INFORME CORTE No. 3

Sensores IMU – Raspberry Pi – IP–WFI

Pinilla Javier 57837, Palencia Campo 34799

Noviembre de 2019

I. RESUMEN

La siguiente practica a realizar es establecer comunicación entre la unidad IMU, STM32 y la Raspberry Pi, objetivo principal hacer adquisición y procesamiento de datos con los cuales se busca calibrar y obtener los ángulos PITCH y ROLL del acelerómetro y giroscopio, además de eso obtener la visualización de las gráficas de estos sensores calibrados y sin calibrar, de los ángulos y sus ejes correspondientes.

Como proceso final de la Rasberry Pi, debemos poder visualizar el sistema operativo de la tarjeta desde cualquier dispositivo con conexión ethernet por medio de direcciones IP o WIFI.

Palabras Clave – Raspberry Pi, IMU, Aplicativos, Software, Adquirir datos.

II. INTRODUCCIÓN

A través de este documento se representa la información que se realizo en la practica durante las clases, para el complemento de la conexión de la unidad de medición IMU, con la STM32, y Raspberry Pi basados en los programas entregados por el docente, y realizando modificaciones pequeñas en los puertos COM donde se conecto la STM, adicionalmente se realizan modificación en los pines de conexión en la STM y estableciendo la conexión adecuado con la Raspberry Pi entre SCL y SDA entre la STM32F411 y la unidad de medición IMU.

A través del software Matlab, se procesa un programa donde se realiza una lectura de la IMU por medio de la cual se atienen algunos datos para luego mostrar los rangos y ejes de medidas sin calibración y luego de este procedimiento también calibrados pudiendo realizar una comparación de posicionamiento y muestra de los ángulos PITCH y ROLL del sensor acelerómetro y giroscopio.

En este caso se hace la introducción al funcionamiento y empleabilidad el cual por medio de la Raspberry Pi, se hace el procesamiento de los datos de todos los procesos vistos y siguientes explicados por el docente, de esta manera se va realizando el desarrollo de la practica la cual se enfoca en la conexión por modio WIFI a la Raspberry Pi. se descarga el sistema operativo lite de la página web de Rasberry, para instalar en la Placa, para poder utilizar la manipulación del software, la activación de permisos, y la configuración de la Raspberry y lograr poder conectarla con el Computador portátil atravez de un software libre de internet que se llama VNC Server, y en él, poder visualizar en la pantalla del computador, la imagen del sistema operativo instalado en la Rasberry.

III. DESARROLLO DE CONTENIDO

Marco Teórico:

Raspberry PI: es una placa computadora (SBC) de bajo coste, se podría decir que es un ordenador de tamaño reducido, del orden de una tarjeta de crédito, desarrollado en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012. El concepto es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. La Raspberry Pi la han definido como una maravilla en miniatura, que guarda en su interior un importante poder de cómputo en un tamaño muy reducido. Es capaz de realizar cosas extraordinarias.

Raspberry Pi utiliza una arquitectura para el procesador ARM distinta a la que estamos acostumbrados a utilizar en nuestros ordenadores de sobremesa o portátiles. Esta arquitectura es de tipo RISC (Reduced Instruction Set Computer), es decir, utiliza un sistema de instrucciones realmente simple lo que le permite ejecutar tareas con un mínimo consumo de energía.

El diseño de la Raspberry Pi incluye:

- Un Chipset Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos Turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía),
- Un procesador gráfico (GPU) Video Core IV
- Un módulo de 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB).
- Un conector de RJ45 conectado a un integrado lan9512 - jzx de SMSC que nos proporciona conectividad a 10/100 Mbps
- 2 buses USB 2.0
- Una Salida analógica de audio estéreo por Jack de 3.5 mm.
- Salida digital de video + audio HDMI
- Salida analógica de video RCA
- Pines de entrada y salida de propósito general
- Conector de alimentación micro USB
- Lector de tarjetas SD

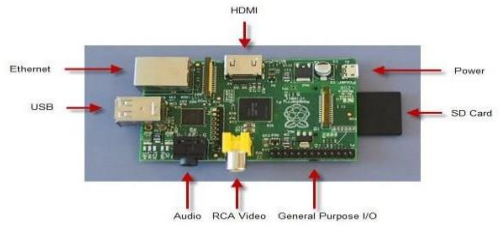


Fig 1 Rasberry Pi.

IMU: Un sensor inercial o también conocido como IMU (unidad de medición inercial) es un componente capaz de obtener la posición, orientación y velocidad de cualquier dispositivo donde sea utilizado. Su construcción puede ser un poco compleja debido a que está compuesto por 3 diferentes sensores. Dentro de este se incorporan giroscopios, acelerómetros y magnetómetros.

Su funcionamiento: cada uno de los sensores agregados aporta una función para lograr un único resultado. El giroscopio se encarga de medir los giros realizados, mientras que el acelerómetro mide la aceleración lineal que se realiza hacia cualquier lado y por último el magnetómetro obtiene información acerca del norte magnético para siempre estar ubicado con respecto al campo magnético de la tierra.

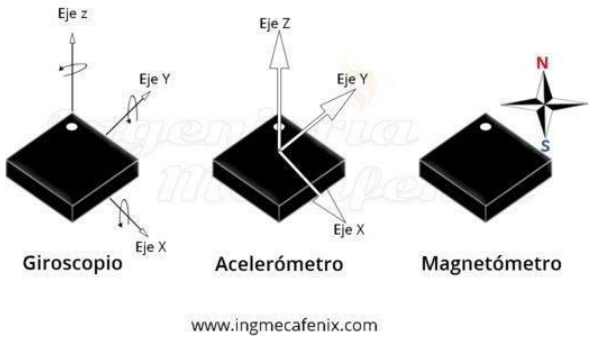


Fig 2 Funcionamiento Sensor IMU

VNC : es un programa de software libre basado en una estructura cliente-servidor que permite observar las acciones del ordenador servidor remotamente a través de un ordenador cliente. VNC no impone restricciones en el sistema operativo del ordenador servidor con respecto al del cliente: es posible compartir la pantalla de una máquina con cualquier sistema operativo que admita VNC conectándose desde otro ordenador o dispositivo que disponga de un cliente VNC portado.

IV. LABORATORIO

Para este laboratorio nos es suministrado un código el cual es dado por el docente de la materia, en el cual compilaba a través del compilador online MBED, lo cual se requiere la configuración de los pines de comunicación I2C en la tarjeta stm32F, y que adicionalmente se modifica el puerto de comunicación, a partir de este primer programa compilado lo cargamos en nuestra STM32, adicionalmente conectamos nuestra STM32 a la raspberry e iniciamos Phyton, para ejecutar programa de lectura y graficación de los ángulos pitch y roll de nuestro sensor inercial IMU, y obtener los ángulos PITCH y ROLL, acontinuación se muestra la calibración del sensor IMU:

```
#Calibrados
for i in range(0,3):
    for j in range(0,raw):
        datos2[j][i+2] = ((datos2[j,i+2])-offsets[i])*SENSITIVITY_ACCEL
        datos2[j][i+5] = ((datos2[j,i+5])-offsets[i+3])*SENSITIVITY_GYRO
print("...datos2 \n",datos2)

h = plt.figure(3)
ax3 = h.subplots(2,2)
h.suptitle('Acelerómetro calibrado MPU6050')
ax3[0,0].plot(datos2[:,0], datos2[:,2])
ax3[0,0].set_title('ax')
ax3[0,1].plot(datos2[:,0], datos2[:,3])
ax3[0,1].set_title('ay')
ax3[1,0].plot(datos2[:,0], datos2[:,4])
ax3[1,0].set_title('az')
ax3[1,1].plot(datos2[:,0], datos2[:,2], label='ax')
ax3[1,1].plot(datos2[:,0], datos2[:,3], label='ay')
ax3[1,1].plot(datos2[:,0], datos2[:,4], label='az')
ax3[1,1].set_title('ax, ay y az')
ax3[1,1].legend(loc='lower right')
h.show()

i = plt.figure(4)
ax4 = i.subplots(2,2)
i.suptitle('Giróscopio calibrado MPU6050')
ax4[0,0].plot(datos2[:,0], datos2[:,5])
ax4[0,0].set_title('gx')
ax4[0,1].plot(datos2[:,0], datos2[:,6])
ax4[0,1].set_title('gy')
ax4[1,0].plot(datos2[:,0], datos2[:,7])
ax4[1,0].set_title('gz')
ax4[1,1].plot(datos2[:,0], datos2[:,5], label='gx')
ax4[1,1].plot(datos2[:,0], datos2[:,6], label='gy')
ax4[1,1].plot(datos2[:,0], datos2[:,7], label='gz')
ax4[1,1].set_title('gx, gy y gz')
ax4[1,1].legend(loc='lower left')
i.show()
```

Y nuevamente realizamos una nueva compilación y hacemos el mismo proceso de lectura de la unidad de medición IMU, y obtenemos unas graficas como las siguientes.

Las graficas se tomaron con el sensor posicionado en paralelo al suelo y fijo en la mesa.

Adicionalmente se implementó una función sensorial, que consiste en la implementación de filtros complementarios para obtener una señal de mejor calidad con un único sensor, como lo muestra acontinuación:

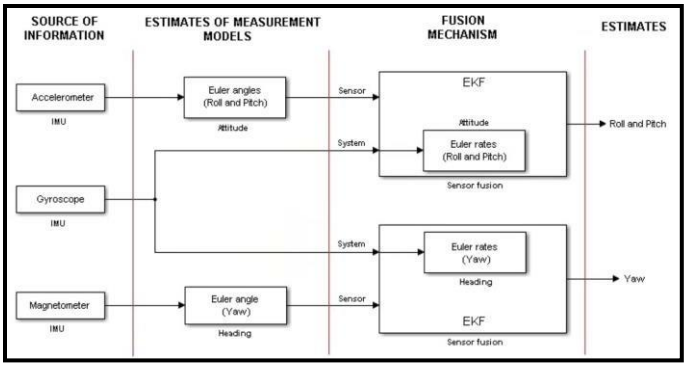
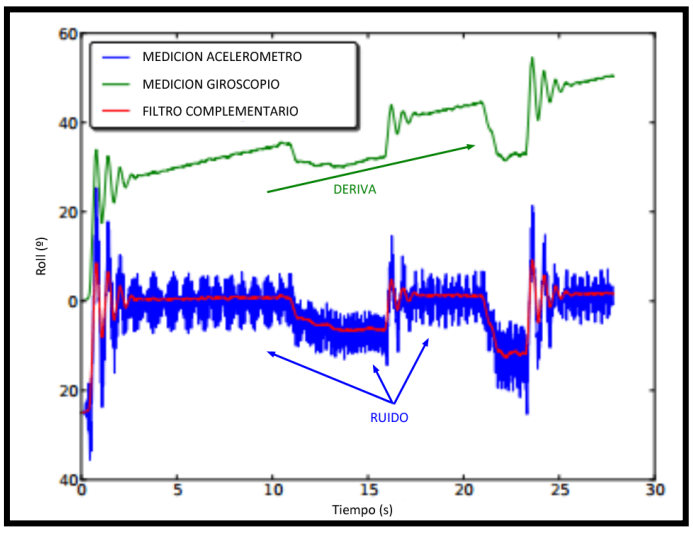


Imagen tomada de Prsentacion Clase12.

En cuanto al Filtro complementario es la utilización y combinación de un filtro pasa Alto y filtro pasa Bajo, teniendo como resultado y comportamiento de un filtro pasa banda, Existen varias formulaciones para un filtro complementario. En su expresión más sencilla, el filtro complementario puede expresarse.

$$\theta = A \cdot (\theta_{prev} + \theta_{gyro}) + B \cdot \theta_{accel}$$

Donde A y B son dos constantes que, inicialmente, puede tomarse 0.98 y 0.02 respectivamente. Podemos calibrar el filtro simplemente variando los valores de A y B siempre que cumplamos la condición de que sumen 1 entre ellos.



El filtro complementario tiene un comportamiento de la siguiente manera: un filtro de paso alto para la medición del giroscopio y un filtro de paso bajo para la señal del acelerómetro.

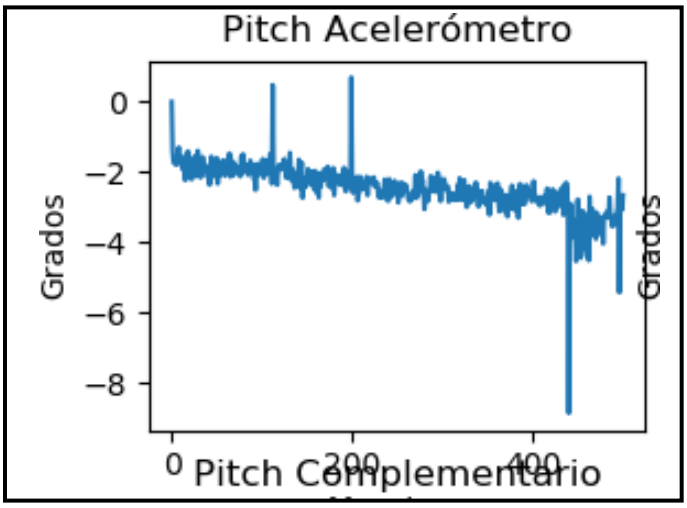
Para la programación de los ángulos PITH y ROLL, se realizo la conversión de radianes a grados, y posterior mente se grafica en eje x la muestra y en el eje Y los grados. Para los dos ángulos obteniendo las 6 graficas adjuntas.

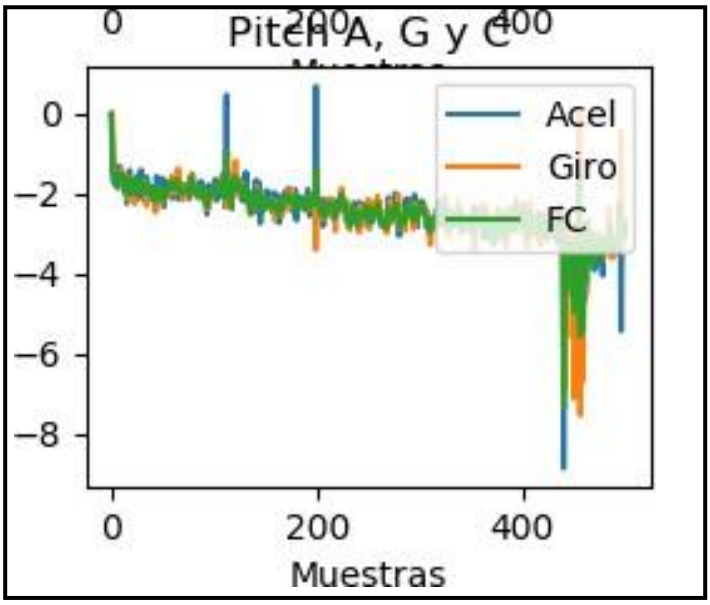
```
#ANGULOS
Roll[raw][0] = raw
Pitch[raw][0] = raw
#Aceletrómetro
for i in range(0,raw):
    Roll[i][0] = i
    Pitch[i][0] = i
    Roll[i+1][1] = (math.atan2(datos2[i,3],datos2[i,4]))*rad2deg
    Pitch[i+1][1] = (math.atan2(-datos2[i,2],pow(((datos2[i,3]*datos2[i,3])+(datos2[i,4]*datos2[i,4])),2)))*rad2deg
    #print("Roll[%d+1][1]: ",i,Roll[i+1][1])
    #Giroscópio
    Roll[i+1][2] = Roll[i][3]+((datos2[i,5]*dt)*rad2deg)
    Pitch[i+1][2] = Pitch[i][3]+((datos2[i,6]*dt)*rad2deg)
    #print("Roll[%d+1][2]: ",i,Roll[i+1][2])
    #Filtro complementario
    Roll[i+1][3] = (A*Roll[i+1][2])+(B*Roll[i+1][1])
    Pitch[i+1][3] = (A*Pitch[i+1][2])+(B*Pitch[i+1][1])
    #print("Roll[%d+1][3]: ",i,Roll[i+1][3])
```

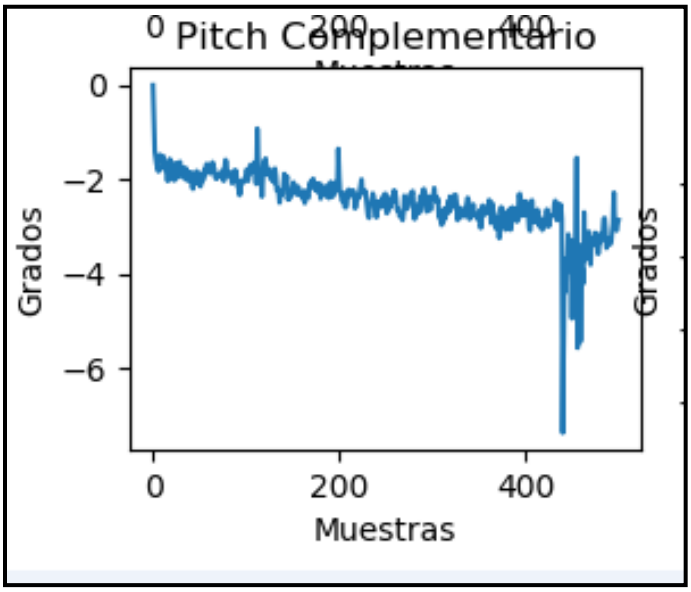
```
j = plt.figure(5)
ax5 = j.subplots(2,2)
j.suptitle('ÁNGULO ROLL')
ax5[0,0].plot(Roll[:,0], Roll[:,1])
ax5[0,0].set_title('Roll Acelerómetro')
ax5[0,0].set_xlabel("Muestras")
ax5[0,0].set_ylabel("Grados")
ax5[0,1].plot(Roll[:,0], Roll[:,2])
ax5[0,1].set_title('Roll Giroscópio')
ax5[0,1].set_xlabel("Muestras")
ax5[0,1].set_ylabel("Grados")
ax5[1,0].plot(Roll[:,0], Roll[:,3])
ax5[1,0].set_title('Roll Complementario')
ax5[1,0].set_xlabel("Muestras")
ax5[1,0].set_ylabel("Grados")
ax5[1,1].plot(Roll[:,0], Roll[:,1], label='Acel')
ax5[1,1].plot(Roll[:,0], Roll[:,2], label='Giro')
ax5[1,1].plot(Roll[:,0], Roll[:,3], label='FC')
ax5[1,1].set_title('Roll A, G y C')
ax5[1,1].set_xlabel("Muestras")
ax5[1,1].set_ylabel("Grados")
ax5[1,1].legend(loc='upper left')
j.show()

k = plt.figure(6)
ax6 = k.subplots(2,2)
k.suptitle('ÁNGULO PITCH')
ax6[0,0].plot(Pitch[:,0], Pitch[:,1])
ax6[0,0].set_title('Pitch Acelerómetro')
ax6[0,0].set_xlabel("Muestras")
ax6[0,0].set_ylabel("Grados")
ax6[0,1].plot(Pitch[:,0], Pitch[:,2])
ax6[0,1].set_title('Pitch Giroscópio')
ax6[0,1].set_xlabel("Muestras")
ax6[0,1].set_ylabel("Grados")
ax6[1,0].plot(Pitch[:,0], Pitch[:,3])
ax6[1,0].set_title('Pitch Complementario')
ax6[1,0].set_xlabel("Muestras")
ax6[1,0].set_ylabel("Grados")
ax6[1,1].plot(Pitch[:,0], Pitch[:,1], label="Acel")
ax6[1,1].plot(Pitch[:,0], Pitch[:,2], label="Giro")
ax6[1,1].plot(Pitch[:,0], Pitch[:,3], label="FC")
ax6[1,1].set_title('Pitch A, G y C')
ax6[1,1].set_xlabel("Muestras")
ax6[1,1].set_ylabel("Grados")
ax6[1,1].legend(loc='upper right')
k.show()
```

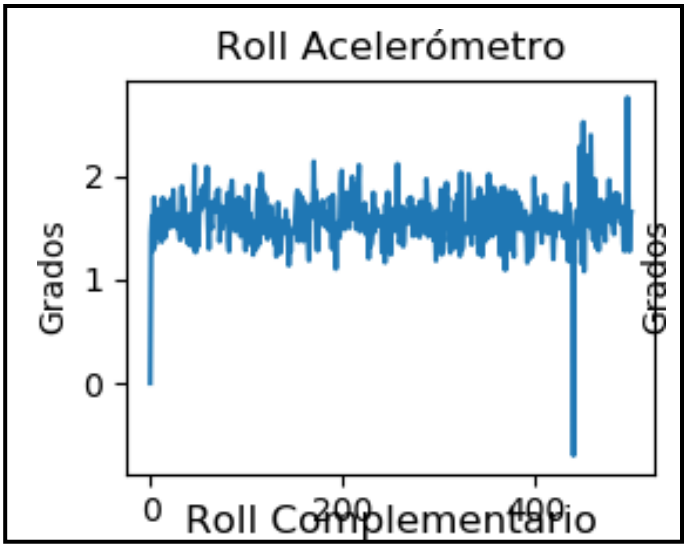
ACELEROMETRO Y GIROSCOPIO AGULO PITCH

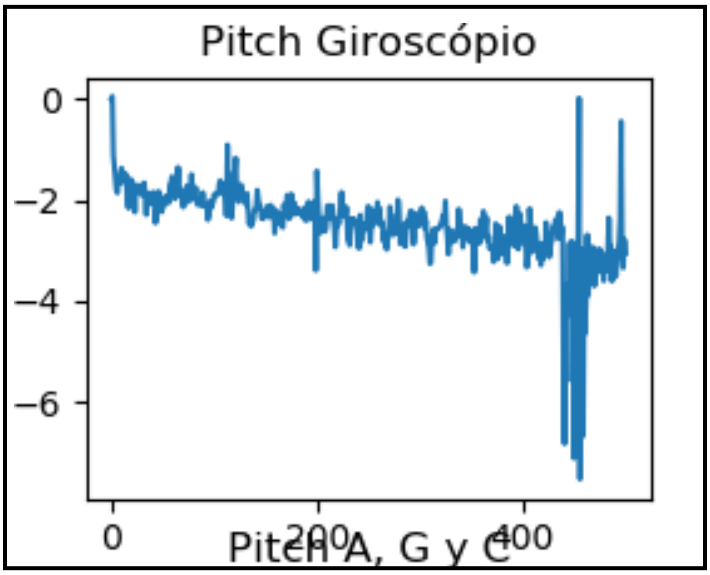


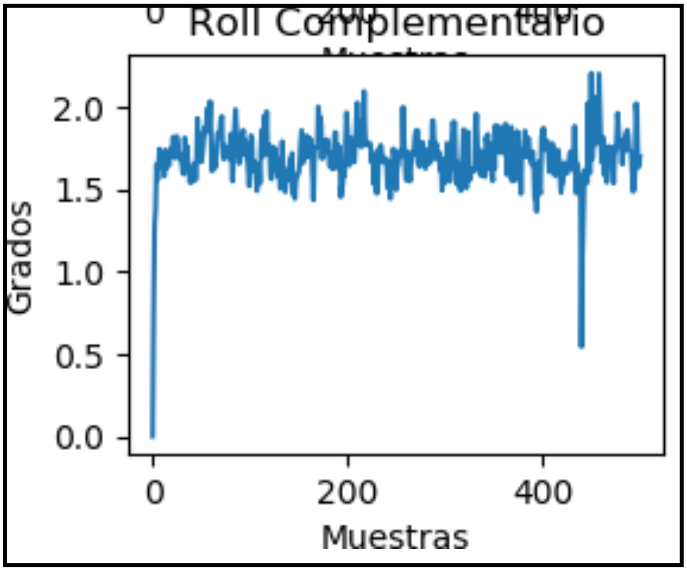


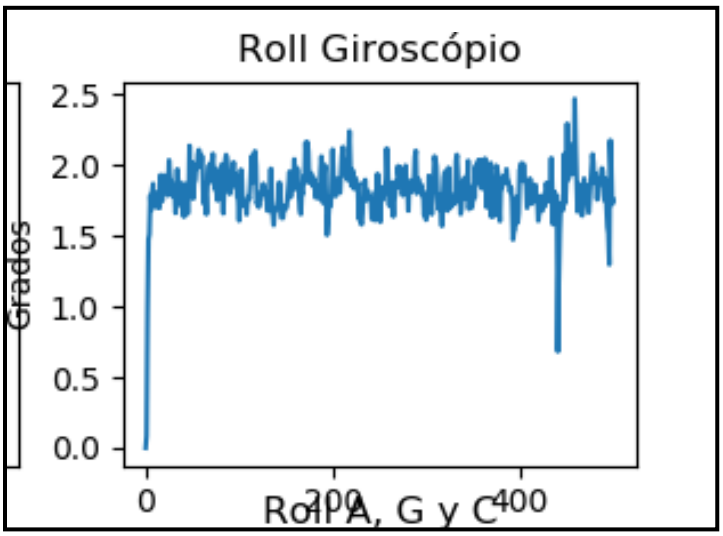


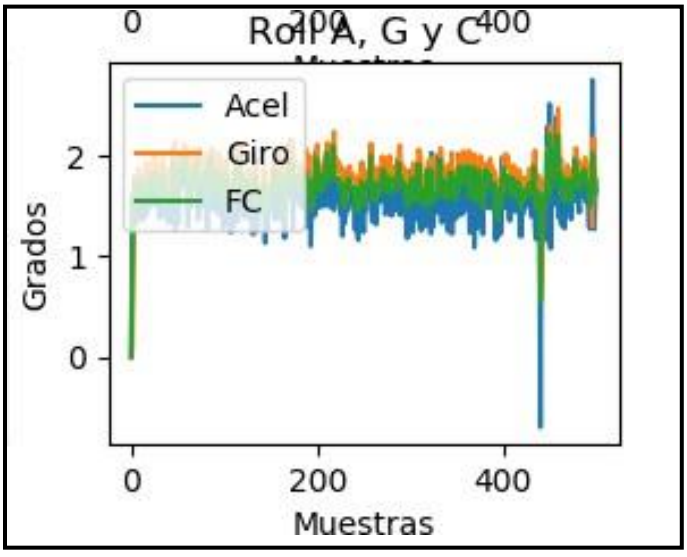
ALGOLO ROLL











V. CONCLUSIONES

Después de realizar la conexión de la IMU a través de la STM32F y evaluar las gráficas obtenidas de los angulso PITCH y ROLL, generando movimientos se ve un variación bastante baja en la medición, ya que el valor se mantiene entre un rango de +/- 0.25° grados dando una precisión bastante alta y confiable

Se genero un error entre la STM y las raspberrry generando bloqueo en el procesamiento, y haciendo que la raspberry no identificara el dispositivo conectado por el puerto USB, bloqueando cualquier proceso adioconal, se utilizo otra raspberry y no se dio solución, se solucionó utilizando una STM diferente y ahí si identifico el dispositivo conectado y adicionalmente se pudo procesora por medio Phyton el programa de lectura adquisición de datos 2Pl y obteniendo como resultado la lectura del sensor IMU.

VI. REFERENCIAS

- "Download Raspbian for Raspberry Pi", *Raspberry Pi*, 2019. [Online]. Available: https://www.raspberrypi.org/downloads/raspbian/.
- "RASPBERRY PI – Historia de la Informática", Histinf.blogs.upv.es, 2019. [Online]. Available: https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/.
- "Home | Mbed", Arm Mbed, 2019. [Online]. Available: https://www.mbed.com/en/.
- "Descargue VNC Viewer | VNC Connect", Realvnc.com, 2019. [Online]. Available: https://www.realvnc.com/es/connect/download/viewer/.
- "Sensor inercial o Sensor IMU - Ingeniería Mecafenix", Ingeniería Mecafenix, 2019. [Online]. Available: https://www.ingmecafenix.com/automatizacion/sensores/sensor-inercial/. [Accessed: 11- Oct- 2019].
- "como conectarse a la raspberry pi por VNC - minitutorial - rpi remotamente", YouTube, 2019. [Online]. Available: https://www.youtube.com/watch?v=09fIAyeJi68.
- Luis Llamas. (2019). Medir la inclinación con IMU, Arduino y filtro complementario. [online] Available at: https://www.luisllamas.es/medir-la-inclinacion-imu-arduino-filtro-complementario/ [Accessed 29 Nov. 2019]