQUETES

Se utilizarán los paquetes **roserial** [3] y **roserial_arduino** [4] para crear un nodo en cada Arduino y que puedan publicar tópicos a través del puerto serial que contengan la información de interés.

Para comunicar ROS con la aplicación donde se desarrollará la interfaz gráfica, se utilizará el paquete **rosbridge_suite** [5], que permite la comunicación entre ROS y otros programas a través de una API JSON.

IV. CRONOGRAMA DE ACTIVIDADES

El cronograma de actividades muestra semana a semana el progreso planteado del proyecto. Este se puede detallar en la Tabla I.

TABLE I
DIAGRAMA DE ACTIVIDADES PROPUESTO.

SEMANAS	ACTIVIDADES
1	Instalación y configuración de ROS Kinetic para Ubuntu 16.04 en PC y Raspbian para Raspberry Pi Zero W.
2	Lectura de acelerómetro y giroscopio (MPU6050), lectura de señal RF y pruebas iniciales los motores.
3	Instalación de entorno y librerías necesarias para la comunicación entre el Arduino y Raspberry Pi a través roserial.
4	Implementación de roserial para comunicación entre Arduino y Raspberry Pi.
5	Lectura de sensores disponibles y publicación en tópicos.
6	Establecimiento de comunicación inalámbrica entre PC y Raspberry Pi mediante ROS MASTER URI.
7	Diseño de interfaz gráfica web para el monitoreo de los datos de los sensores.
8	Implementación de controladores PID.
9	Optimización de constantes de los controladores PID.
10	Realización de pruebas de vuelo en ambiente controlado.

REFERENCES

- [1] "Yaw pitch roll image." <http://espygolfapp.com/game-management-how-to-get-better-at-golf/how-to-use-the-golf-team-coaching-app/techniques-of-the-golf-team-coaching-app/yaw-pitch-roll-image/>. Revisado el: 2019-02-15.
- [2] "Download raspbian for raspberry pi." <https://www.raspberrypi.org/downloads/raspbian/>. Revisado el: 2019-02-15.
- [3] "roserial." <http://wiki.ros.org/roserial>. Revisado el: 2019-02-15.
- [4] "roserial_arduino." http://wiki.ros.org/roserial_arduino. Revisado el: 2019-02-15.
- [5] "rosbridge_suite." http://wiki.ros.org/rosbridge_suite. Revisado el: 2019-02-15.