

# Comunicación serial RPi con la STM32 y el filtro complementario

1 <sup>st</sup> Orlando Hernández Martín	2 <sup>st</sup> Juan Camilo Castro Cargas	3 <sup>st</sup> Luis David Beltrán	4 <sup>st</sup> Nicolás Gómez Sierra
Ingeniería Mecatrónica	Ingeniería Mecatrónica	Ingeniería Mecatrónica	Ingeniería Mecatrónica
Universidad ECCI	Universidad ECCI	Universidad ECCI	Universidad ECCI
Bogotá, Colombia	Bogotá, Colombia	Bogotá, Colombia	Bogotá, Colombia
orlando.hernandezm@ecci.edu.co	juanc.castrov@ecci.edu.co	luisd.beltranh@ecci.edu.co	Nicolas.gomez@ecci.edu.co

## I. RESUMEN

En el siguiente informe se hace una explicación breve de la comunicación serial entre las Raspberry con la STM y el filtro complementario. Se hace una introducción al código para la comunicación y por otra parte se explican algunas de las partes donde podemos tener errores integrados tanto el giroscopio como en el acelerómetro, donde nos enfocaremos en algunas variables estadísticas de los tres sensores, donde vamos a tomar el enfoque la estimación de los dos ángulos de Euler.

**Palabras claves**—STM, Filtro complementario, Raspberry, ángulos de Euler, acelerómetro, giroscopio.

## II. INTRODUCCIÓN

En la actualidad cualquier tipo de dispositivo usa diferentes tipos de comunicación, entre dispositivos embebidos la más usada es la comunicación serial, que es un protocolo de comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. puede ser utilizada para adquisición de datos, control, depuración de código, etc.

## III. CONEXIÓN SERIAL IMU, STM Y RASPBERRY

### Definiciones generales, descripción de la práctica y análisis

La conexión serial (paso o transferencia de datos un bit a la vez de manera secuencial y sistemática[1]) es la base para permitir un intercambio de información regulado y constante entre la IMU MPU6050 (sensor), la STM (embebido encargado de la adquisición de datos) y la RASPBERRY (ordenador encargado de la calibración y procesamiento de todos los datos obtenidos), sin intervención por parte de otro ordenador o sistema. Es a través del protocolo I2C (protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. [2]) que se realiza la conexión serial entre los tres dispositivos. El código construido en python posibilita que para la Raspberry, la obtención y calibración de datos provenientes de la Stm y la IMU, sea legible e interpretable a través de gráficas que muestran el comportamiento del dispositivo dentro de los parámetros de medida necesarios. De manera que, tomando como punto de enfoque inicial, la programación de la Raspberry, se observa sus periféricos

y se comprueba su conexión. El puerto de comunicación serial de la raspberry ACM0; /dev/ttyACM0 (.abstract control model"[3]) muestra la conexión y reconocimiento de la stm.

### Acelerómetro no calibrado

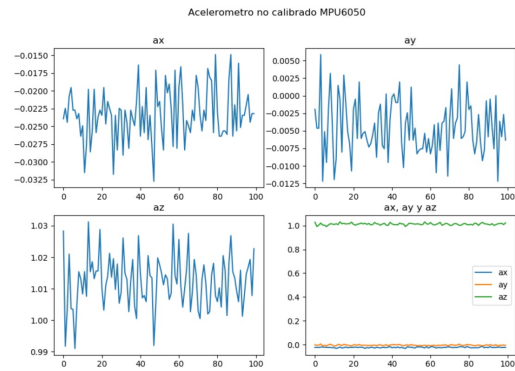


Figura 1. Acelerómetro no calibrado

Clara muestra de ruido, poca visibilidad en los resultados menos notables, siendo estos los que muestran los cambios mas mínimos. A simple vista se nota cuál de las gráficas es la calibrada y cuál es la que no está calibrada se aproximan a los valores de X, Y en 0, Z a 1 gravedad, pero tiene un desfase con la referencia, se hace con muestreo, no en tiempo real.

### Acelerómetro calibrado

En estas graficas se evidencia el movimiento general hecho de derecha a izquierda. En todos los casos se ve el cambio de aceleración en aumento, llegando a un pico y estabilizandose para permanecer constante (muestra que mantiene una misma posición durante el tiempo) hasta percibir la perturbación (que muestra de nuevo un cambio de aceleración). Las dos graficas de la calibrada y la que no está son muy similares, esta se muestra más cerca o ajustados al valor real, en X y en Y se aproximan a 0, en Z casi se encuentra en 1 gravedad, la calibración se hace en la bpi, que no es lo más eficiente.

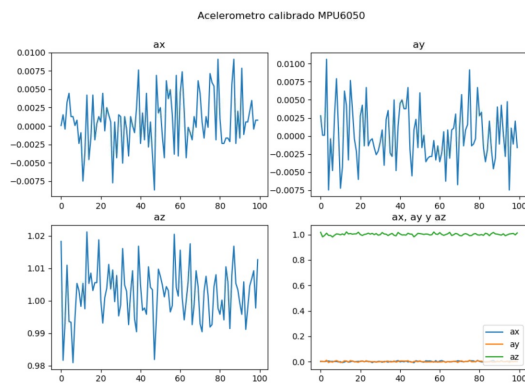


Figura 2. Acelerómetro calibrado

### Giroscopio no calibrado

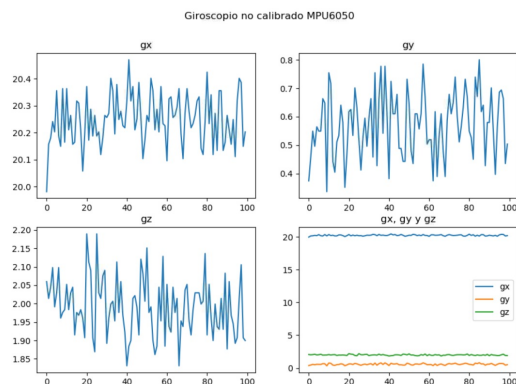


Figura 3. Giroscopio no calibrado

Muy cercano a 0, tiene mucho ruido, nos arroja resultados en grados por segundo, variaciones positivas y negativas, pero esto son resultados erróneos.

### Giroscopio calibrado

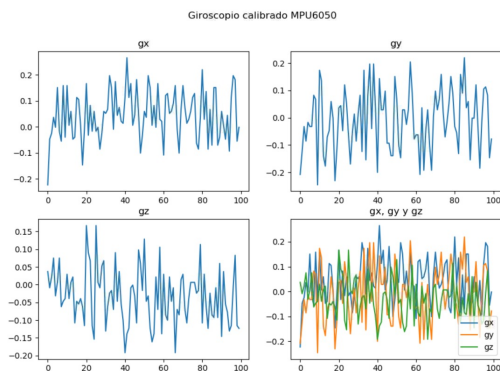


Figura 4. Giroscopio calibrado

En estas graficas se denotan los cambios de posicion de

manera evidente, siendo los unicos picos evidentes. Se debe hacer con la imu en estado de reposo, los 3 ejes están más ajustados al valor de referencia, se puede filtrar para eliminar el ruido para usarlo con fusión sensorial.

### Angulo Pitch

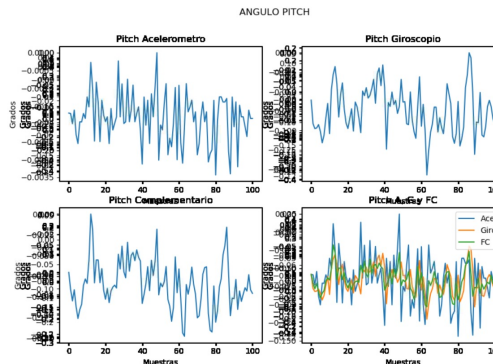


Figura 5. Angulo Pitch

En este se muestra giro en el eje Y, aquí se observan los grados en los cual se encuentra nuestro sensor IMU, en este caso se encuentra en 0 grados ya que está en estado de reposo, hay un poco de ruido, pero este es normal e insignificante, moviéndolo tiene que marcarnos los grados con algo de exactitud, al mover la IMU mientras hay adquisición de datos los errores tienden a 0, cuando se mueve la IMU en algún momento no estabiliza hasta dejarla en total reposo el filtro es un poco lento para estabilizar afectando el angulo ROLL.

### Angulo Roll

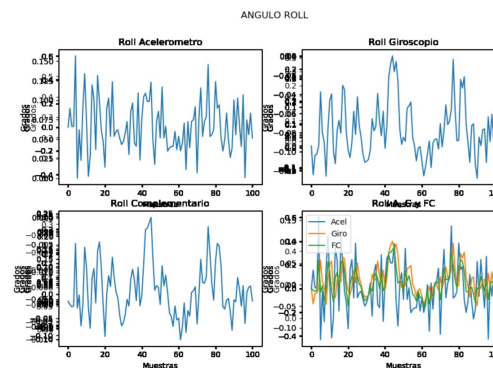


Figura 6. Angulo Roll

En la gráfica .

## IV. CONCLUSIONES

1. El acelerometro es susceptible a los ruidos.

2. Usando los dos sensores (girosocipio y acelerometro) tenemos el roll and pitch, estimando dos de los grados de Euler, sin necesidad de usar el magnetometro.

3. El giroscopio es el sensor principal para los tres ángulos de Euler.

4. Se hace manejo de los ejes del giroscopio manteniendolos lejos del cero y teniendo un buen manejo del acelerometro, para mantener una grafica mas limpia si tanto variación de ruido.

5. La estabilidad de la imu depende del movimiento de los ángulos, una desventaja del filtro complementario es el tiempo de estabilización, además no consigue filtrar de manera eficiente la información para hacer una fusión entre los dos sensores (giroscopio y acelerometro).

#### REFERENCIAS

- [1] "Puerto serial," <https://www.profesionalreview.com/2020/03/07/puerto-serie-que-es-para-que-sirve-y-tipos/>, consultado: 2020-03-24.
- [2] "I2c – puerto, introducción, trama y protocolo," <https://hetpro-store.com/TUTORIALES/i2c/>, consultado: 2020-10-08.
- [3] "Difference between /dev/ttyacm0 and /dev/ttyS0 (arduino ide ports under linux)," <https://arduino.stackexchange.com/questions/19648/difference-between-dev-ttyacm0-and-dev-ttyS0-arduino-ide-ports-under-linux>, consultado: 2020-03-24.