

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

18-6-2021

MPU-6050

Filtro complementario

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

Erick Arellano

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

Unidades de Medición Inercial

Conocidas como IMU por sus siglas en inglés, Inertial Measurement Unit, son dispositivos capaces de medir aceleración, velocidad y orientación de un cuerpo.

Las IMU's se clasifican de acuerdo con los grados de libertad que ofrecen, comúnmente se les dice DOF (por sus siglas en inglés degrees of freedom). Un grado de libertad es la cantidad de magnitudes diferentes que estas unidades son capaces de definir, por ejemplo, los ángulos de posición o la aceleración. Con ello en mente tenemos las siguientes clasificaciones:

- IMU de 6 DOF. Unidades capaces de medir aceleración y ángulos ya que incorporan un acelerómetro y un giroscopio.
- IMU de 9 DOF. Unidades capaces de medir aceleración, ángulos de posición y orientación ya que incorporan un acelerómetro, un giroscopio y un magnetómetro.
- IMU de 10 DOF. Poseen las mismas características que una IMU de 9 DOF pero incorpora además un barómetro para determinar a que altura se encuentra la unidad, existen algunas IMU que erróneamente se clasifican en esta categoría debido a que también miden temperatura sin embargo al no ser una magnitud relevante para medir movimiento o posición no se considera como un grado de libertad.

Acelerómetro y giroscopio

Acelerómetro: Como su nombre lo indica, entrega mediciones de aceleración en los tres ejes, X Y y Z. Debido a la construcción de este circuito es capaz de detectar la aceleración de la gravedad, como sabemos esta magnitud es una constante universal con un valor 9.8 m/s^2 , esto es de gran ayuda pues con eso podemos tener una magnitud como referencia, sin embargo esto también puede representar un problema pues si el sensor es capaz de detectar incluso la aceleración de la tierra, fácilmente puede detectar cualquier alteración en la aceleración de si mismo o del cuerpo sobre el que se encuentre. Debido a que las lecturas que entrega este dispositivo es de aceleración se requiere, de hacer ciertos arreglos matemáticos para poder obtener ángulos.

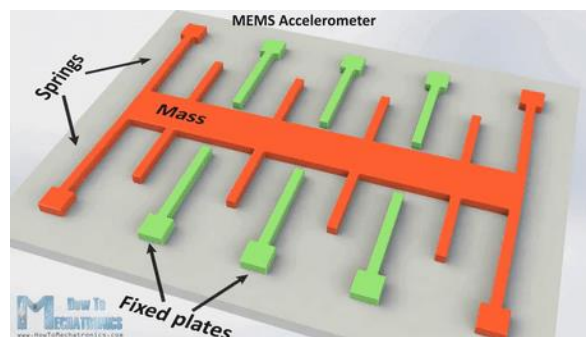


Figura 1: Acelerómetro capacitivo.

Giroscopio: Este sensor entrega ángulos de movimiento por lo cual solo se necesita leer su salida sin hacer ningún cambio. Este dispositivo es muy útil al momento de realizar mediciones de movimientos muy pequeños o movimientos muy rápidos



Figura 2: Groscopio

Cuando estos dos instrumentos trabajan en conjunto podemos obtener mediciones precisas, sin embargo tenemos que hacer ciertos arreglos para que podamos combinar las lecturas de ambos dispositivos y conseguir ángulos de posición.

Obtención de acoplamiento de señales

Para poder obtener ángulos utilizando el acelerómetro es primero pensaremos en un sistema de sólo dos ejes.

Recordando el análisis dimensional sabemos que, si se aplica una fuerza, en este caso una fuerza de aceleración, en el eje X (A_x) y una mas en el eje Y (A_y) obtendremos una resultante con su propio ángulo.

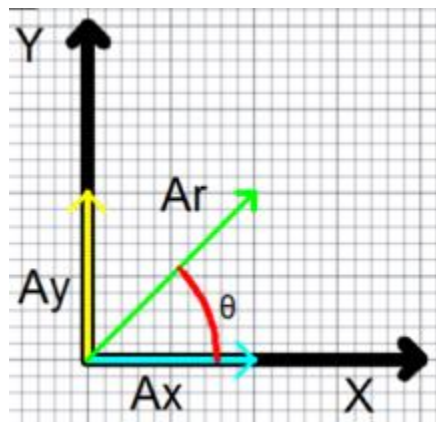


Figura 3: Resultante de dos vectores

Para obtener el valor de la resultante, así como su ángulo se hace uso de las siguientes ecuaciones:

Ecuación 1:
$$Ar = \sqrt{Ax^2 + Ay^2}$$

Ecuación 2:
$$\theta = \tan^{-1} \frac{Ay}{Ax}$$

Debido a que los movimientos que nosotros vamos a medir serán en tres dimensiones haremos uso de los Cosenos directores para obtener el ángulo de movimiento en cada plano. Esta herramienta matemática es utilizada para obtener el ángulo de una resultante en tres dimensiones dada la componente en sus respectivos planos.

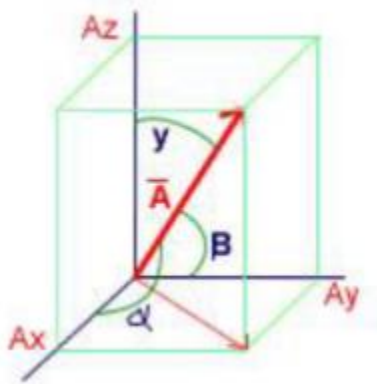


Figura 4: Vectores y resultante en 3D

Para calcular estos ángulos se hace un análisis similar al que se hace en dos dimensiones, primero es necesario calcular la resultante de aplicar tres aceleraciones diferentes en cada uno de los tres ejes y después encontrar los ángulos de acuerdo con sus tres componentes. Para encontrar el valor de la resultante se hace uso de la siguiente ecuación.

Ecuación 3:
$$Ar = \sqrt{Ax^2 + Ay^2 + Az^2}$$

Y para encontrar los tres ángulos (α , β , γ) es necesario hacer uso de las siguientes tres ecuaciones:

Ecuación 4:
$$\alpha = \cos^{-1} \frac{Ax}{Ar}$$

Ecuación 5:
$$\beta = \cos^{-1} \frac{Ay}{Ar}$$

Ecuación 6:
$$\gamma = \cos^{-1} \frac{Az}{Ar}$$

Con estas ecuaciones obtendremos los ángulos utilizando únicamente el acelerómetro.

Para leer los ángulos del giroscopio no hace falta hacer mas que leer los datos en crudo pues ya se entregan los ángulos como tal.

Para poder tener lecturas precisas de movimiento es necesario combinar las señales del giroscopio y del acelerómetro, pero también es necesario filtra estas señales ya que recordemos que en el acelerómetro cualquier cambio o movimiento o incluso una simple vibración generará una variación en las lecturas, es por eso que se requiere de una etapa de filtrado que elimine estas variaciones o bien que sean ignoradas.

Filtrado

En la literatura existen muchos y diferentes tipos de filtros, hasta el momento de los mas precisos y populares el Filtro de Kalman. Dicho filtro fue desarrollado en la década de los 60 por el Ingeniero eléctrico y matemático Rudolf Emil Kalman. Este es uno de los descubrimientos mas importantes en la historia de la ingeniería aeroespacial, ya que esta herramienta matemática fue utilizada como parte del sistema guía de la nave Apolo.

El filtro de Kalman permite realizar una predicción del próximo valor que se va a medir, compararlo con el valor medido y compensar el error entre ambas lecturas y finalmente utilizar estas compensaciones para predecir la siguiente lectura.

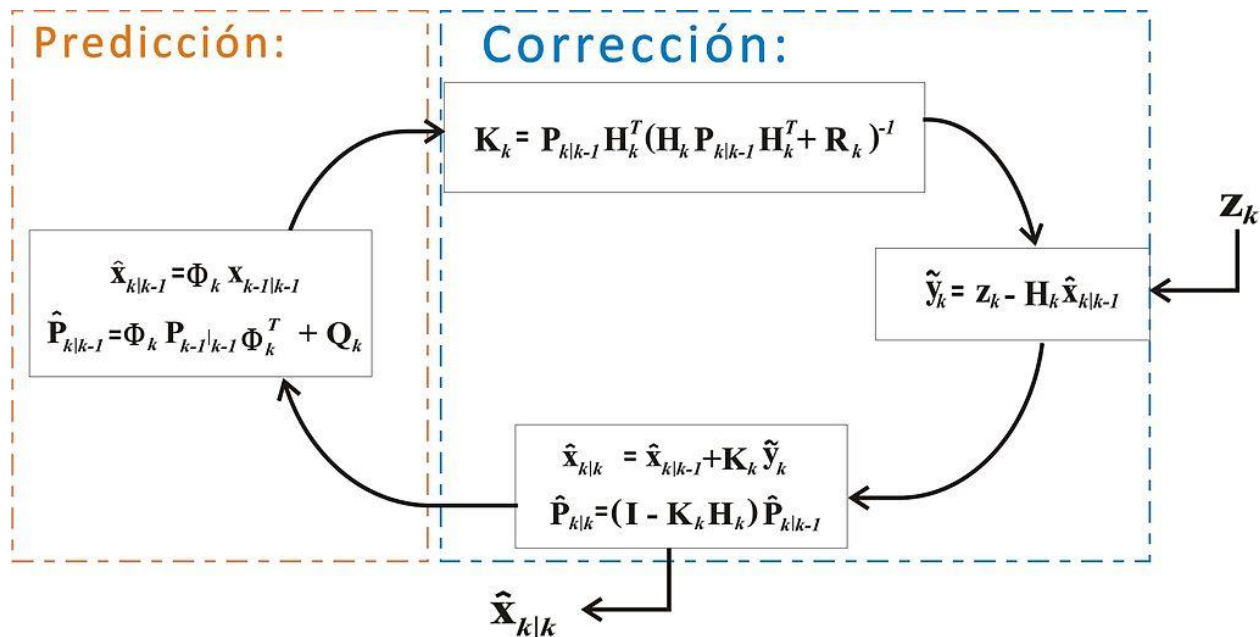


Figura 5: Filtros de Kalman

A pesar de que esta herramienta matemática es muy robusta se vuelve demasiado complicado implementarla ya que nuestro microcontrolador no tiene la memoria ni la capacidad suficiente para poder hacer uso de esta herramienta, para casos como este, se diseñó el filtro complementario.

El filtro complementario es una simplificación del filtro de Kalman que permite combinar las señales del giroscopio y del acelerómetro evitando el ruido que entra en cada una de estas lecturas. Esto se logra combinando un filtro pasa bajas para las señales del acelerómetro y un filtro pasa altas para el giroscopio. El filtro también elimina la predicción a través de la estadística para hacerlo más sencillo. A pesar de que pierde este análisis el filtro complementario es una herramienta bastante precisa y exacta, además de que no requiere de una gran cantidad de recursos computacionales.

La siguiente ecuación representa la forma matemática de aplicar este filtro:

$$\text{Ángulo} = 0.98 * (\text{Ángulo} + (AG * \Delta t)) + 0.02 * AA \quad \dots \text{ecuación 7}$$

Donde:

- Ángulo = Ángulo Filtrado.
- AG = Ángulo Obtenido del Giroscopio.
- AA = Ángulo Obtenido del Acelerómetro.
- Δt = Tiempo transcurrido (en ms) desde la última vez de la aplicación del filtro, recomendado 10 ms.

Esta ecuación se aplica para cada uno de los tres ángulos en cada uno de los ejes.

En la siguiente imagen podemos ver una grafica en la cual se muestra un solo ángulos pero en tres diferentes señales, una para el giroscopio, otra para el acelerómetro una al final que muestra la señal después de aplicar el filtro.

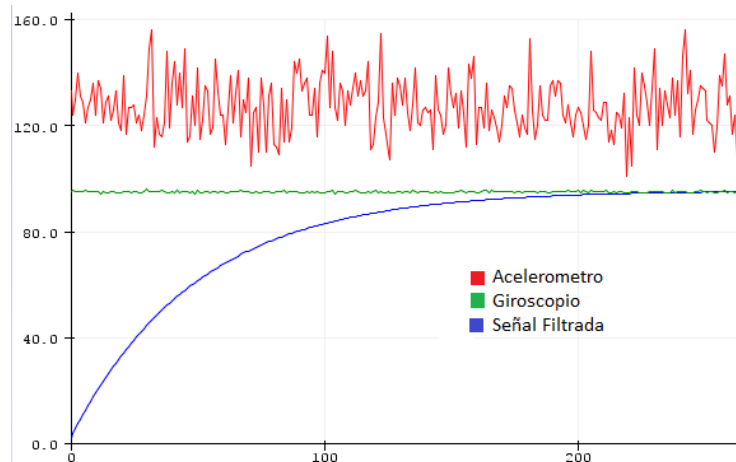


Figura 6: Señales obtenidas

Implementación

MPU5060

Para este trabajo se ha utilizado una IMU de 6 grados de libertad denominada como MPU-6050, incorpora un giroscopio y un acelerómetro, cada uno de 3 grados de libertad.

Este dispositivo trabaja de 3.3 volts a 5, dispone de comunicación I2C para utilizar con diferentes dispositivos como son PIC's, Tarjetas de desarrollo, Arduino, ESP32 e incluso con dispositivos mas avanzados y complejos como Raspberry.

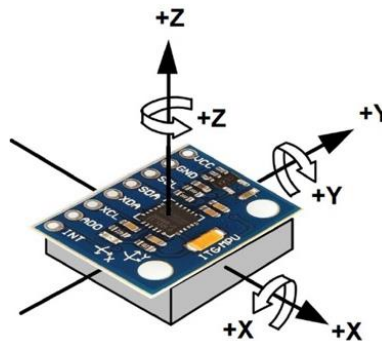


Figura 7: IMU MPU-6050

Arduino Nano

Mientras este proyecto se llevaba a cabo el uso de la tarjeta de desarrollo ESP32 se vió mermado debido a ciertas complicaciones utilizando la comunicación I2C pues después de intentar utilizar el MPU6050 en conjunto con una pantalla OLED y la comunicación serial, la tarjeta dejó de responder, aun se desconoce la razón de la falla y el porque no responde, se realizaron pruebas con una tarjeta Arduino UNO, Arduino NANO y con PIC 18f24j11 y el MPU junto al OLED funcionaban de manera correcta. El conseguir una nueva tarjeta implicaba una cantidad de tiempo de la cual no se disponía, debido a la fecha de entrega de este proyecto.

Por estas complicaciones se hizo uso de la tarjeta Arduino Nano, la cual tiene algunas características similares con la tarjeta ESP32. Arduino Nano está basada en un microcontrolador de la familia AVR, ATmega328P, cuenta con comunicación serial I2C y provee una alimentación de 3.3v, además permite una comunicación serial con una velocidad de has 115200 baudios.



Figura 8: Arduino Nano

Pantalla OLED

La pantalla o display Oled SSD1306 de 128x64 pixeles, con un tamaño de 0.96 pulgadas, es un dispositivo electrónico tipo led, que permite controlar cada píxel individualmente y mostrar tanto texto como gráficos. Además, al ser de tipo OLED no necesita de retroiluminación, lo que hace que su consumo de energía sea mucho menor y posea un mayor contraste.



Figura 9: OLED 124 x 68 SSD1306.

Diagrama de conexión

Como nuestros dispositivos cuentan con capacidad para utilizar la comunicación I2C, se hizo uso de este protocolo para simplificar las cosas, dando como resultado el siguiente diagrama.

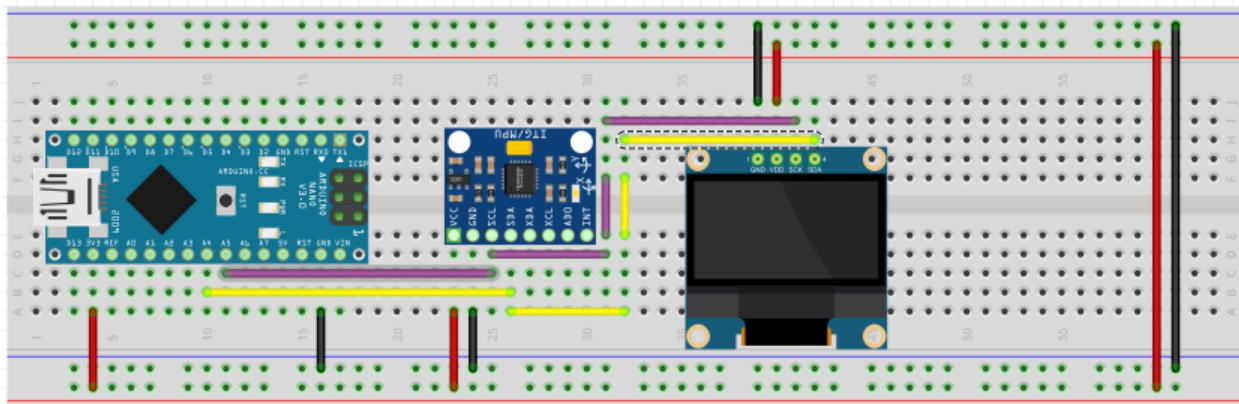


Figura 10: Diagrama del circuito

Una vez que el circuito estaba armado fue necesario realizar la programación del instrumento, para ello se requirió de utilizar las bibliotecas Wire.h la cual habilita la comunicación I2C, y las bibliotecas Adafruit_GFX.h y Adafruit_SSD1306.h que permiten utilizar la pantalla OLED, el siguiente paso fue definir las direcciones I2C de cada uno de los dispositivos además de la declaración de los objetos


```

1 #include <Wire.h>
2 #include <Adafruit_SSD1306.h>
3 #include <Adafruit_GFX.h>
4
5 #define MPU 0x68
6 #define MPU2 0x69
7 #define OLED_RESET 4
8 Adafruit_SSD1306 display(OLED_RESET);
9 #define OLED_address 0x3c

```

Figura 11: Bibliotecas utilizadas

Lo siguiente es hacer la configuración inicial, declaración de variables y comenzar la transmisión de datos en el puerto serial.

```

void setup() {
  Wire.begin();
  Wire.begin();
  display.begin(SSD1306_SWITCHCAPVCC, 0x3c);
  display.clearDisplay();

  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(115200);
}

```

Figura 12: Configuración inicial.

A continuación, comienza el ciclo infinito en el cual se puede ver en el siguiente diagrama:

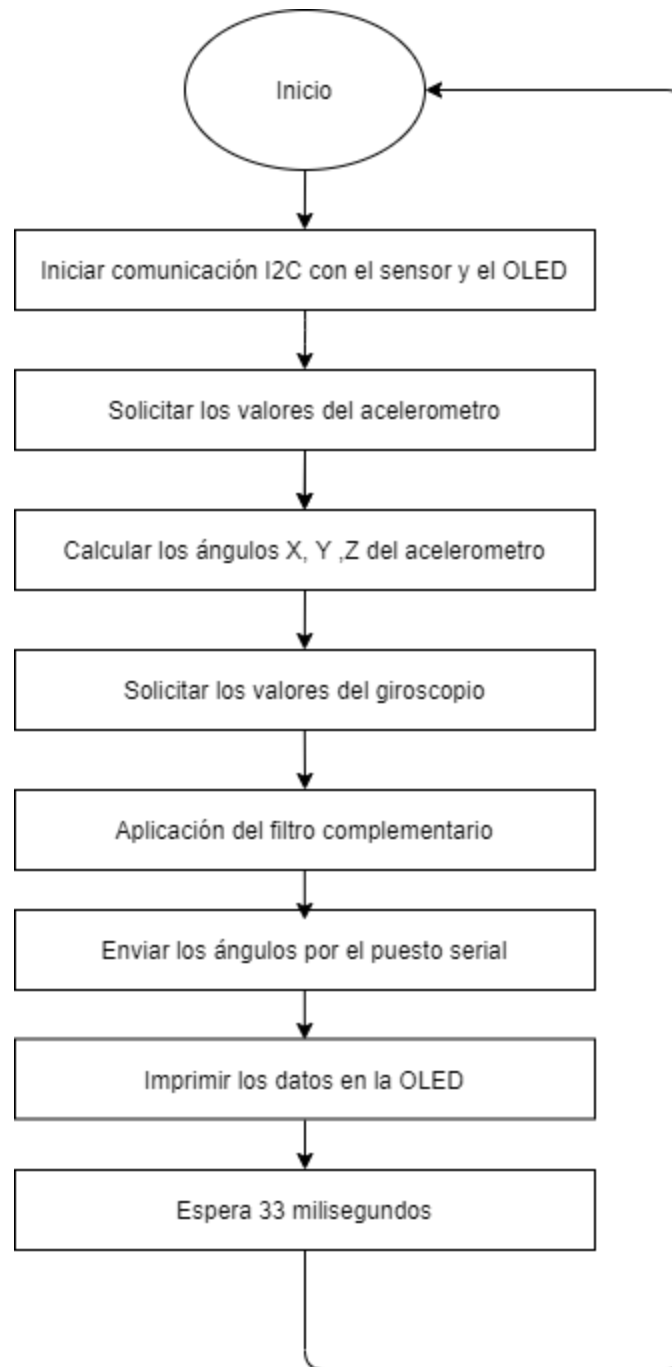


Figura 13: Diagrama del bucle infinito.

Como se puede observar este dispositivo es capaz de medir los ángulos de su posición y mostrarlos en la pantalla oled en caso de que no se cuente con una PC, en caso contrario también es capaz de mandar los datos a través del puerto serial para en nuestro caso funcionar en conjunto con Matlab.

En nuestro programa de Matlab se creó una función que dibujara en cubo y se utilizó el código proporcionado en clase para adquirir los datos del puerto serial y mostrarlos en

gráficas o bien en este caso se imprimen en la ventana de comandos mientras los ángulos van afectando la posición del cubo que previamente se ha dibujado.

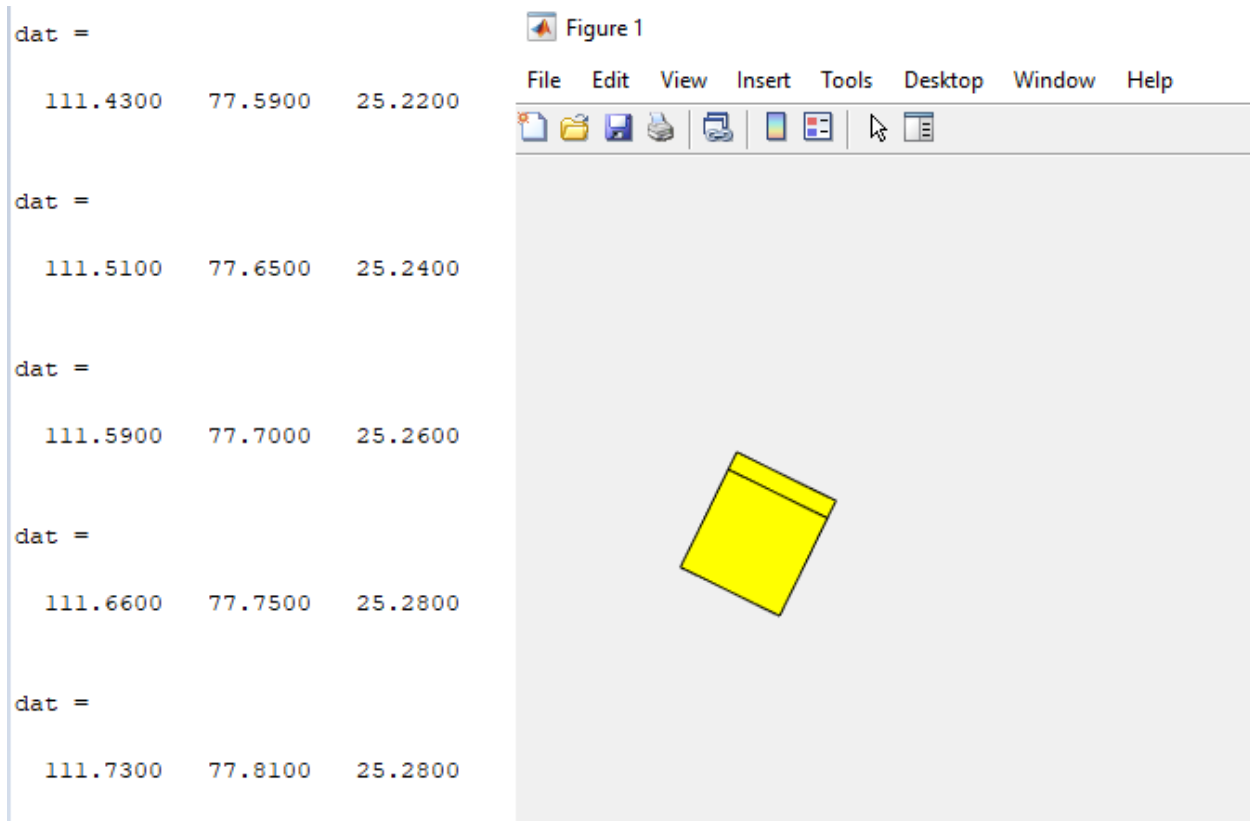


Figura 14: Lecturas en Matlab.

El proyecto posee diferentes consideraciones, las cuales se enumeran a continuación:

- 1) Cuando se enciende el sensor enciende y comienza a enviar datos enviará una lectura de 0° en cada uno de los ejes, dichas lecturas van a ir incrementando su valor, esto se debe a que la primera acción que realiza el sensor es un proceso de calibración en el que establece la posición actual del sensor.

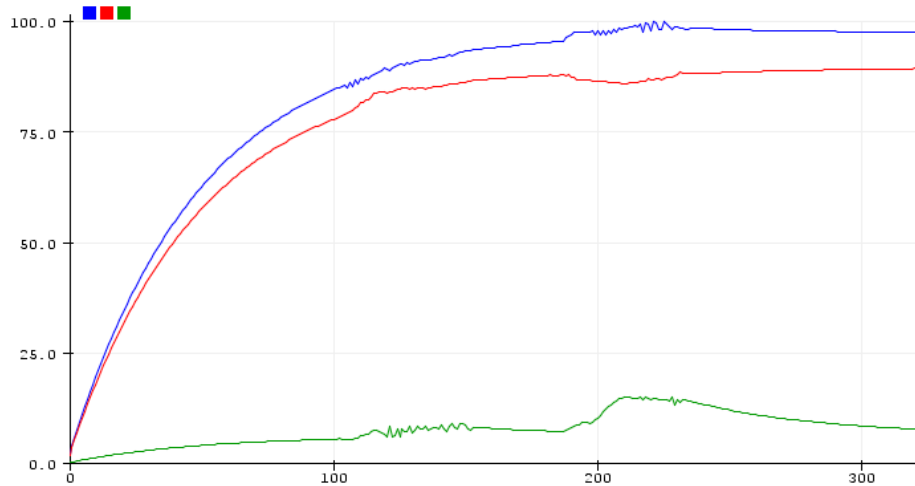


Figura 15: Gráfica de las lecturas del IMU.

- 2) Cuando se utiliza la animación en Matlab para mover el cubo esta se desplaza en dos ejes únicamente, ya que el tercero se ve mermado por la cantidad de grados de libertad que posee nuestro instrumento, ya que los valores de ángulos que se envían no son suficientes para indicar la posición del objeto, en caso de que quisiéramos hacer una animación completa se requeriría una IMU de 9 grados de libertad.

Conclusiones

En este proyecto se consiguió aprender el como utilizar el filtro complementario para conseguir lecturas precisas y correctas sobre la orientación de nuestro dispositivo, a pesar de que involucró un análisis matemático y de diferentes pruebas y errores a lo largo de estos días, ahora es una herramienta útil y fácil de implementar.

Fueron de vital importancia hacer uso de las diferentes habilidades aprendidas a lo largo del curso de Biosensores, como saber implementar la adquisición de datos con el software Matlab o incluso saber interpretar las graficas obtenidas de cada una de las señales recibidas.

Aun posee un largo trabajo camino por recorrer este instrumento, actualmente solo se ha conseguido el como usarla de manera correcta, sin embargo las aplicaciones de este instrumento podrían ser innumerables, además la herramienta matemática utilizada en el MPU-6050 también se podría aplicar en IMU's de 9 grados de libertad para conocer con precisión y exactitud cada uno de los ángulos de orientación del dispositivo.

Referencias

- Suárez, L. (2013). *Introducción a la programación en MATLAB*. Mayagüez, P.R.: Luis E. Suárez.
- Ramírez Giovanni, Presentaciones y artículos de la clase de Biosensores; Instituto de Ingeniería, UNAM, 2021.

- Velasco Graciela, Presentaciones y apuntes de la clase Matemáticas Aplicadas, ICAT, UNAM, 2020.
- Installing ESP32 in Arduino IDE (Windows, Mac OS X, Linux) | Random Nerd Tutorials. (2021). Retrieved 18 June 2021, from <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>
- InvenSense Datasheet MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2
- Vishay Datasheet, 128 x 64 Graphic OLED