



UNIVERSIDAD DEL CAUCA
INGENIERÍA EN AUTOMÁTICA INDUSTRIAL
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL.

**Incidencia del algoritmo de integración de las velocidades angulares en el
proceso de calibración del giróscopo de una unidad de medida inercial (IMU).**

MONOGRAFÍA PRESENTADA COMO REQUISITO PARCIAL PARA OPTAR AL
TÍTULO DE INGENIERO EN AUTOMÁTICA INDUSTRIAL

**CAMILO ANDRÉS BOTINA TRUJILLO
FABIÁN ANDRÉS VERGARA BUSTOS**

Director: Mg. Fabio Realpe
Codirector: PhD. Carlos Felipe Rengifo Rodas

Cauca, 2018

Resumen

En el presente trabajo se diseñó y construyó un sistema mecatrónico compuesto por una plataforma modular con tres grados de libertad y una interfaz gráfica de usuario o GUI, para ajustar los parámetros de calibración de una IMU y reducir los errores agregados por el usuario en el procedimiento manual. El sistema se probó realizando adquisiciones de datos automatizadas para la MPU 6050 y ajustando sus medidas de aceleración lineal y velocidad angular desde la GUI. La prueba consistió en colocar la IMU en la plataforma y realizar diez veces una trayectoria de desplazamiento que contenía tres estados consecutivos: reposo inicial, movimiento en el espacio, reposo final. A partir de los conjuntos de datos adquiridos se ajustó el acelerómetro empleando las lecturas de la aceleración lineal en intervalos estáticos, luego se ajustó el giroscopio usando algoritmos de integración numérica y los valores de la orientación inicial y la velocidad angular entre dos posiciones estáticas consecutivas.

Palabras claves : IMU, acelerómetros, giroscopios, integración numérica.

El presente trabajo va dirigido como una expresión de gratitud a las personas que estuvieron presentes con su voz de aliento y, en especial agradecemos a nuestros padres por su infinito amor y apoyo incondicional.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Trabajos Relacionados	2
1.3. Objetivos	5
1.3.1. Objetivo general	5
1.3.2. Objetivos específicos	5
1.4. Resumen de la propuesta	5
1.5. Estructura del documento.	6
2. Marco Conceptual	7
2.1. Acelerómetro, Giroscopio, Unidad de Medida Inercial (IMU)	7
2.2. Modelo de error del sensor	8
2.3. Esquema para la obtención de datos y el ajuste de los parámetros del modelo de error.	12
2.3.1. Esquema para la obtención de datos	12
2.3.2. Esquema para el ajuste de los parámetros del modelo de error.	13

2.3.3.	Estimación de los parámetros del modelo de error por medio del método Mínimos Cuadrados (LM)	14
2.4.	Representación matemática de la orientación	15
2.4.1.	Matrices de rotación	16
2.4.2.	Ángulos de Euler	17
2.4.3.	Cuaterniones.	18
2.4.3.1.	Propagación de cuaterniones en el tiempo.	20
2.5.	Métodos de integración numérica	21
2.5.1.	Integración de un cuaternión	21
2.5.2.	Tustin y Runge-Kutta	21
2.5.3.	Crouch-Grossman	23
3.	Diseño de la plataforma GIMVEBO	25
3.1.	Requerimientos	25
3.1.1.	Funcionales	25
3.1.2.	Mecánicos	25
3.2.	Diseño y fabricación de la plataforma mecánica.	26
3.2.1.	Fabricación	28
3.3.	Arquitectura hardware del sistema de control y adquisición de datos.	28
3.4.	Elección de la solución electrónica definitiva	31
3.4.1.	Construcción del espacio morfológico	31
3.4.2.	Reducción del espacio Morfológico	33
3.4.3.	Componentes hardware	35

3.5. Inter-conexión de componentes	39
3.6. Solución software.	40
3.6.1. Interfaz gráfica de usuario.	41
3.6.2. Control	42
3.6.3. Adquisición de datos.	43
3.6.4. Ajuste de las medidas.	44
3.6.4.1. Acelerómetro	45
3.6.4.2. Giroscopio	45
4. Resultados	47
4.1. Diagramas de cajas y bigotes	47
4.2. Obtención de datos	49
4.3. Resultados para el acelerómetro	50
4.4. Resultados para el giroscopio	53
4.4.1. Unicidad del cuaternión	53
4.4.2. Desempeño de los algoritmos de integración	55
4.4.2.1. Resultados para RK4, CG4 y Tustin	56
4.4.2.2. Análisis de resultados	58
4.4.3. Runge-Kutta (RK4)	59
4.5. Tiempo de ejecución para RK4, CG4 y Tustin.	61
5. Conclusiones	63

5.1. Trabajos futuros.	65
Anexos	70
A. Análisis estático-movimiento	71
A.1. Análisis estático.	71
A.1.1. Base	75
A.1.2. Brazo 1.	76
A.1.3. Brazo 2	77
A.1.4. Brazo 3	78
A.2. Estudio de movimiento.	78
B. Manual de usuario	80
B.0.1. Instalación de herramientas software.	80
B.0.1.1. ARDUINO IDE	80
B.0.1.2. Robotics_OpenCM IDE	82
B.0.2. Procedimiento de estimación de parámetros.	84
C. Características eléctricas, electrónicas, mecánicas y económicas	86
C.0.0.1. Fuentes de alimentación y potencia.	86
C.0.0.2. Captura	87
C.0.0.3. Movimiento	88
C.0.0.4. Control	89
C.0.0.5. Almacenamiento	90
C.0.0.6. Trasmisión.	91

Índice de figuras

1.1. Sistema de calibración comercial de Acutronic. Extraída de [15]	2
1.2. Doble-octaedro variable geometry. Extraída de [13]	3
2.1. Ejes no ortogonales del sensor acelerómetro o giroscopio (x^s, y^s, z^s), y ejes BF (x^B, y^B, z^B).Extraída de [6].	9
2.2. La relación entre el voltaje de salida del sensor y la variable física medida, se modela como una función lineal, que describe la escala y el bias de los sensores. Extraída de [32].	11
2.3. Proceso de adquisición de datos. Diseño propio	13
2.4. Rotaciones X roll, Y pitch, Z yaw. Diseño propio.	17
3.1. Gimvebo.	26
3.2. Placa y caja de la base.	27
3.3. Brazos.	28
3.4. Diagrama de bloques de la solución Hardware.	29
3.5. Alternativas Hardware para la solución.	32
3.6. Dynamixel XL320	36
3.7. OpenCM 9.04	36
3.8. Arduino pro micro	37

3.9. Step up	37
3.10. Adaptador	38
3.11. Modulo RF HC-11, Bidireccional	38
3.12. Cable Usb.	38
3.13. Inter-conexión de componentes.	39
3.14. Modelo de la solución software.	41
3.15. Interfaz gráfica de usuario.	42
3.16. Diagrama de flujo algoritmo de control.	43
3.17. Diagrama de flujo algoritmo adquisición de datos.	44
3.18. Diagrama de flujo del código del acelerómetro,	45
3.19. Diagrama de flujo del código del giroscopio.	46
4.1. Diagrama de cajas y bigotes. Diseño propio	47
4.2. Aceleración estática.	49
4.3. Velocidad Angular.	50
4.4. Aceleración lineal en el tiempo.	51
4.5. Medidas sin ajustar y ajustadas.	51
4.6. Magnitudes RK4,Tustin,CG4.	54
4.7. Magnitud vs Tiempo (RK4-CG4-Tustin)	54
4.8. Ángulos α obtenido con las medidas sin ajustar y ajustadas RK4.	56
4.9. Ángulos α obtenido con las medidas sin ajustar y ajustadas CG4.	57
4.10. Ángulos α obtenido con las medidas sin ajustar y ajustadas Tustin	58
4.11. Velocidad angular constante y medida.	60

A.1. Cuerpo sometido a Tensión	72
A.2. Desplazamiento	72
A.3. Deformación unitaria	73
A.4. Base	75
A.5. Brazo 1	76
A.6. Brazo 2.	77
A.7. Brazo 3.	78
A.8. Análisis de movimiento	79
B.1. <i>Microsoft Store</i>	81
B.2. <i>Microsoft Store - Arduino IDE.</i>	81
B.3. <i>Arduino IDE.</i>	82
B.4. <i>Página descarga Robotics.</i>	83
B.5. <i>Contenido archivo descargado.</i>	83

Índice de tablas

3.1. Requerimientos de la solución.	40
4.1. Resultados Acelerómetro.	52
4.2. Magnitudes RK4,Tustin,CG4.	55
4.3. Resumen del diagrama de cajas y bigotes para RK4.	56
4.4. Resumen del diagrama de cajas y bigotes para CG4.	57
4.5. Resumen del diagrama de cajas y bigotes para Tustin	58
4.6. Resultados con ω constante-medida.	60
4.7. Tiempos de ejecución para los algoritmos.	62
A.1. Componentes deformación unitaria.	73
C.1. Características fuentes de alimentación.	86
C.2. Características MPU6050	87
C.3. Características motores.	88
C.4. Características tarjetas de control para actuadores	89
C.5. Características tarjetas adquisición y almacenamiento de datos	90
C.6. Características transmisión.	91

Capítulo 1

Introducción

1.1. Motivación

Una Unidad de Medida Inercial (IMU, *Inertial Measurement Unit*) permite determinar la orientación de un cuerpo mediante un grupo de sensores inerciales como el acelerómetro y giroscopio. Muchas aplicaciones contemporáneas dependen en gran medida de este dispositivo, por ejemplo, vehículos aéreos micro (MAV) [1], vehículos aéreos no tripulados (UAV) [2, 3], entre otros.

Este dispositivo requiere un ajuste en los parámetros de calibración antes de ser puesto en funcionamiento, dado que una IMU basada en la tecnología de los Sistemas Micro Electro Mecánicos (MEMS, *Micro Electro Mechanical Systems*) es afectada por bias, imprecisión, desalineación y ortogonalidad en los ejes [4], por lo que errores no deseados se añaden a los datos leídos por la IMU.

El ajuste en los parámetros de calibración de sensores inerciales es una técnica bien conocida en el campo de la navegación inercial [5], en consecuencia nuevos estudios han estado emergiendo durante las últimas décadas, destacándose entre los más recientes los relacionados con el uso de algoritmos de integración numérica en cuaterniones [6, 7]. Los procesos de ajuste de parámetros de calibración requieren una adquisición de datos de aceleración lineal y velocidad angular, que se puede realizar de forma manual o mediante plataformas mecánicas de un grado de libertad, sin embargo, tales procedimientos requieren una manipulación extensa y tediosa por parte del usuario ya que la IMU tiene que ser reposicionada varias veces [8]. En este contexto, el presente trabajo pretende evaluar un algoritmo de integración que preserve

la unicidad del cuaternión y reduzca el esfuerzo en el ajuste de los parámetros de calibración, desarrollando un sistema compacto de 3 grados de libertad para lograr minimizar las interacciones y errores añadidos por el usuario.

1.2. Trabajos Relacionados

En el mercado no existe un sistema económico y compacto específicamente diseñado para el ajuste de los parámetros de calibración de una IMU, en cambio existen sistemas que suelen ser muy costosos y de gran tamaño, por ejemplo, el sistema de calibración de Acutronic [9], ilustrado en la Figura 1.1. También existen sistemas que podrían usarse o adaptarse para realizar la tarea, por ejemplo, un rueda de bicicleta [10], una plataforma de piezas de LEGO [11] o plataformas giratorias [12], sin embargo, todos tienen inconvenientes en la precisión y en el rango de rotación limitada (limitaciones de ángulo que impiden la rotación continua) por el diseño mecánico.



Figura 1.1: Sistema de calibración comercial de Acutronic. Extraída de [15]

Otros ejemplos de plataforma son: *variable-geometry truss* Figura 1.2 y *carpal wrist* de Virginia Tech [13]. Ambas cuentan con un marco accionado por varios actuadores lineales. La cinemática, el control y la construcción de dichas plataformas es bastante complejos. Una ventaja de estas plataformas es que son capaces de producir movimiento lineal, por lo tanto, es posible generar aceleraciones superiores a $9,8 \frac{m}{s^2}$ (1 g), que de otro modo sólo podrían generarse con aceleración centrípeta.

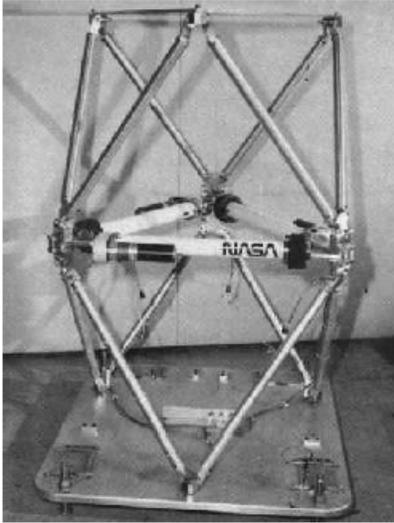


Figura 1.2: Doble-octaedro variable geometry. Extraída de [13]

Las numerosas contribuciones de los diferentes grupos de investigación han permitido que los métodos de calibración de IMUs evolucionen continuamente [14, 15]. En [16] los autores realizan un resumen completo de las diferentes técnicas que se han desarrollado para estimar los parámetros de calibración de una IMU.

En la década de los 90 se desarrolló un procedimiento de calibración donde los giroscopios y acelerómetros de tres ejes son calibrados en un procedimiento que hace uso de la gravedad local y de un conjunto de rotaciones [17], el proceso consiste en colocar la IMU en seis posiciones a lo largo de los ejes para determinar el bias y el factor de escala del acelerómetro y el giroscopio; dicho procedimiento no obtiene los errores de ortogonalidad entre los ejes y no asegura la alineación de los ejes del sensor. En 2002 Shin E y El-Sheimy N amplían el método anterior mediante la incorporación de los parámetros de ortogonalidad y desalineación en el procedimiento de calibración, proponiendo un esquema multiposición, donde las referencias para la calibración del acelerómetro y giroscopio son respectivamente la gravedad y la velocidad de rotación de la tierra, sin embargo estas referencias conducen a problemas de observabilidad [18]. Por otro lado Jurman en 2007 [19] propuso un esquema de calibración donde el giroscopio no requiere un equipo de alta precisión, los datos de velocidad angular se obtienen por medio de una vuelta completa de la IMU alrededor de su eje normal y se emplea un algoritmo de integración para obtener ángulos de orientación a partir de velocidades angulares. En general, el método tiene la desventaja de los algoritmos iterativos que requieren grandes cálculos, además emplea ángulos de Euler para representar la orientación,

lo que conduce al problema de *Gimbal Lock*, fenómeno que consiste en perder un grado de libertad en un espacio de tres dimensiones, y ocurre cuando dos ejes coinciden [20, 21, 22]. El problema de *Gimbal Lock* se puede corregir fácilmente con el uso de cuaterniones que evitan los problemas de singularidad y adicionalmente reducen los recursos computacionales.

Las ventajas de los cuaterniones se extendió a los trabajos propuestos en [6], [23, 24], donde se presenta un procedimiento de calibración para corregir la imprecisión, bias, desalineación y ortogonalidad en los ejes, tanto para el acelerómetro como para la triada del giroscopio. Se establece un procedimiento de múltiples posiciones estáticas para la calibración de la triada del acelerómetro, mientras que la calibración del giroscopio se realiza mediante la integración de las velocidades angulares entre dos posiciones estáticas consecutivas, de esta forma se logra estimar las posiciones de la gravedad en la nueva orientación. La calibración de los giroscopios se obtiene finalmente minimizando los errores entre estas estimaciones y las referencias dadas por la gravedad del acelerómetro calibrado. El método mencionado realiza la calibración mediante 36 y 50 orientaciones distintas que se obtienen manualmente, un proceso que se hace tedioso y lento. Además, para encontrar la condición de estado estático y reducir el efecto del ruido, los datos se promedian durante un periodo de tiempo (*detector cuasi-estático*). Aunque estos esquemas no requieren ninguna plataforma de calibración, la precisión puede verse afectada por las perturbaciones externas y el diseño óptimo del esquema de rotación. Particularmente en [23], se utiliza la gravedad y el campo magnético del ambiente como referencias para calibrar el giroscopio. Este método estima los parámetros de error del giroscopio mediante un algoritmo evolutivo. No obstante, este esquema tiene diferentes referencias cinemáticas y puede ser fácilmente afectado por las variaciones en el entorno magnético.

Los métodos descritos anteriormente tienen en común que usan un algoritmo de integración en cuaterniones para la calibración del giroscopio, sin embargo, el algoritmo no preserva la unicidad del cuaternión, por lo que pueden generar errores en la estimación de la orientación seguidos por errores en los parámetros de ajuste de la medida. Adicionalmente, se realiza un procedimiento de re-normalización que puede conllevar a un comportamiento inestable después de un tiempo de integración [25].

El método para estimar los parámetros de calibración que se empleará en el presente trabajo, consiste en realizar múltiples posiciones estáticas en la calibración de la triada del acelerómetro, mientras que el ajuste de los parámetros de calibración del giroscopio se realiza integrando las velocidades angulares entre dos posiciones estáticas consecutivas [6].

1.3. Objetivos

1.3.1. Objetivo general

El objetivo principal de este proyecto es determinar la incidencia del algoritmo de integración de las velocidades angulares en la estimación de los parámetros de calibración del giroscopio de una unidad de medida inercial (IMU).

1.3.2. Objetivos específicos

1. Determinar la capacidad de preservación de la norma de un cuaternión para diferentes algoritmos numéricos de integración cuando se aplican a medidas de velocidad angular.
2. Desarrollar una plataforma hardware-software para la calibración automática del giroscopio y del acelerómetro de una unidad de medida inercial (IMU).
3. Elaborar un análisis comparativo entre los resultados de la orientación estimada de una misma IMU calibrada y sin calibrar, utilizando distintas trayectorias de movimiento y diferentes algoritmos de integración de velocidades angulares.

1.4. Resumen de la propuesta

El presente trabajo consiste en estimar los parámetros de calibración de un acelerómetro y un giroscopio de una unidad de medida inercial IMU de referencia MPU 6050, utilizando una plataforma mecánica de tres de grados de libertad, e implementando algoritmos de integración numérica en cuaterniones de velocidad angular. La plataforma mecánica fue diseñada en SolidWorks[®] [26] un software de Diseño Asistido por Computadora (CAD, *Computer Assisted Design*) para modelado mecánico en 2D y 3D. El algoritmo y la interfaz gráfica se implementaron en MatLab[®] [27]. La interfaz permite al usuario observar los parámetros óptimos y los resultados obtenidos después del ajuste en los parámetros de calibración.

Finalmente, se determinó la incidencia del algoritmo de integración de las velocidades angulares en la estimación de los parámetros de calibración del giroscopio y, la capacidad de preservación de la norma de un cuaternión en los algoritmos clásicos de integración numérica cuando se aplican a medidas de velocidad angular.

1.5. Estructura del documento.

Este trabajo está dividido en cinco capítulos (incluido el presente) que siguen una traza lógica y que van desde la base teórica hasta la implementación y verificación de los algoritmos de integración numérica. En el **Capítulo II** se tratan los aspectos claves para que el lector pueda alcanzar una comprensión detallada del proceso que se lleva a cabo en el ajuste de los parámetros de calibración de la IMU, incluidos los algoritmos de integración y el modelo de error del sensor. En el **Capítulo III** se detallan las características del sistema de adquisición de datos, se incluye el diseño mecánico y la arquitectura hardware y software. En el **Capítulo IV** se evalúa la capacidad de preservar la norma de un cuaternión para diferentes algoritmos numéricos de integración cuando se aplican a medidas de velocidad angular y, se realiza un análisis comparativo de los resultados de la orientación estimada para la misma IMU, utilizando medidas ajustadas y sin ajustar, distintas trayectorias de movimiento y diferentes algoritmos de integración de velocidad angular. En el **Capítulo V** se presentan las conclusiones obtenidas después del estudio y se plantean las líneas de investigación del proyecto y algunas propuestas para trabajos futuros.

Capítulo 2

Marco Conceptual

Este capítulo presenta la teoría necesaria para el desarrollo de los capítulos posteriores. En la sección 2.1 se definen los conceptos básicos asociados a la IMU y a sensores inerciales. En la sección 2.2 se define el modelo de error para el conjunto acelerómetro - giroscopio. En la sección 2.3 se define el esquema de obtención de datos y ajuste de parámetros de calibración, además la técnica empleada en la estimación de los parámetros. En la sección 2.4 se presentan, con sus respectivas ventajas y desventajas, los modelos matemáticos para representar la rotación. Finalmente, en la sección 2.5 se resumen los métodos de integración numérica empleados en el presente trabajo.

2.1. Acelerómetro, Giroscopio, Unidad de Medida Inercial (IMU)

La MPU6050 es un sistema que permite medir velocidad, rotación y fuerzas gravitacionales de un cuerpo a través de la conjunción de sensores inerciales, como el *acelerómetro* y el *giroscopio*, los cuales miden respectivamente, la aceleración lineal y la velocidad de giro respecto a un eje [28]. Este tipo de sensores se encuentran en teléfonos inteligentes, juegos de vídeo, controles remotos y, en muchas otras aplicaciones cotidianas, razón por la que se han convertido en tendencia tecnológica mundial.

Antes de analizar las medidas que proporciona un acelerómetro o giroscopio, se debe saber que la información la envían usando un protocolo de comunicación específico como I2C, SPI

o UART, en este proyecto utilizaremos el puerto I2C. Además, la salida es un nivel de tensión dentro de un rango predefinido que deberá convertirse en un valor digital usando un módulo de Conversión Analógica-Digital (ADC, *Analog to Digital Converter*) [29].

La ecuación 2.1 permite obtener la lectura del acelerómetro en unidades de gravedad (m/s^2).

$$Aceleración_{xyz} \left(\frac{m}{s^2} \right) = (4g/2^{16}) [Ax Ay Az] \quad (2.1)$$

El vector $[Ax Ay Az]$ corresponde a la lectura de aceleración en cada eje, 2^{16} es el rango máximo en la salida de la MPU6050 que tiene un módulo ADC de 16 bits, $4g$ es el rango de escala máximo configurado por defecto en la librería *I2Cdevlib*.

Simplificando la ecuación 2.1 se obtiene que:

$$Aceleración_{xyz} \left(\frac{m}{s^2} \right) = (g/2^{14}) [Ax Ay Az] \quad (2.2)$$

El giroscopio emplea el mismo módulo ADC del acelerómetro y la salida es un valor que está linealmente relacionado con la tasa de cambio de un ángulo.

La ecuación 2.3 permite obtener la lectura del giroscopio en las unidades ($^{\circ}/s$)

$$Velocidad\ angular\ (^{\circ}/s) = ((500^{\circ}/s)/2^{16}) [Gx Gy Gz] \quad (2.3)$$

El vector $[Gx Gy Gz]$ corresponde a la lectura de giroscopio en cada eje, $500^{\circ}/s$ es el rango de escala máximo configurado por defecto en la librería *I2Cdevlib*.

2.2. Modelo de error del sensor

En una IMU ideal los ejes del acelerómetro y del giroscopio tienen que definir y compartir un único marco de referencia ortogonal. El acelerómetro detecta la aceleración y el giroscopio mide la velocidad angular a lo largo de sus respectivos ejes. Sin embargo, debido a la inexactitud en la construcción de una IMU real, las dos tríadas forman dos tramas distintas, es decir, están mal alineadas, también los sensores individuales no son perfectos debido a que sus ejes

no son ortogonales.

Para describir el modelo de error se tienen en cuenta las consideraciones propuestas en [6].

- La triada no ortogonal y ortogonal del acelerómetro se designan respectivamente por las siglas AF y AOF. Para el giroscopio solo se cambia AF por GF y AOF por GOF.
- La triada ortogonal que representa, por ejemplo las coordenadas del chasis de la IMU, se define por la sigla BF.
- El eje “x” de AOF y AF coinciden.
- El eje “y” de AOF se encuentra en el plano abarcado por los ejes “x” e “y” de AF.

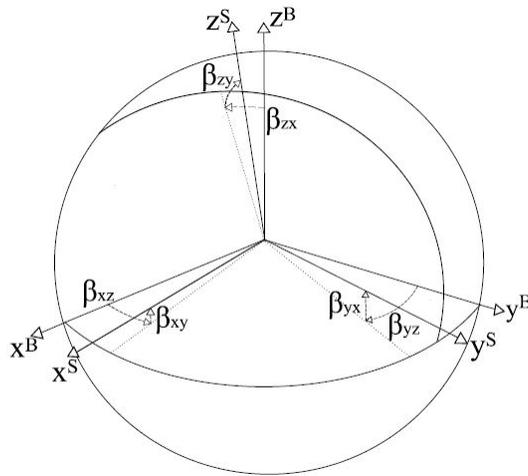


Figura 2.1: Ejes no ortogonales del sensor acelerómetro o giroscopio (x^s, y^s, z^s), y ejes BF (x^B, y^B, z^B). Extraída de [6].

El buen funcionamiento de una IMU implica que los ejes de los sensores inerciales deben ser ortogonales, pero debido a la imprecisión en la construcción de la IMU esto rara vez ocurre. Entonces si el marco de coordenadas BF difiere del marco AF y GF en ángulos pequeños, como se ilustra en la Figura 2.1, las medidas de aceleración o velocidad angular en una marco no ortogonal (AF o GF) se pueden transformar en medidas estimadas en el marco ortogonal BF con la ecuación 2.4. (Más detalles pueden ser consultados en [30, 31]).

$$s^B = T s^S, \quad T = \begin{bmatrix} 1 & -\beta_{yz} & \beta_{zy} \\ \beta_{xz} & 1 & -\beta_{zx} \\ -\beta_{xy} & \beta_{yx} & 1 \end{bmatrix} \quad (2.4)$$

Donde s^B y s^S corresponden respectivamente, a la aceleración (o velocidad angular) en el marco de coordenadas BF y en el AF (o GF). La variable β_{ij} es la rotación del i –ésimo eje del acelerómetro (o giroscopio) alrededor del j –ésimo eje del BF. (Ver Figura 2.1).

En el método para estimar los parámetros de ajuste de las medidas de aceleración y giro de una IMU se supone que el marco de coordenadas BF coincide con el marco ortogonal AOF del acelerómetro, en tal caso, los ángulos xz , xy , yx son cero, entonces en el caso de los acelerómetros la Eq. 2.4 se convierte en:

$$a^O = T^a a^S, \quad T^a = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ 0 & 1 & -\alpha_{zx} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

En la Eq. 2.5 se cambió la letra β referida al caso general por la letra α referida al caso del acelerómetro, mientras que a^O y a^S denotan la aceleración en AOF y AF, respectivamente.

Como se mencionó anteriormente, el giroscopio y el acelerómetro deberían referirse al mismo marco de referencia ortogonal, en nuestro caso AOF. Entonces, usando la Eq. 2.4, para el giroscopio se tiene la Eq 2.6.

$$\omega^O = T^g \omega^S, \quad T^g = \begin{bmatrix} 1 & -\gamma_{yz} & \gamma_{zy} \\ \gamma_{xz} & 1 & -\gamma_{zx} \\ -\gamma_{xy} & \gamma_{yx} & 1 \end{bmatrix} \quad (2.6)$$

En la Eq. 2.6, ω^O y ω^S denotan la velocidad angular en AOF Y GF, respectivamente.

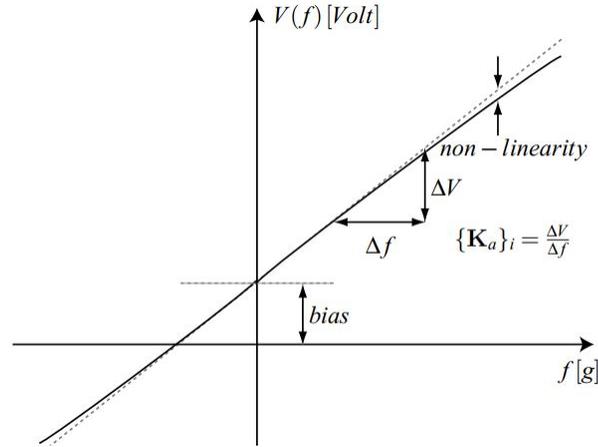


Figura 2.2: La relación entre el voltaje de salida del sensor y la variable física medida, se modela como una función lineal, que describe la escala y el bias de los sensores. Extraída de [32].

Los sensores MEMS emiten un voltaje proporcional a las cantidades físicas detectadas (aceleración y velocidad angular). La relación entre la tensión de salida y la cantidad física que actúa a lo largo de los ejes del sensor está dada por la hoja de datos del fabricante, pero la escala real varía entre diferentes muestras y señales de entrada (debido a las no linealidades inherentes de los sensores) [32]. Además, a menudo hay un pequeño off-set o bias en la señal de salida del sensor, es decir, aunque no haya fuerza actuando sobre el sensor se produce una salida distinta de cero, conforme ilustrado en la Figura 2.2. Por lo anterior, para el conjunto acelerómetro-giroscopio se define en la Eq. 2.7 la matriz diagonal del factor de escalabilidad que relaciona la entrada y salida del sensor y, en la Eq. 2.8 el bias.

$$K^a = \text{diag}(S_{xa}, S_{ya}, S_{za}), \quad K^g = \text{diag}(S_{xg}, S_{yg}, S_{zg}) \quad (2.7)$$

$$b^a = \begin{bmatrix} b_{xa} \\ b_{ya} \\ b_{za} \end{bmatrix}, \quad b^g = \begin{bmatrix} b_{xg} \\ b_{yg} \\ b_{zg} \end{bmatrix} \quad (2.8)$$

El modelo completo de error del sensor para el acelerómetro es:

$$a^O = T^a K^a (a^S + b^a + v^a) \quad (2.9)$$

y para el giroscopio:

$$\omega^O = T^g K^g (\omega^S + b^g + v^g) \quad (2.10)$$

En las ecuaciones 2.9 y 2.10, v^a y v^g son la medida del ruido en el acelerómetro y el giroscopio, respectivamente. Sin embargo, el ruido se descarta en las mediciones porque en el procedimiento de ajuste de los parámetros de calibración se promedian las señales.

2.3. Esquema para la obtención de datos y el ajuste de los parámetros del modelo de error.

2.3.1. Esquema para la obtención de datos

Físicamente el experimento consiste en adquirir las medidas de velocidad angular y aceleración lineal que entrega la IMU cuando se encuentra en:

1. Una posición estática inicial.
2. Movimiento, traducido a rotaciones en el espacio.
3. Una posición estática final.

El experimento se repite 10 veces para obtener un conjunto amplio de datos. El proceso es ilustrado en la Figura 2.3.

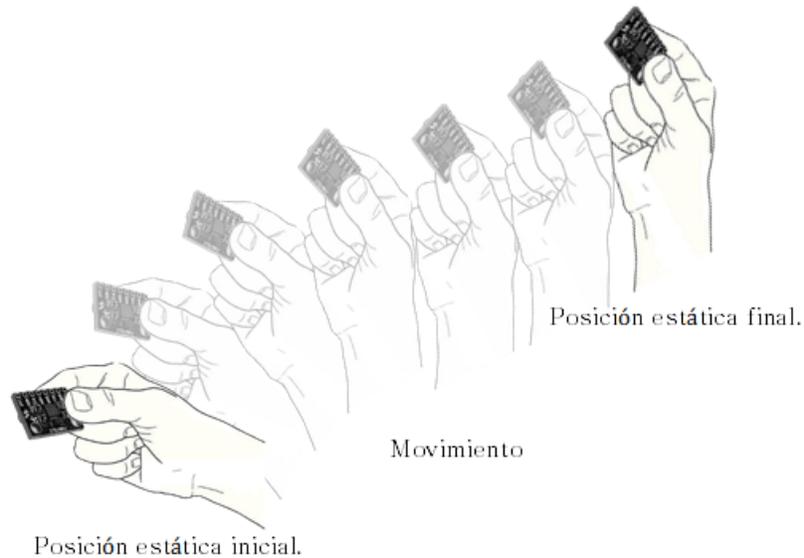


Figura 2.3: Proceso de adquisición de datos. Diseño propio

2.3.2. Esquema para el ajuste de los parámetros del modelo de error.

Una vez conformado el conjunto de datos con el que se trabaja, se adoptan las dos hipótesis formuladas en [24] para facilitar el método de ajuste de parámetros de calibración, estas establecen las siguientes referencias físicas y matemáticas:

- **Para el acelerómetro, Propiedad 1:** La magnitud de la aceleración estática medida en la IMU debe ser igual a la gravedad local.
- **Para el giroscopio, Propiedad 2:** El vector de orientación medido por el acelerómetro en el intervalo estático final debe ser igual al vector de orientación estimado mediante la integración numérica.

A partir de las dos propiedades mencionadas, se infiere que el procedimiento para estimar los parámetros de calibración es secuencial, es decir, primero se ajustan las medidas del acelerómetro y después las del giroscopio.

Resumiendo, los parámetros de ajuste de la aceleración lineal se estiman empleando un procedimiento de múltiples posiciones estáticas y el vector físico de gravedad como referencia (*Propiedad 1*), mientras que para el giroscopio, la posición del vector de gravedad en la orien-

tación final se estima mediante la integración de la orientación inicial y la velocidad angular entre dos posiciones estáticas consecutivas. Los parámetros del giroscopio se obtienen finalmente utilizando un algoritmo de mínimos cuadrados que minimiza los errores entre las estimaciones y, las referencias dadas por las medidas del acelerómetro ajustadas. (*Propiedad 2*).

2.3.3. Estimación de los parámetros del modelo de error por medio del método Mínimos Cuadrados (LM)

La estimación de los parámetros de calibración de los sensores se puede abordar como un problema de optimización de una función objetivo. Si lo que se desea lograr es una mayor exactitud en las mediciones, entonces dicha función debe estar relacionada con el error entre la medición y el valor real. Por lo tanto, se define el error cuadrático medio (RMS, *Root Mean Square*) como función a optimizar.

El vector de la Eq. 2.11 detalla los parámetros desconocidos en la función del modelo de error del acelerómetro y, la Eq. 2.12 determina la función objetivo.

$$\theta^{acc} = [\alpha_{yz}, \alpha_{zy}, \alpha_{zx}, s_{ax}, s_{ay}, s_{az}, b_{ax}, b_{ay}, b_{az}] \quad (2.11)$$

$$L(\theta^{acc}) = \sum_{k=1}^M (\|g\|^2 - \|h(a_k^S, \theta^{acc})\|^2)^2 \quad (2.12)$$

En la Eq. 2.12 $\|g\|$ es la gravedad local y corresponde a la referencia física para el ajuste de los parámetros, $h(a_k^S, \theta^{acc})$ es la función del modelo de error que depende de los parámetros de calibración y las lecturas de la IMU sin las medidas ajustadas, M es el número de posiciones distintas temporalmente estables y a_k^S es el vector de aceleración (medido en el marco AF) obtenido en una ventana temporal con intervalo estático. Luego, se estiman los parámetros minimizando la Eq. 2.12 con el algoritmo Levenberg-Marquardt (LM).

En el ajuste de la medida del giroscopio, los parámetros de bias se obtienen mediante el promedio de las señales leídas por el giroscopio en un intervalo estático. Para estimar los demás parámetros de calibración se define un operador g en la Eq 2.13, cuyas entradas son:

- Una secuencia de n lecturas del giroscopio ω_i medidas entre dos intervalos estáticos consecutivos $k - 1 - \text{ésimo}$ y $k - \text{ésimo}$.
- Un vector unitario $u_{a,k-1}$ dado por el acelerómetro con las medidas de aceleración ajustadas, este vector representa la dirección de la gravedad en el intervalo estático inicial.

Posteriormente, el operador retorna la estimación de un vector unitario de orientación que representa la orientación en el intervalo estático final $u_{g,k}$.

$$u_{g,k} = \mathfrak{g} [\omega_i, u_{a,k-1}] \quad (2.13)$$

En la Eq. 2.13 el operador \mathfrak{g} es un algoritmo de integración que calcula la orientación final a través de la integración de la velocidad angular y la orientación inicial. El vector de parámetros desconocidos θ^{gyro} y la función objetivo para calibrar el giroscopio $L(\theta^{gyro})$ se presentan respectivamente en las ecuaciones 2.14 y 2.15.

$$\theta^{gyro} = [\gamma_{yz}, \gamma_{zy}, \gamma_{xz}, \gamma_{zx}, \gamma_{xy}, \gamma_{yx}, s_{gx}, s_{gy}, s_{gz},] \quad (2.14)$$

$$L(\theta^{gyro}) = \sum_{k=2}^M \|u_{a,k} - u_{g,k}\|^2 \quad (2.15)$$

En la Eq. 2.15 M es el número de intervalos estáticos, $u_{a,k}$ es el vector de aceleración que se obtiene promediando las lecturas del acelerómetro en el $k - \text{ésimo}$ intervalo estático y $u_{g,k}$ es el vector de orientación estimado en la Eq. 2.13. Por último se estiman los parámetros θ^{gyro} minimizando la Eq. 2.15 con el método LM.

2.4. Representación matemática de la orientación

Como se definió en la sección 2.3 el esquema de ajuste de las medidas del giroscopio requieren rotar y determinar la orientación de la IMU en el espacio tridimensional, por tanto, es importante elegir la representación adecuada para modelar matemáticamente el concepto de rotación y orientación. Estos conceptos son físicamente familiares y a menudo se usan sin

distinción, debido a que las orientaciones suelen ser representadas como una rotación con respecto a un eje de coordenadas (conocido también como base). Hay diferentes formas para representarlas bajo un lenguaje matemático, cada uno con ventajas y desventajas propias.

Entre las formulaciones más comunes para definir la orientación de un objeto respecto a un sistema de referencia fijo, están los ángulos de Euler, las matrices de rotación y los cuaterniones. Idealmente, se busca que la representación matemática cumpla con algunas propiedades importantes:

- **Eficiencia:** Es preferible contar con una representación que ocupe el mínimo espacio en memoria para optimizar el costo computacional.
- **Robustez:** Algunas representaciones como los ángulos de Euler presentan redundancia, ya que una misma rotación puede ser reproducida de diferentes formas, esto puede causar problemas si no se manipula adecuadamente.
- **Fácil manipulación y visualización:** La elección debe contar con una fácil manipulación y una visualización geométrica amigable, esto retribuye directamente en el costo computacional.

2.4.1. Matrices de rotación

La matriz de rotación o de transformaciones espaciales (DCM, *Direction Cosine Matrix*) es una matriz 3×3 , cuyas columnas representan los vectores unitarios del objeto proyectados sobre los ejes del sistema de referencia. Desafortunadamente, las matrices de rotación tienen serios problemas computacionales. Usualmente se usan matrices homogéneas para representarlas, ya que la manipulación resulta más sencilla. Sin embargo, esto implica trabajar con información redundante. De una matriz homogénea de 4×4 , nueve entradas son utilizadas para representar una estructura con solo tres grados de libertad y sus índices $(4, i)$ e $(i, 4)$, con $i \in \{1, 2, 3\}$ son ceros. Además, se debe cuidar en todo momento que sus columnas permanezcan ortonormales y su determinante sea positivo. Si en la implementación se tienen limitaciones de memoria resulta ser una representación ineficaz [33].

La matriz de rotación o de transformaciones espaciales presentada en la Eq 2.16 relaciona un sistema de referencia fijo con el de un objeto, de forma que se puede expresar un vector del sistema de referencia asociado al objeto con el sistema fijo.

$$R_B^A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.16)$$

siendo r_{ij} el coseno del ángulo entre el eje i del sistema de referencia fijo y el eje j del sistema de referencia del objeto.

2.4.2. Ángulos de Euler

El espacio de orientaciones puede ser parametrizado por los ángulos de Euler. Para esto, la orientación del sistema de referencia de un objeto respecto a un sistema de referencia fijo se describe como una serie de rotaciones alrededor de los ejes ortogonales (x, y, z) . Estas rotaciones son usualmente llamadas X roll, Y pitch, Z yaw y se muestran en la Figura 2.4.

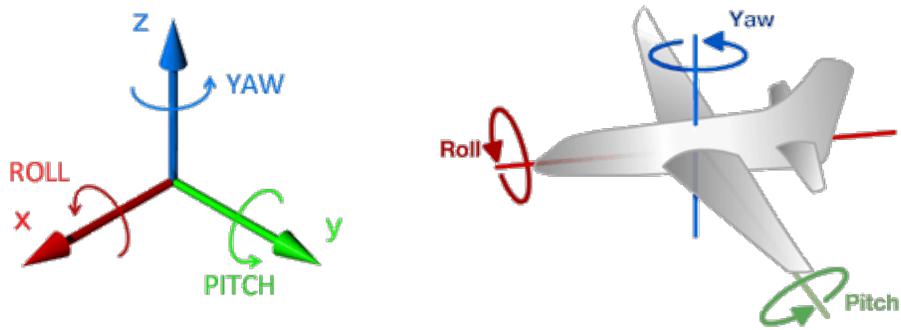


Figura 2.4: Rotaciones X roll, Y pitch, Z yaw. Diseño propio.

Cada una de estas rotaciones se puede representar matricialmente de la siguiente manera:

$$X - roll = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \quad (2.17)$$

$$Y - pitch = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad (2.18)$$

$$Z - \text{yaw} = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

Cualquier producto entre tres de estas matrices, tal que dos consecutivas no tengan el mismo eje de rotación, permite representar una orientación.

La principal desventaja de los ángulos de Euler es que matemáticamente existe una singularidad inherente que se conoce comúnmente por su nombre en inglés, *Gimbal lock* [20, 21, 22]. Esto fue una de las razones por la que la industria aeroespacial, comenzó hacer uso de otras herramientas como los cuaterniones para representar las orientaciones.

2.4.3. Cuaterniones.

La representación conocida matemáticamente como cuaternión ha venido ganando popularidad. Fue introducido por Sir William Rowan Hamilton (1805 - 1865) como una generalización para los números complejos [34, 35], sin embargo, fue Arthur Cayley (1821-1895) quien describió rotaciones a partir de la multiplicación de cuaterniones y, a partir de trabajos realizados por Shoemaker, K. [36, 37] se han destacado las virtudes de los cuaterniones sobre cualquier tipo de representación, generando interés en diversas áreas como robótica, animación por computadora, procesamiento de señales, cinemática inversa, física, etc.

Matemáticamente los cuaterniones pueden considerarse números con una parte real y tres partes imaginarias distintas:

$$q = a + bi + cj + dk \quad (2.20)$$

En la Eq.2.20 a, b, c, d son números reales y i, j, k son operadores complejos ortogonales que cumplen las siguientes condiciones:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (2.21)$$

Por tanto, $ij = -ji = k$ y $jk = -kj = i$ y $ki = -ik = j$. Nótese que la multiplicación no es conmutativa.

El conjugado y el inverso de un cuaternión en analogía a los números complejos, se dan respectivamente por:

$$\bar{q} = a - bi - cj - dk \quad (2.22)$$

$$q^{-1} = \frac{\bar{q}}{|q|^2} \quad (2.23)$$

donde,

$$|q| = \sqrt{a^2 + b^2 + c^2 + d^2} \quad (2.24)$$

Ahora, dado un vector $v \in \mathbb{R}^3$ con $\bar{v} = (u_x, u_y, u_z)$, se define en la Eq. 2.25 el cuaternión correspondiente con parte real cero.

$$\tilde{v} = 0 + u_x i + u_y j + u_z k \quad (2.25)$$

Un subconjunto del grupo de los cuaterniones es el conjunto S^3 que en la literatura corresponde a los cuaterniones unitarios, los cuales constituyen una esfera unitaria en el espacio con dimensión $n = 4$. Este conjunto juega un papel de suma importancia en relación con las rotaciones de un objeto rígido, puesto que con los cuaterniones unitarios se pueden representar rotaciones en tres dimensiones de manera muy sencilla y cualquier rotación θ alrededor de un vector $a \in \mathbb{R}^3$, con $|a| = 1$ en el espacio euclidiano \mathbb{R}^3 , puede obtenerse por medio del siguiente cuaternión unitario.

$$q = \cos\left(\frac{\theta}{2}\right) + \bar{a} \operatorname{sen}\left(\frac{\theta}{2}\right) \quad (2.26)$$

Para rotar un vector $\bar{v} = (u_x, u_y, u_z)$ mediante cuaterniones, primero obtenemos el cuaternión \tilde{v} como se definió en 2.25 y calculamos el siguiente producto.

$$\tilde{v}' = q \tilde{v} q^{-1} \quad (2.27)$$

En la Eq. 2.27 el cuaternión \tilde{v}' corresponde a la nueva orientación del vector rotado.

2.4.3.1. Propagación de cuaterniones en el tiempo.

Hasta aquí se ha explicado cómo se representa la orientación del objeto en movimiento con respecto a un marco de referencia fijo. A continuación se muestra cómo se va transformando esa representación a lo largo del tiempo, en función de las distintas rotaciones en los tres ejes del espacio, más concretamente de las tres velocidades angulares, que son en definitiva lo que medirá el giroscopio. Para el caso de los cuaterniones la ecuación diferencial cinemática que define su propagación es la Eq. 2.28, donde ω_x , ω_y y ω_z son las velocidades angulares de la rotación de los tres ejes del sensor.

$$\dot{\vec{q}} = \begin{bmatrix} \dot{a} \\ \dot{b} \\ \dot{c} \\ \dot{d} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.28)$$

o, lo que es lo mismo

$$\dot{\vec{q}} = \begin{bmatrix} \dot{a} \\ \dot{b} \\ \dot{c} \\ \dot{d} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \vec{q} \quad (2.29)$$

De acuerdo a los requerimientos de eficiencia, robustez y fácil manipulación, se considera que los cuaterniones son la representación más adecuada, porque las singularidades siempre pueden evitarse, por tener tan sólo cuatro parámetros a actualizar y por presentar menos errores en la computación.

2.5. Métodos de integración numérica .

2.5.1. Integración de un cuaternión

Ahora que se representa la orientación como un cuaternión, se plantea el problema de valor inicial para calcular el cambio de la orientación, dada la velocidad angular de un cuerpo y la orientación inicial, por medio de la integración numérica de la Eq. 2.30 que corresponde a una forma abreviada de la Eq. 2.29. (Más detalles pueden ser consultados en [38, 39, 40]).

$$q'(t) = \frac{1}{2}(\omega(t))q(t) \quad (2.30)$$

En la Eq. 2.30 $\omega(t)$ es el cuaternión correspondiente a la velocidad angular y $q(t)$ es el cuaternión de la orientación inicial. La nueva orientación del cuaternión se obtiene por medio de métodos de integración numérica para la resolución de Ecuaciones Diferenciales Ordinarias (EDO) de primer orden. Sin embargo, cuando se integra con métodos clásicos se observa que el cuaternión \bar{q} que representa la nueva orientación no tiene magnitud unitaria, siendo esto una propiedad inherente en la definición de la Eq. 2.30. Usualmente este problema se soluciona con la re-normalización del cuaternión como se muestra en la Eq. 2.31.

$$q(t+h) = \frac{\bar{q}}{|\bar{q}|} \quad \text{donde } \bar{q} = q(t) + hq'(t) \quad (2.31)$$

No obstante, las normalizaciones pueden introducir errores en la precisión y su acumulación puede afectar la estimación de la orientación [25]. Además, la necesidad de re-normalizar los cuaterniones reduce en gran medida la velocidad de integración [41]. Por lo tanto, se implementa el método de integración Crouch-Grossman que preserva la unicidad del cuaternión que se resume en la sección 2.5.3 y, los métodos clásicos como el de Tustin y Runge-Kutta resumidos en la sección 2.5.2.

2.5.2. Tustin y Runge-Kutta .

La idea general de los métodos clásicos de integración numérica es sustituir el Problema de Valor Inicial Eq. 2.32, por la ecuación integral equivalente 2.33.

$$\begin{aligned} q'(t) &= \frac{1}{2}f(\omega(t), q(t)) \\ q(t_0) &= q_0 \end{aligned} \quad (2.32)$$

$$q(t_{n+1}) = \frac{1}{2}[q(t_n) + \int_{t_n}^{t_{n+1}} f(\omega(t), q(t))dt] \quad (2.33)$$

Luego, se aproxima la integral mediante un método numérico.

- Tustin: es un método implícito y aproxima la integral de la Eq. 2.33 mediante la técnica de los trapecios. En la Eq. 2.34 se presenta la aproximación de la integral.

$$\int_{t_n}^{t_{n+1}} f(\omega(t), q(t))dt \simeq \frac{h}{2}(f(\omega(t_n), q(t_n)) + f(\omega(t_{n+1}), q(t_{n+1}))) \quad (2.34)$$

Siendo, h el tamaño de paso por iteración del método, $\omega(t_n)$ y $\omega(t_{n+1})$ las velocidades angulares dadas por el giroscopio, $q(t_n)$ el cuaternión de orientación inicial dado por el acelerómetro y $q(t_{n+1})$ al ser desconocido se estima en la Eq. 2.35 empleando el método de Euler.

$$\bar{q}(t_{n+1}) = q(t_n) + hf(\omega(t_n), q(t_n)) \quad (2.35)$$

Con la Eq. 2.36 se determina el cuaternión $q(t_{n+1})$ empleando el método de Tustin.

$$q(t_{n+1}) = \frac{1}{2}[q(t_n) + \frac{h}{2}(f(\omega(t_n), q(t_n)) + f(\omega(t_{n+1}), \bar{q}(t_{n+1})))] \quad (2.36)$$

Finalmente, como el método es implícito requiere la solución de la Eq. 2.35 cada que se obtiene el cuaternión $q(t_{n+1})$, en consecuencia es un método muy lento.

- Runge-Kutta: se describe el método de integración explícito de cuarto orden (RK4), puesto que mantiene un buen balance entre el coste del método y la precisión que se consigue con el mismo [42]. El método de RK4 determina la integral de la Eq. 2.33 mediante la regla de Simpson. En la Eq. 2.37 se presenta la aproximación de la integral.

$$\int_{t_n}^{t_{n+1}} f(\omega(t), q(t))dt \simeq \frac{h}{6}(f(\omega(t_n), q(t_n)) + 2f(\omega(t_{n+\frac{1}{2}}), \bar{q}(t_{n+\frac{1}{2}})) + 2f(\omega(t_{n+\frac{1}{2}}), \bar{q}(t_{n+\frac{1}{2}})) + f(\omega(t_{n+1}), \bar{q}(t_{n+1}))) \quad (2.37)$$

Usualmente la expresión formal del algoritmo de RK4 se describe de la siguiente forma.

$$q(t_{n+1}) = \frac{1}{2}[q(t_n) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)] \quad (2.38)$$

$$k_i = f(q^i, \omega(t_n + c_i h)); \quad q^i = q_n, \text{ para } i = 1 \quad (2.39)$$

$$q^i = q_n + h \sum_{j=1}^{i-1} a_{ij} k_j, \text{ para } i > 1 \quad (2.40)$$

Las constantes necesarias son:

$$\begin{array}{cccccc} a_{21} = \frac{1}{2} & a_{31} = 0 & a_{41} = 0 & a_{32} = \frac{1}{2} & a_{42} = 0 & a_{43} = 1 \\ c_1 = 0 & c_2 = \frac{1}{2} & c_3 = \frac{1}{2} & c_4 = 1 & & \end{array}$$

2.5.3. Crouch-Grossman

El método de integración numérica denominado Crouch-Grossman (CG) sugiere que la actualización a q_{n+1} desde $q_n \in S^3$ se realice suponiendo todos los cuaterniones en la superficie de una esfera unitaria en R^4 , es decir, la actualización se realiza bajo una restricción. En general, CG une el método clásico de integración numérica de Runge-Kutta y la teoría de grupos de Lie que permite garantizar que cualquier cuaternión obtenido se encuentre en la esfera unitaria.

El esquema implementado en el presente trabajo es un algoritmo CG de cuarto orden (CG4) con 5 etapas, dado que no existen los métodos de CG4 con cuatro etapas (más detalles pueden ser consultados en [\[43, 44\]](#))

Ahora, empleando la teoría grupos de Lie se realiza la integración de la Eq. [2.30](#) de la siguiente forma

$$q(t_{n+1}) = \exp^{(b_5 h k^5)} \exp^{(b_4 h k^4)} \exp^{(b_3 h k^3)} \exp^{(b_2 h k^2)} \exp^{(b_1 h k^1)} q(t_n) \quad (2.41)$$

siendo,

$$k^i = \frac{1}{2}(\omega(t_n + c_i h)) \quad (2.42)$$

Las constantes b_i y c_i son determinados por Jackiewicz en [43] usando minimización por el método LM.

$$\begin{aligned} b_1 &= 0,1370831520630755 & c_1 &= 0 \\ b_2 &= -0,018369853156402 & c_2 &= 0,8177227988124852 \\ b_3 &= 0,7397813985370780 & c_3 &= 0,3859740639032449 \\ b_4 &= -0,190714256550588 & c_4 &= 0,3242290522866937 \\ b_5 &= 0,3322195591068374 & c_5 &= 0,8768903263420429 \end{aligned}$$

Se resalta que el tamaño de paso del método CG4 es $c_i h$, en consecuencia, la señal de la variable física debe ser muestreada con un tiempo asíncrono. Por otra parte, el método de Runge-Kutta requiere que la señal física sea muestreada a la mitad del tamaño de paso h . Por tanto, la señal se muestrea cada $\frac{h}{2}$.

Capítulo 3

Diseño de la plataforma GIMVEBO

La GIMVEBO es una plataforma portable que facilita la adquisición de datos entregados por una IMU. El proceso de diseño, desarrollo y pruebas del prototipo funcional se describe a continuación.

3.1. Requerimientos

La GIMVEBO debe poseer las siguientes características funcionales y mecánicas.

3.1.1. Funcionales

- Proveer diferentes posiciones estáticas que permitan realizar el ajuste en los parámetros del acelerómetro y el bias del giroscopio.
- Proveer trayectorias de movimiento en el espacio tridimensional que permitan ajustar los parámetros del giroscopio.

3.1.2. Mecánicos

- Garantizar la protección de los dispositivos y del operador.
- Resistir temperaturas elevadas.

- Proteger el hardware contra la inserción de elementos sólidos.
- Garantizar la comunicación entre las tarjetas y el computador.
- Ser modular.

3.2. Diseño y fabricación de la plataforma mecánica.

El diseño mecánico de la GIMVEBO se presenta en la Figura 3.1, fue realizado en Solidworks® con base en el funcionamiento y la apariencia del Gimbal triaxial. La viabilidad del diseño se determina a través de un análisis de fuerza y movimiento, las cuales pueden consultarse en el Anexo A. A continuación se presentan detalles del diseño de la base y los brazos.

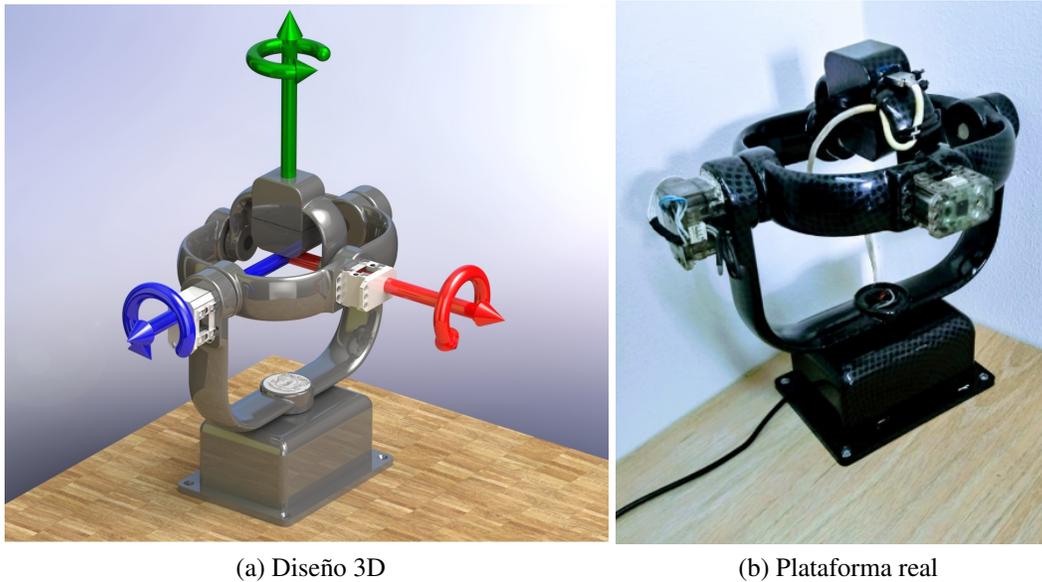


Figura 3.1: Gimvebo.

■ Base

La base consta de dos partes: una placa con ocho orificios y una caja que contiene el sistema de control de movimiento de la plataforma. Para garantizar estabilidad cuatro orificios de la placa se sujetan a una superficie firme y los restantes aseguran la caja. Una vista de las dos partes de la base se muestran en la Figura 3.2.

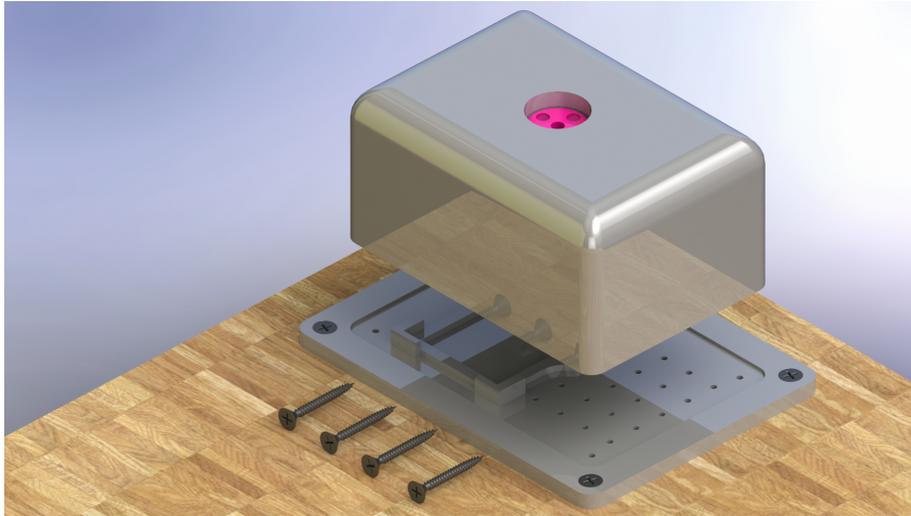


Figura 3.2: Placa y caja de la base.

■ Brazos

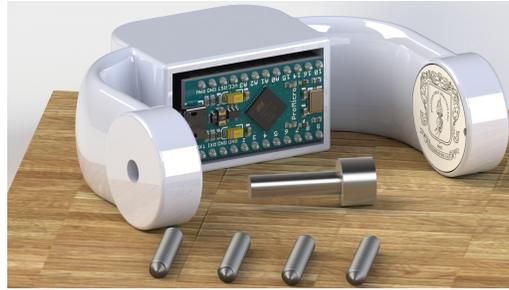
Para evitar que exista balanceo se diseñó un brazo ovalado que gira alrededor del eje «Y» y dos brazos en forma de «U» que giran independientemente alrededor de los ejes «X» y «Z». El acople entre los brazos se realiza mediante cuatro juntas. Además cada brazo posee un servomotor dynamixel XL-320 que permite aplicar con precisión rotaciones en cada eje.

Los brazos que giran entorno a los ejes «X» y «Y» poseen un vaciado por donde pasan los cables de alimentación y control de los motores. El brazo que gira entorno a «Z» posee un compartimento para almacenar el sistema de comunicación y adquisición de datos conformado por la IMU, un módulo RF y la tarjeta Arduino Pro Micro. Una vista del diseño de los dos tipos de brazo se puede apreciar en la Figura 3.3.



(a) Brazo 1

(b) Brazo 2.



(c) Brazo 3.

Figura 3.3: Brazos.

3.2.1. Fabricación

El prototipo físico se obtuvo mediante la fabricación por fundición de filamento (FFF), con la colaboración de la empresa Demos prototipado SAS de la ciudad de Popayán, el material recomendado para cumplir los requerimientos en este tipo de impresión 3D es ABS (acrilonitrilo butadieno estireno) un tipo de plástico resistente, flexible y liviano [45].

3.3. Arquitectura hardware del sistema de control y adquisición de datos.

La arquitectura hardware está formada por aquellos elementos físicos que se encargan de capturar, medir, transmitir y procesar las señales. En la Figura 3.4 se puede visualizar la solución propuesta.

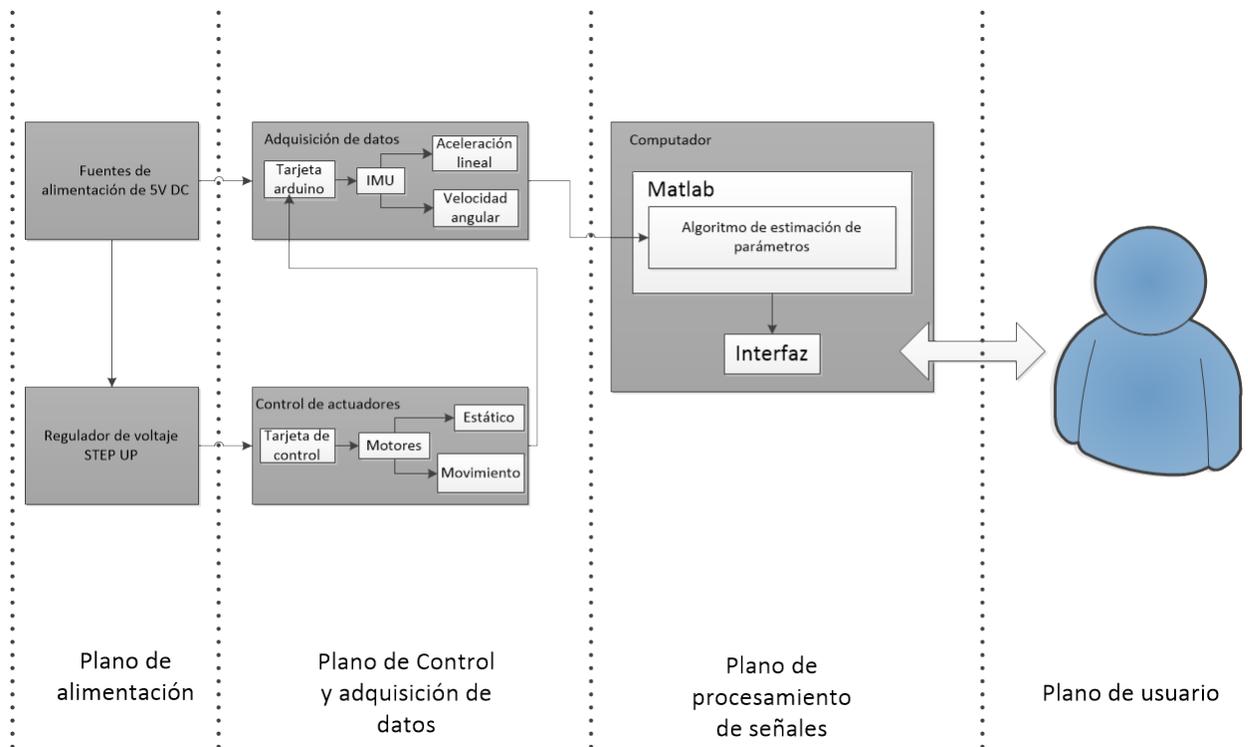


Figura 3.4: Diagrama de bloques de la solución Hardware.

Es fundamental que la interacción y comunicación entre los planos este bien definida para poder realizar un buen ajuste de las medidas de aceleración lineal y velocidad angular de una IMU. A continuación se presentan detalles y alternativas hardware para algunos planos.

Plano de usuario: Es necesario que la persona que interactuó con la plataforma lea detenidamente el manual de usuario disponible en el Anexo B, y conozca el método de ajuste de las medidas que se esta empleando, para así poder interpretar los resultados adecuadamente. La interacción con la plataforma incluye las siguientes actividades: ubicar la IMU en el compartimiento del brazo 3, ejecutar la aplicación de ajuste de medidas de la IMU y realizar la conexión de las líneas de alimentación de tensión y corriente de las fuentes de alimentación DC a las tarjetas de adquisición y control.

Plano de procesamiento de señales: Se realizó una aplicación que recibe los datos de aceleración lineal y velocidad angular y, que se encarga de realizar la estimación de los parámetros de ajuste de las medidas de una IMU. La aplicación entrega la siguiente información:

- Para el acelerómetro se muestran los parámetros y dos gráficas que indican un antes y

un después del ajuste.

- Para el giroscopio se muestra los parámetros y los diagramas de cajas y bigotes que indican la dispersión de las medidas alrededor de la referencia.

Para el funcionamiento de la aplicación se requiere un computador con sistema operativo con windows 8 ó superior.

Plano de control y adquisición de datos: La iteración entre el estado de los actuadores y la adquisición de datos debe seguir la siguiente lógica: la toma de aceleración lineal se realiza cuando los actuadores estén estáticos y la velocidad angular cuando estén en movimiento. Particularmente, para implementar cada uno de los módulos del plano de control y adquisición se evaluaron las siguientes alternativas hardware:

- Imu
 - **MPU 6050**
- Actuadores
 - **Dynamixel XL320**
 - **Dynamixel AX-12**
 - **Motor Brushless**
 - **Motor paso a paso**
- Adquisición de datos
 - **Arduino pro micro**
 - **Arduino nano**
- Control de los actuadores
 - **OpenCM**
 - **Mega 2560 mini**
 - **Arduino uno**
- Comunicación
 - **Cableado**

- **RF(Radio frecuencia)**

Plano de alimentación: Consiste en dos fuentes de alimentación de 5 V: un adaptador y el puerto serial del computador. El adaptador se conectan a la red doméstica y su salida es una señal DC regulada de 5V y 2A, que se usa para garantizar una buena alimentación de los actuadores que trabajan con mayor voltaje y corriente. El voltaje suministrado por el puerto serial del computador se usa para todos los dispositivos que requieren menos corriente para su funcionamiento. Las alternativas consideradas en este plano fueron:

- **Switching power adaptor HW-050200U33**
- **Puerto serial (Computador)**
- **Step UP - Regulador de impulso ajustable Pololu 2.5-9.5V**

3.4. Elección de la solución electrónica definitiva

Para definir la solución electrónica final se empleó la técnica de análisis morfológico desarrollada e implementada por Fritz Zwicky, un reconocido astro-físico y científico aeroespacial. La técnica explora de forma sistemática los futuros posibles, basándose en el estudio de todas las combinaciones que resultan al descomponer un sistema. En la primera fase, el espacio morfológico descompone el sistema en múltiples variables o componentes, que permiten desplegar múltiples escenarios o combinaciones conocidas como criterios de inclusión. En la segunda fase, se reduce el espacio morfológico para identificar los escenarios que son inviables (por ejemplo, por incompatibilidad), la reducción se puede realizar a través de la definición de criterios de exclusión los cuales pueden ser económicos o técnicos etc. [46].

El espacio morfológico está compuesto por todas las combinaciones que se obtuvieron al descomponer el sistema, para construir el espacio morfológico se debe seleccionar cuidadosamente los subsistemas que componen el sistema bajo inspección.

3.4.1. Construcción del espacio morfológico

En la Figura 3.5 se pueden apreciar los dispositivos que han sido contemplados y que se consideran como posibles componente de la solución electrónica definitiva.

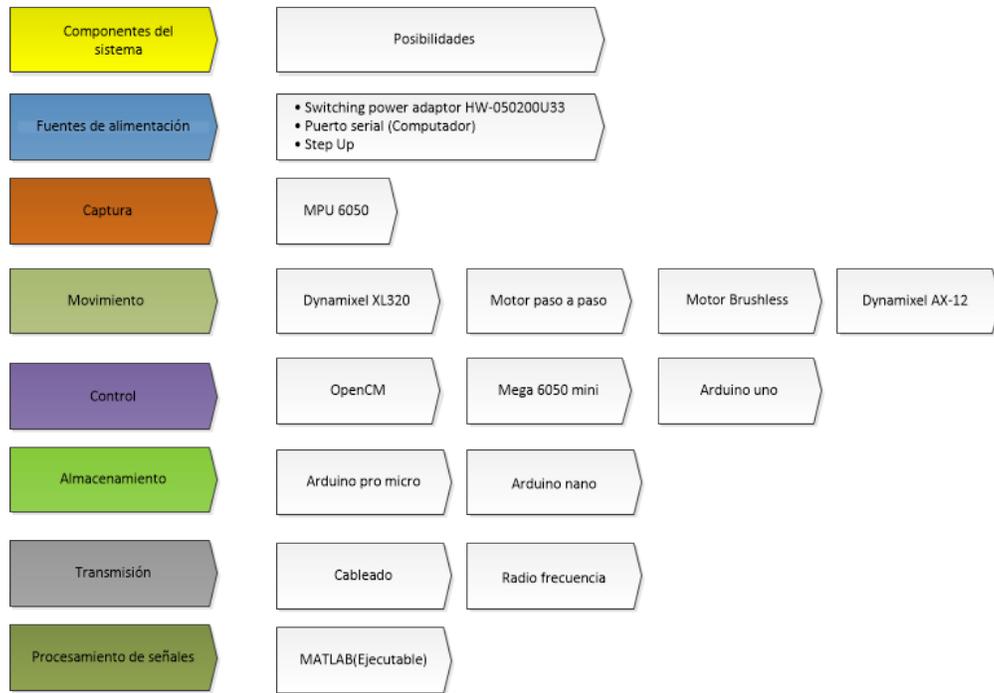


Figura 3.5: Alternativas Hardware para la solución.

De acuerdo al número de dispositivos que hacen parte de cada plano, es posible determinar de manera cuantitativa el espacio morfológico aplicando la Eq. 3.1.

$$E.M = D.Alimentación * D.Captura * D.Movimiento * D.Control * D.Almacenamiento * D.Transmisión * D.Procesamiento \quad (3.1)$$

donde,

E.M = Espacio Morfológico.

D. Alimentación = 1

D. Captura = 1

D. Movimiento = 4

D. Control = 3

D. Almacenamiento = 2

D. Transmisión = 2

D. Procesamiento = 1

$$E.M = 1 * 1 * 4 * 3 * 2 * 2 * 1 = 48$$

Estas 48 posibilidades reflejan 48 escenarios que deben ser analizados con el propósito de optar por el más conveniente.

Para determinar la elección mas viable y funcional, es necesario considerar cada una de las características de los dispositivos, pues es aquí en donde se puede realizar un análisis profundo que permitirá establecer criterios para descartar aquellos que sean inviables.

3.4.2. Reducción del espacio Morfológico

Para obtener un número manejable de posibilidades se hace una reducción del espacio morfológico descartando todas aquellas opciones que son inconsistentes o incompatibles. Así, basándose en las características eléctricas, electrónicas, mecánicas y económicas de cada uno de los dispositivos, las cuales se pueden consultar en el Anexo C, se definieron los siguientes criterios de exclusión.

Los criterios de exclusión permiten seleccionar los dispositivos que garantizan la viabilidad y la calidad de la solución electrónica, son los siguientes:

1. Tamaño: es necesario tener en cuenta el tamaño de los elementos ya que la plataforma se plantea como una solución portable.
2. Disponibilidad: facilidad de conseguir los elementos a nivel local o nacional.
3. Compatibilidad: propiedad que permite la comunicación e interacción con uno o más dispositivos.
4. Inter-conexión física: reducir al máximo las conexiones entre los elementos ya que el cableado sera interno.

5. Sobre-dimensionamiento: dispositivos con amplias funcionalidades que puedan resultar exageradas para el objetivo del proyecto.
6. Niveles de tensión: la alimentación de los dispositivos debe poder proveerse en cualquier escenario con condiciones mínimas.
7. Alto costo: el costo final debe ser asequible a un amplio público.
8. Requerimiento de dispositivos adicionales: deben requerir la mínima cantidad de elementos adicionales para su implementación.
9. Amplio rango de operación: para generar todas las trayectorias posibles el rango de operación y control de los motores debería ser idealmente de 360°.
10. Precisión: ángulo mínimo de control del motor.
11. Herramientas de desarrollo: disponibilidad y compatibilidad de herramientas que pueden ser utilizadas según la preferencia del desarrollador.
12. Soporte y material de apoyo: cantidad de información disponible que permita conocer, configurar e implementar un dispositivo.
13. Escalabilidad: característica de un dispositivo que permite incorporar nuevas funcionalidades en el transcurso del tiempo.
14. Consumo: los dispositivos deben ser de bajo consumo de corriente.

Con los criterios previamente definidos se realiza un análisis para descartar uno o varios dispositivos de cada grupo.

- Fuentes de alimentación DC

Basados en los criterios de exclusión y requerimientos de la plataforma, en este grupo no se descartan dispositivos porque en conjunto los tres que se consideraron brindan una solución robusta, que beneficia en gran parte la portabilidad de la GIMVEBO.

- Adquisición de datos

El dispositivo MPU6050 es el sensor que cumple con la mayoría de los criterios de exclusión, además es ampliamente utilizado debido a su fácil configuración, instalación, tensión de alimentación baja, buena precisión de lectura de datos, bajo consumo de corriente y potencia, flexibilidad de comunicación, amplia información y bajo costo.

- **Movimiento (Actuadores).**

El motor paso a paso y el motor brushless se descartan porque son los que poseen mayor tamaño y peso entre los dispositivos considerados, además no son compatibles con algunas tarjetas de control, requieren numerosas conexiones físicas y necesitan controlar su posición a través de un driver, usualmente costoso. Adicionalmente, el rango de torque ofrecido es significativamente superior al que se requiere para el proyecto. Resumiendo, los motores paso a paso no satisfacen los criterios 1,3,4,5 y 8. En consecuencia, la opción más viable son los servomotores de la familia dynamixel.

Para definir el servomotor que mejor se ajusta al proyecto se analiza la capacidad de satisfacer completamente los criterios de exclusión 1,6 y 14. Después del análisis se concluye que el servomotor XL-320 es el que ofrece mejor desempeño.

- **Tarjetas de control.**

Las tarjetas Mega2560mini y Arduino uno se descartan porque no cumplen con los criterios 1,3,4,8 y 11. Sin embargo, cabe aclarar que la principal limitante de estas tarjetas es que necesitan de hardware externo para comunicarse con los servomotores.

- **Almacenamiento.**

La arduino nano no satisface los criterios de exclusión 1,7 y 14. Por lo tanto, para el almacenamiento de datos generados por la IMU la Arduino Pro Micro es la opción mas viable.

- **Transmisión de datos.**

Basados en los criterios de exclusión y requerimientos de la plataforma, en este grupo se emplea la comunicación por RF para conocer el estado de adquisición de datos(Reposo inicial-Movimiento-Reposo final) y la comunicación cableada para enviar los datos.

3.4.3. Componentes hardware

Basados en los requisitos expuestos en la sección 3.4.2, se seleccionaron los siguientes componentes para construir el hardware de control y adquisición de datos.

Actuadores

- **Dynamixel XL320**



Figura 3.6: Dynamixel XL320

Ampliamente utilizados en robótica, disponen de bucles de control ajustados, poseen capacidad de consulta de variables internas como posición, temperatura, voltaje aplicado, etc. Además, pueden comunicarse empelando lógica TTL. Las especificaciones técnicas de estos actuadores se pueden consultar en el Anexo C.

Tarjetas de control

- **Controladora ROBOTIS OpenCM 9.04**



Figura 3.7: OpenCM 9.04

Esta tarjeta controladora está especialmente diseñada para la interacción bidireccional con los dynamixel XL-320, alimenta los actuadores a través de conectores compatibles y dispone de librerías para controlarlos en un entorno basado en la programación de Arduino. Las especificaciones técnicas pueden ser consultadas en el Anexo C.

- **Arduino Pro Micro**



Figura 3.8: Arduino pro micro

Esta tarjeta permite la adquisición de datos de la IMU mediante la comunicación I2C, su tamaño permite hacer de la GIMVEBO una plataforma portable, las especificaciones técnicas de esta tarjeta se pueden consultar en el Anexo C.

Elementos de potencia

- **Regulador de voltaje Step up**



Figura 3.9: Step up

El Step up fabricado por la empresa Pololu posee un amplio rango de salida que garantiza la tensión requerida para el funcionamiento de la tarjeta OpenCm 9.04 y de los actuadores. Las especificaciones técnicas de estos actuadores se pueden consultar en el Anexo C.

- **Adaptador HW-050200U3W**



Figura 3.10: Adaptador

Este adaptador proporcionan 5 V - 2 A. Las especificaciones técnicas del adaptador se puede consultar en el Anexo C.

Comunicación

- **Módulo RF Serial 433 MHZ Bidireccional, HC-11**

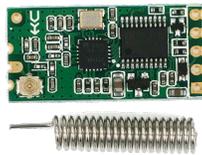


Figura 3.11: Modulo RF HC-11, Bidireccional

Este módulo cuenta con múltiples modos de transmisión y su proceso de configuración es sencillo, sus características técnicas pueden consultarse en el Anexo C.

- **Cable Usb.**



Figura 3.12: Cable Usb.

Con puerto Usb serial y micro-b.

3.5. Inter-conexión de componentes

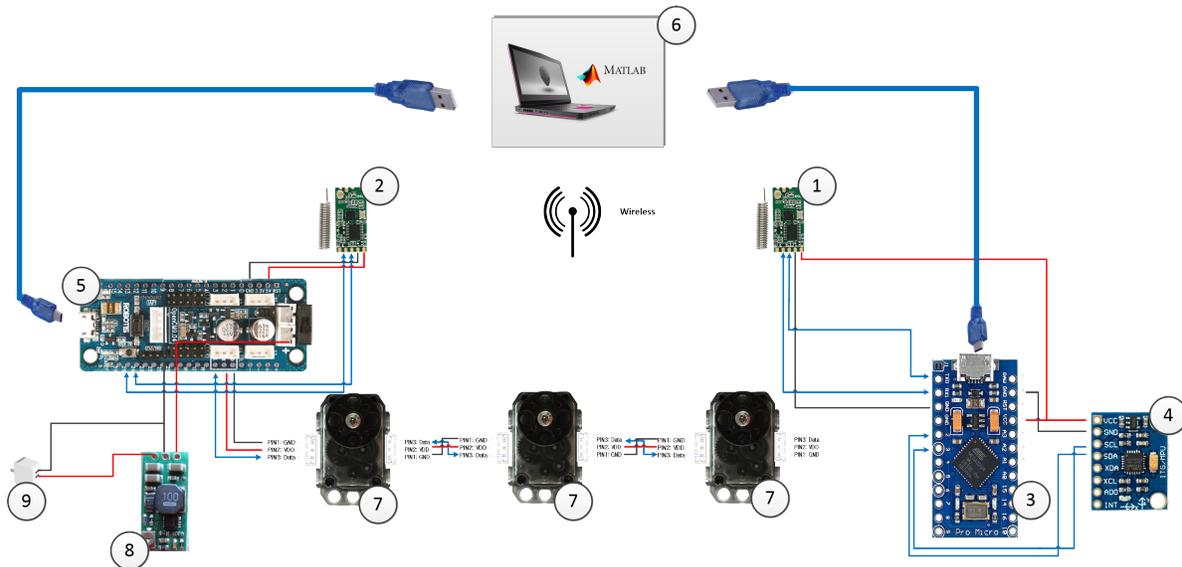


Figura 3.13: Inter-conexión de componentes.

En la Figura 3.13 las conexiones rojas y negras corresponden respectivamente a la alimentación y la tensión de referencia. A través de los dispositivos RF(1-2) la Arduino Pro Micro (3) recibe de la OpenCM 9.04 (5) la información del tipo de variable física que debe leer la IMU (4) y, luego la envía al puerto serial del computador (6). En el control de los motores la OpenCM 9.04 (5) recibe del computador (6) el estado del procedimiento del ajuste en las medidas y envía a cada motor (7) por medio de comunicación TTL la acción a realizar. Respecto a la alimentación, el Adaptador (9) energiza con 5 V al Step up (8) que produce un voltaje de 7.4 V para alimentar la OpenCM 9.04 (5) que a su vez energiza los motores (7) con 7.4 V y al RF(2) con 5 V. Además, el RF(1) es energizado con 5 V provenientes de la Arduino Pro Micro (3).

La elección de la solución electrónica definitiva utilizada en el desarrollo de la plataforma GIMVEBO cumple satisfactoriamente los requerimientos básicos para su funcionamiento. Sin embargo, como los dispositivos son económicos estos han impuesto limitantes en la capacidad de almacenamiento de datos y en el procesamiento. Una de las principales limitantes se presentó en la tarjeta de adquisición de datos Arduino Pro Micro, la cual solo logra garantizar un tiempo de muestreo mínimo de 20 ms y almacenar 200 muestras.

3.6. Solución software.

Se implementó una solución software basada en los requerimientos presentados en la Tabla 3.1, los cuales fueron definidos a partir del objetivo del proyecto.

Especificaciones	
Problema	<ul style="list-style-type: none"> ■ Ajuste de los parámetros de calibración.
Dominio del problema	<ul style="list-style-type: none"> ■ Interacción del usuario con el proceso. ■ Funcionamiento e interacción de los elementos hardware.
Características de uso	<ul style="list-style-type: none"> ■ El usuario empleará la aplicación para configurar y generar la prueba de ajuste de parámetros. ■ El usuario podrá observar la adquisición de datos y los resultados del experimento ejecutado.
Características funcionales	<ul style="list-style-type: none"> ■ Configurar el número de datos obtenidos por el acelerómetro. ■ Visualización gráfica de la adquisición de datos. ■ Configurar el número de posiciones y trayectorias en el experimento. ■ Almacenamiento de los datos obtenidos por la IMU. ■ Visualización de los resultados del experimento.
Público objetivo	<ul style="list-style-type: none"> ■ Todas las personas que estén interesadas en el proceso de estimación de parámetros de una IMU y que hallan leído el manual de usuario incluido en el anexo C.
Características técnicas	<ul style="list-style-type: none"> ■ Dispositivos con sistema operativo Windows 8 o superior. ■ Dispositivo con puerto USB disponible. ■ Aplicaciones instalas <i>ARDUINO IDE</i> y <i>ROBOTICS_openCM IDE</i>.

Tabla 3.1: Requerimientos de la solución.

La versión final de la solución software implementada esta compuesta por cuatro subsistemas, conforme se ilustra en la Figura 3.14.

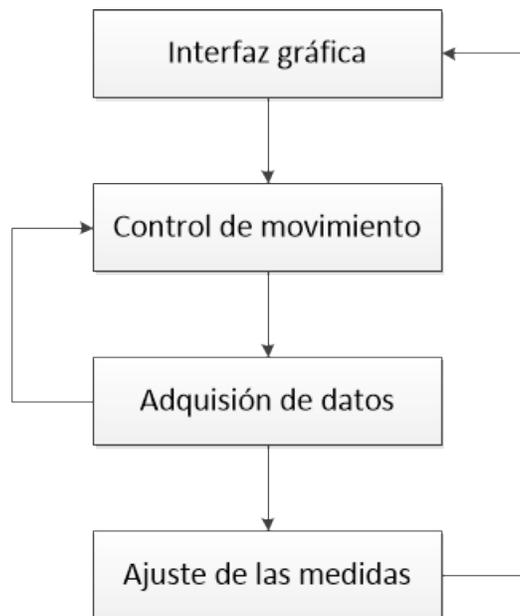


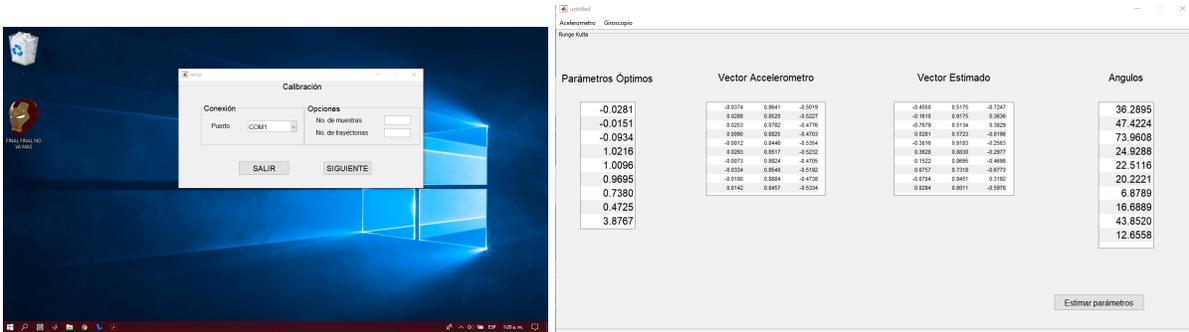
Figura 3.14: Modelo de la solución software.

Una breve descripción de cada subsistema se presenta a continuación.

3.6.1. Interfaz gráfica de usuario.

La interfaz gráfica de usuario se desarrolló en el lenguaje de programación M¹ y permite configurar parámetros del sistema y visualizar los resultados para cada algoritmo de integración numérica a comparar. Para el desarrollo de la interfaz gráfica se aprovechó el potencial del toolbox de MatLab[®] para GUI. En la Figura 3.15 se presenta la ventana principal y los resultados para el algoritmo de RK4.

¹Lenguaje de programación propio de la herramienta MatLab[®]



(a) Ventana principal

(b) Resultados RK4

Figura 3.15: Interfaz gráfica de usuario.

3.6.2. Control

Este subsistema cuenta con un código desarrollado en la aplicación *ROBOTICS_openCM IDE*, el bucle principal ejecuta, de forma individual las siguientes acciones:

1. Fijar la posición de los tres actuadores para situar la IMU en la posición estática inicial, y luego enviar la orden para adquirir los datos para aceleración lineal.
2. Verificar que la adquisición de datos haya concluido satisfactoriamente.
3. Fijar una velocidad constante a los motores y enviar la orden para adquirir los datos de velocidad angular.
4. Enviar la orden para adquirir los datos de aceleración lineal para la posición estática final.

A continuación en la Figura 3.16 se presenta el diagrama de flujo del subsistema.

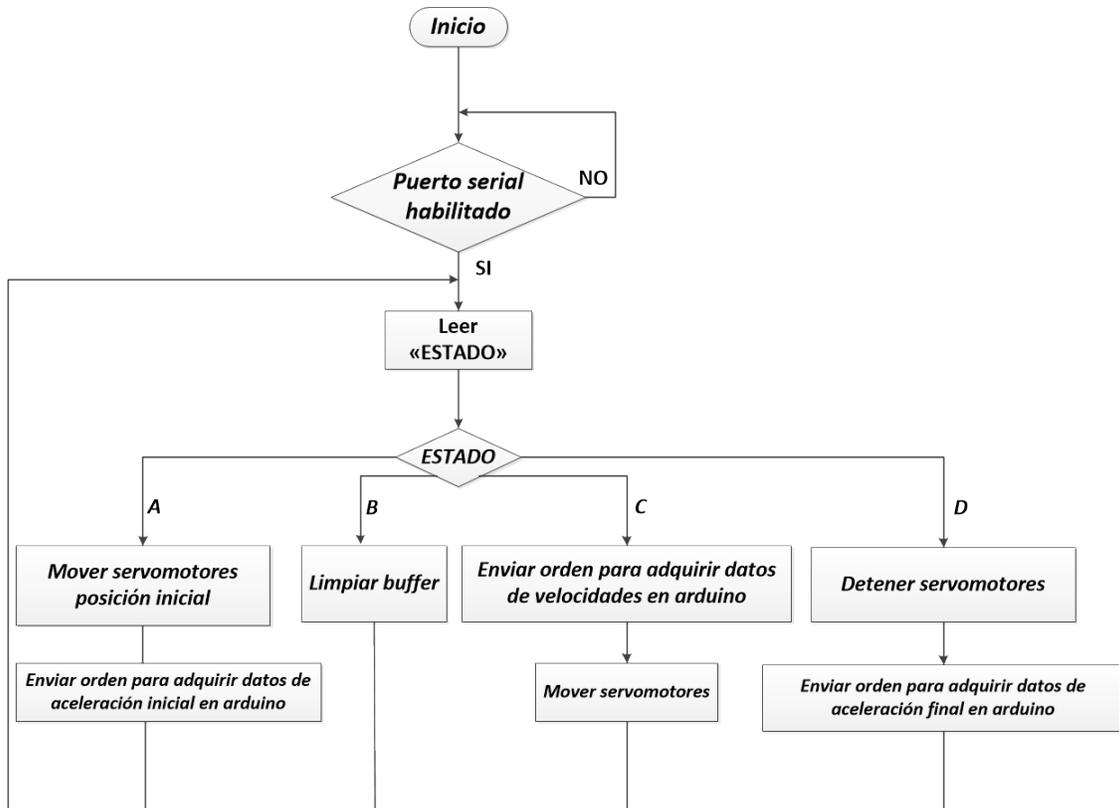


Figura 3.16: Diagrama de flujo algoritmo de control.

3.6.3. Adquisición de datos.

La adquisición de datos se dividió en las siguientes etapas:

- Medir la aceleración lineal y velocidad angular.
- Almacenar las medidas tomadas.

El software se desarrolló en el aplicativo *ARDUINO IDE*, se utilizaron las librerías *I2Cdev.h* y *MPU6050.h*, las cuales configuran y establecen la comunicación utilizando el protocolo de comunicación I2C.

Para el caso del giroscopio se debe garantizar un tiempo de muestreo constante, por tanto

estos datos deben ser almacenados en la tarjeta Arduino Pro-Micro para después ser enviados a la aplicación. A continuación en la Figura 3.17 se presenta el diagrama de flujo del subsistema.

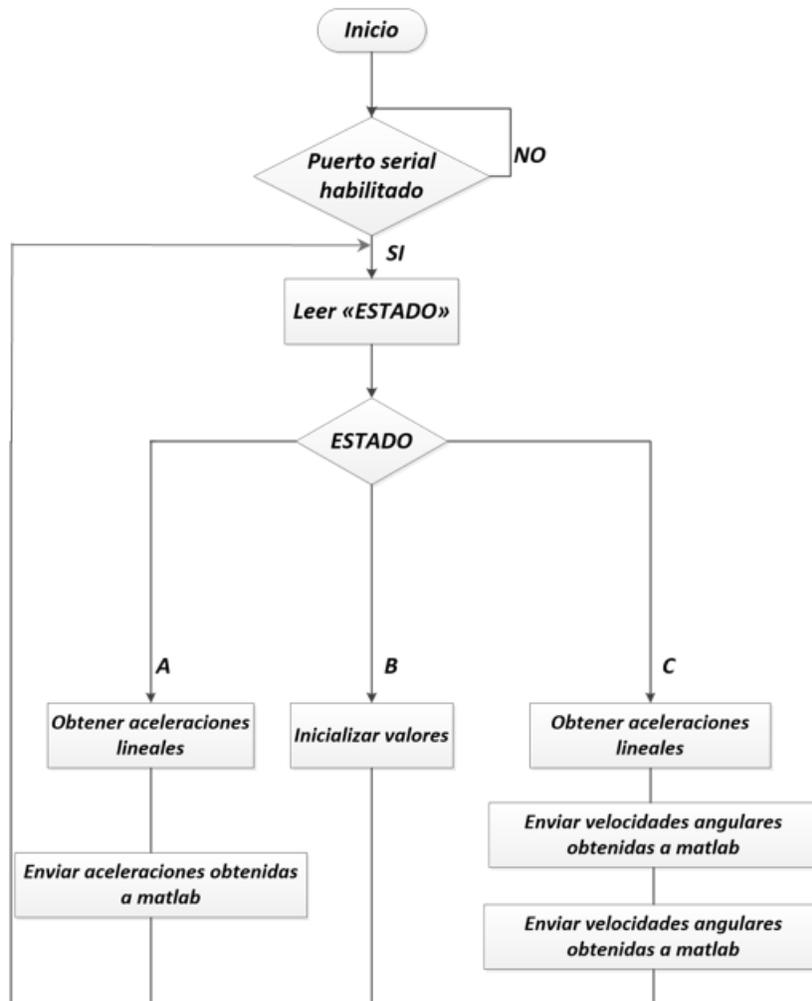


Figura 3.17: Diagrama de flujo algoritmo adquisición de datos.

3.6.4. Ajuste de las medidas.

El ajuste de las medidas consiste en recibir los datos de la aceleración lineal (A_x, A_y, A_z) - velocidad angular (G_x, G_y, G_z), estimar los parámetros del modelo de error y presentar los resultados en la interfaz gráfica de usuario. En la implementación del algoritmo para ajustar las medidas de aceleración y velocidad angular se usó el entorno de programación de Matlab[®], principalmente por su capacidad para analizar y simular los sistemas de diversa

complejidad. En particular, se emplea la rutina de optimización *lsqcurvefit* que esta diseñada especialmente para problemas en donde se desea optimizar empleando el criterio de mínimos cuadrados (LM).

3.6.4.1. Acelerómetro

Se obtiene la aceleración lineal entregada por la IMU en el intervalo estático inicial y final, luego se optimiza la función objetivo definida en la sección 2.3.3 y se determinan los parámetros óptimos. Después se ajustan las medidas del acelerómetro con los parámetros óptimos obtenidos. El diagrama de flujo del código se presenta en la Figura 3.18.

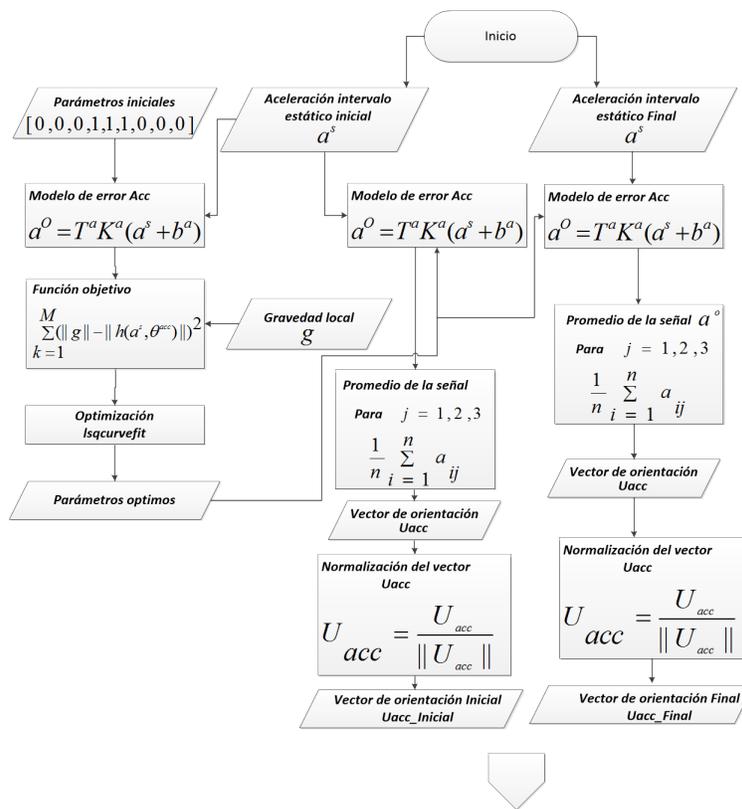


Figura 3.18: Diagrama de flujo del código del acelerómetro,

3.6.4.2. Giroscopio

Se obtienen las velocidades angulares, los vectores de orientación Inicial-Final dados por el acelerómetro ajustado y la matriz de rotación de cosenos directores (*dcm*) que relaciona

un sistema de referencia fijo con el de la IMU. Posteriormente, se obtiene el cuaternión de rotación correspondiente a la matriz (*dcm*) mediante el comando *dcm2quat*.

Dado que primero se debe ajustar el acelerómetro el diagrama de flujo para el giroscopio se presenta en la Figura 3.19 como continuación de la Figura 3.18.

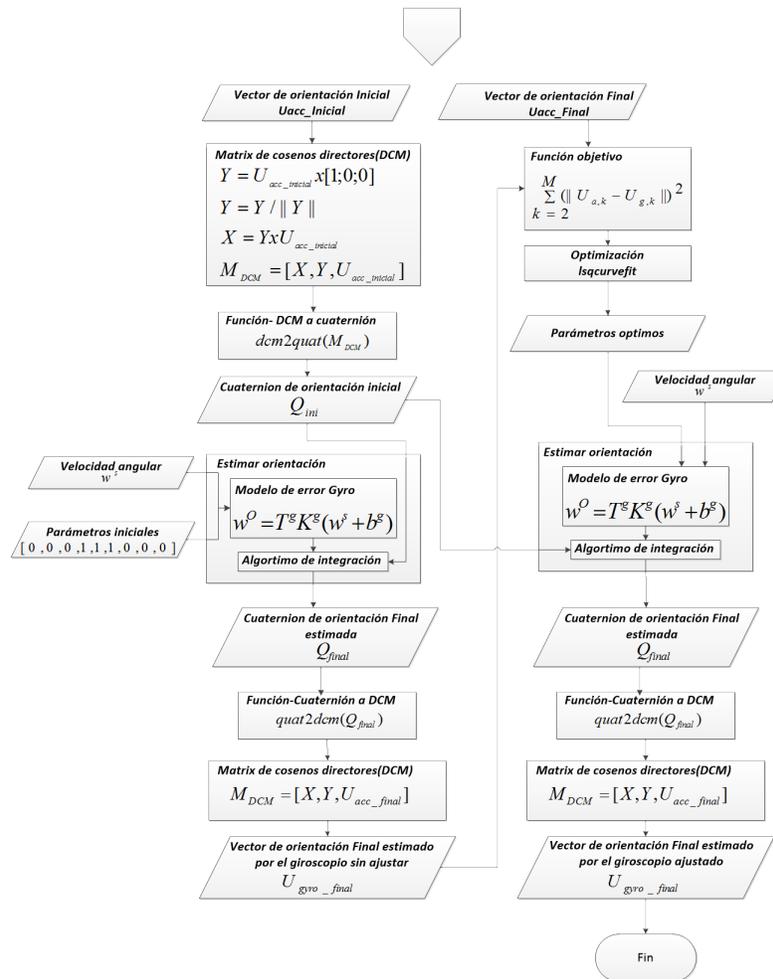


Figura 3.19: Diagrama de flujo del código del giroscopio.

Capítulo 4

Resultados

Los resultados se presentan empleando los diagramas de Cajas y Bigotes (*boxplot*), los cuales son una representación visual que describe la dispersión de los datos alrededor de un valor de referencia.

4.1. Diagramas de cajas y bigotes

Un diagrama de caja y bigotes es un gráfico basado en cuartiles¹ y está compuesto por un rectángulo (la «caja») y dos brazos (los «bigotes»).

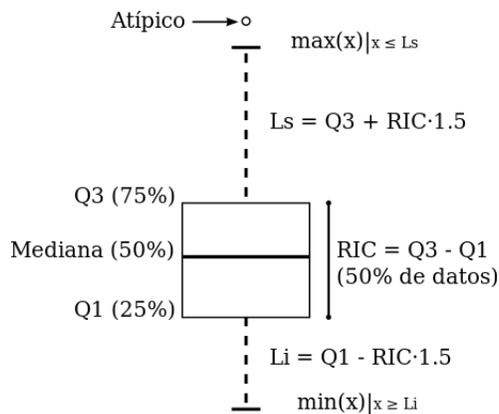


Figura 4.1: Diagrama de cajas y bigotes. Diseño propio

¹valores que dividen un conjunto de datos ordenados en cuatro partes porcentualmente iguales.

A diferencia de la media y la desviación estándar la mediana y el rango intercuartil no son afectados por observaciones extremas, en consecuencia constituyen una mejor medida de la tendencia central y la dispersión de los datos.

- Mediana (Me) o segundo cuartil (Q2).

La mediana estadística es el número central de un grupo de números ordenados por tamaño. Además, es una medida de tendencia central, por tanto, su valor pretende resumir o representar un conjunto de datos.

- Cuartiles.

Los cuartiles $Q1$ y $Q3$ corresponden respectivamente al 25% y 75% de los datos. El primer cuartil $Q1$ indica que el 25% de los datos es menor que o igual a este valor, un análisis semejante se realiza a los otros cuartiles.

- Rango intercuartil.

El rango intercuartil (RIC) es la distancia entre el primer cuartil $Q1$ y el tercer cuartil $Q3$ por lo que el 50% de los datos está dentro de este rango. El valor del RIC describe la dispersión de los datos, por tanto, a medida que aumenta la dispersión el valor de RIC es más grande.

- Límite máximo y mínimo ($Vmin/Vmax$).

Los bigotes tienen un límite de prolongación máximo y mínimo que se determinan con las ecuaciones 4.1, cualquier dato o caso que no se encuentre dentro de este rango es marcado e identificado individualmente como un valor atípico (VAt), es decir, datos que están muy distantes de los otros valores.

$$LimiteInferior(Li) = Q1 - RIC(1,5), LimiteSuperior(Ls) = Q3 + RIC(1,5) \quad (4.1)$$

4.2. Obtención de datos

Basándose en el esquema de obtención de datos definido en la sección 2.3 se obtuvieron datos para 10 experimentos aleatorios, cada experimento consta de un grupo de datos conformado por:

- 100 muestras de aceleración lineal en el intervalo estático inicial.
- 100 muestras de aceleración lineal en el intervalo estático final.
- 200 muestras de velocidad angular obtenidas con tiempo de muestreo igual a 20ms.

A continuación, se presentan los datos obtenidos para un experimento.

En la Figura 4.2, se presenta una forma de onda típica de la magnitud de aceleración lineal medida en los tres ejes para los intervalos estáticos inicial y final.

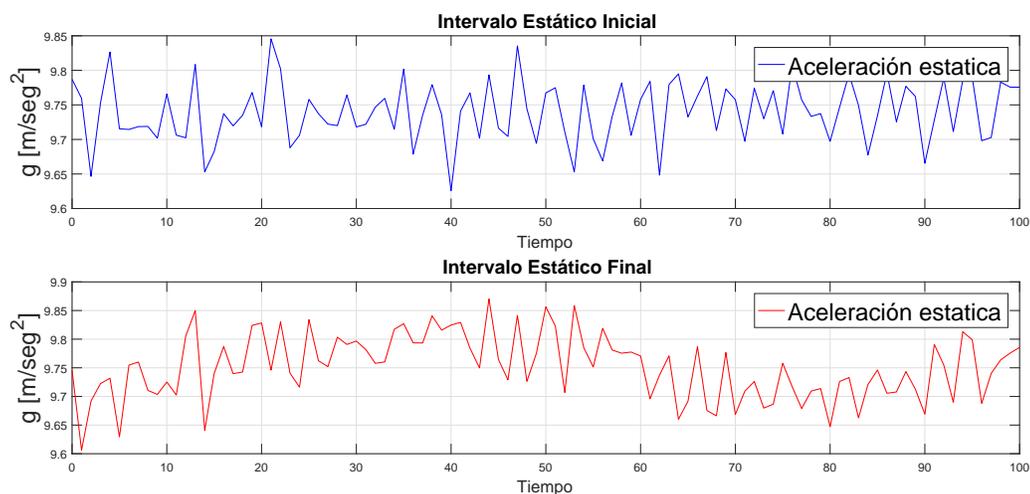


Figura 4.2: Aceleración estática.

Por otra parte, en la Figura 4.3, se presenta una forma de onda típica de la velocidad angular durante el movimiento de la IMU.

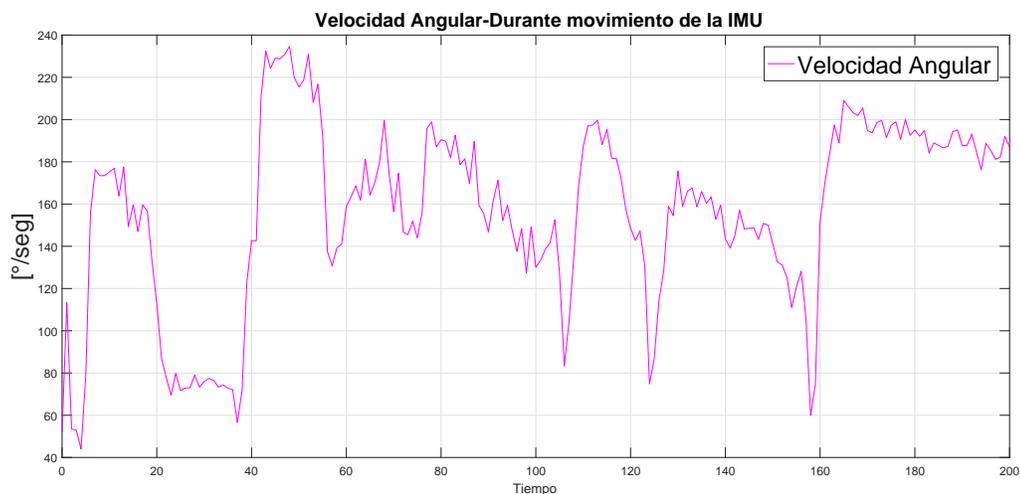


Figura 4.3: Velocidad Angular.

4.3. Resultados para el acelerómetro

Debido que el acelerómetro constituye la referencia matemática para ajustar el giroscopio, se expone en primera instancia los resultados del ajuste en las medidas del acelerómetro. En el procedimiento se utiliza el valor de la gravedad local $g = 9,7992 \text{ m/s}^2$ [47] como la referencia física para el ajuste.

En la Figura 4.4 se presenta la evolución en el tiempo de 500 muestras de aceleración lineal obtenidas por el sensor acelerómetro sin ajustar y ajustadas, luego en la Figura 4.5 se muestra el diagrama de cajas y bigotes para representar la dispersión de las mismas muestras.

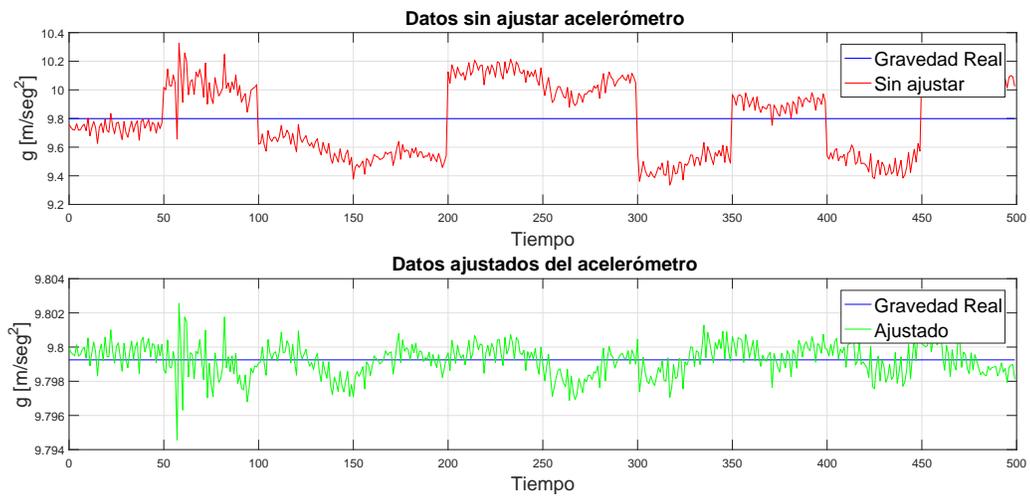


Figura 4.4: Aceleración lineal en el tiempo.

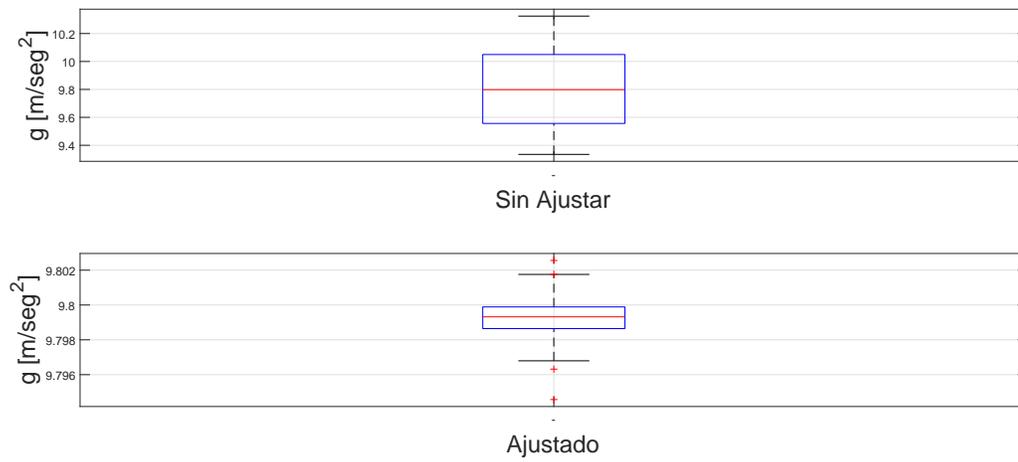


Figura 4.5: Medidas sin ajustar y ajustadas.

En la Tabla 4.1 se sintetizan los valores obtenidos de los diagramas de cajas y bigotes.

Referencia física de ajuste $g = 9.7992 \text{ m/s}^2$		
Información del Diagrama de Cajas y Bigotes	Sin ajustar	Ajustado
Primer Cuartil (Q1)	9.5565	9.7986
Mediana (Q2)	9.7984	9.7993
Tercer Cuartil (Q3)	10.0497	9.7999
Rango intercuartil (RIC)	0.4932	0.0013
Valor mínimo	9.3349	9.7968
Valor máximo	10.3247	9.8017
Valor atípico	— — —	9.8026
Valor atípico	— — —	9.8018
Valor atípico	— — —	9.7963
Valor atípico	— — —	9.7946

Tabla 4.1: Resultados Acelerómetro.

Los datos presentados en la Tabla 4.1 permiten concluir que después del ajuste de la aceleración lineal se redujo:

1. Respecto al valor de referencia, el 25 % de las magnitudes de aceleración lineal comprendidas entre:
 - a) Valor mínimo y $Q1$
 - b) $Q3$ y el valor máximo
2. El RIC, es decir, la dispersión del 50 % de las magnitudes alrededor del valor de referencia es menor.
3. La diferencia entre el valor de la mediana respecto al valor de referencia.

Adicionalmente, se emplea el error cuadrático medio (ECM Eq. 4.2) para cuantificar el error entre la referencia física respecto a la medición antes y después del ajuste de los parámetros.

$$ECM = \frac{1}{N} \sum_{k=1}^N (g - a_k^o)^2 \quad (4.2)$$

En la Eq. 4.2, g es la gravedad local y corresponde a la referencia física, a_k^o es la magnitud

de la aceleración lineal y $N = 500$ corresponde a la cantidad de muestras. Los resultados obtenidos del error cuadrático medio con las medidas del acelerómetro:

- Sin ajustar es de 0,0667.
- Ajustadas es de $8,9378 \times 10^{-7}$.

Con los resultados presentados se puede apreciar que el ajuste de las medidas redujo significativamente el error respecto a la referencia física utilizada. Por tanto, se concluye que se realizó un ajuste correcto en las medidas del acelerómetro, ahora este sensor puede constituir una referencia confiable para ajustar el giroscopio.

4.4. Resultados para el giroscopio

En esta sección se determina la capacidad de los tres algoritmos de integración para preservar la unicidad de un cuaternión y su desempeño en el ajuste de los parámetros de calibración del giroscopio de la IMU.

4.4.1. Unicidad del cuaternión

En la Figura 4.6 mediante un diagrama de cajas y bigotes se muestra la distribución de 200 magnitudes de cuaterniones obtenidas después de cada paso de integración. Adicionalmente, en la Figura 4.7 se muestra la evolución de la magnitud del cuaternión en el tiempo.

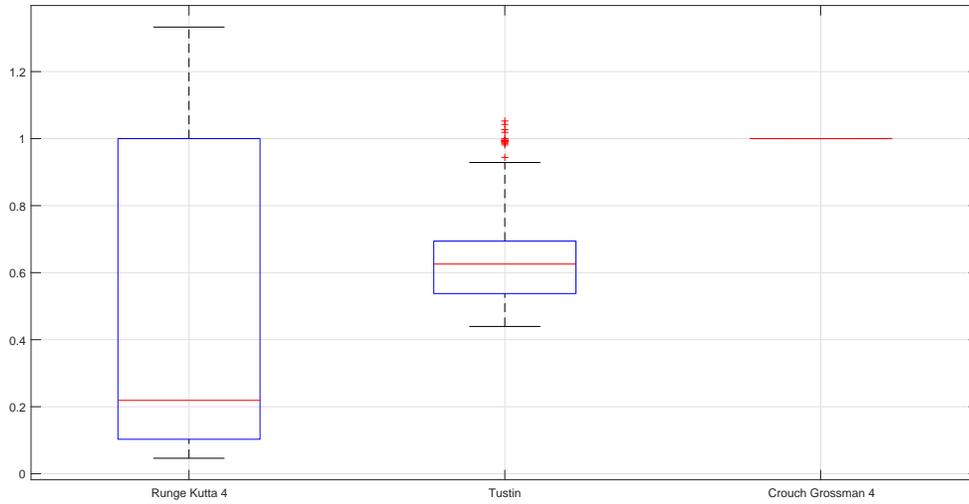


Figura 4.6: Magnitudes RK4,Tustin,CG4.

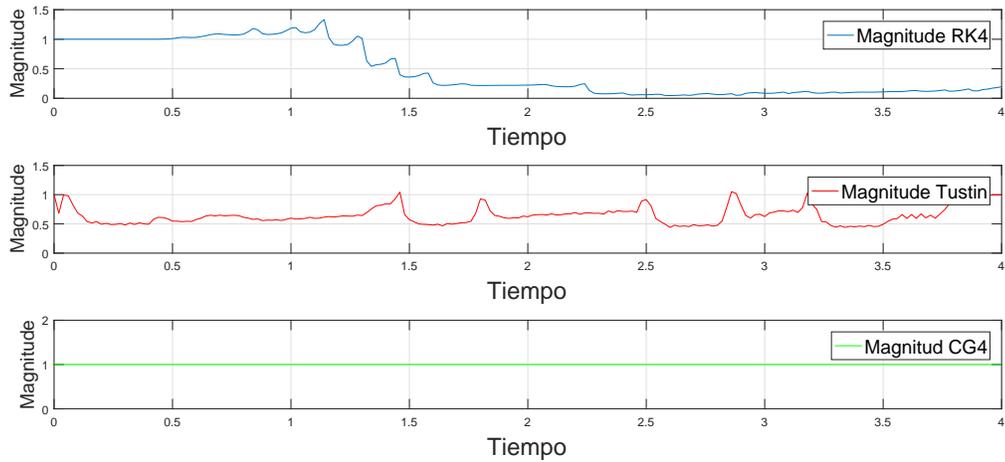


Figura 4.7: Magnitud vs Tiempo (RK4-CG4-Tustin)

En la Tabla 4.2, se resumen los valores obtenidos del diagrama de cajas y bigotes.

Información del Diagrama de Cajas y Bigotes	Runge Kutta	Tustin	Crouch Grossman
Primer Cuartil (Q1)	0.1029	0.5376	1
Mediana (Q2)	0.2190	0.6256	1
Tercer Cuartil (Q3)	1	0.6942	1
Rango intercuartil (RIC)	0.8971	0.1566	0
Valor Mínimo	0.0462	0.4395	1
Valor Máximo	1.3329	0.9287	1
Valores atípicos	— — —	18	— — —

Tabla 4.2: Magnitudes RK4,Tustin,CG4.

En la Tabla 4.2, el valor de la mediana y el RIC evidencian que la unicidad del cuaternión solo se preserva con el algoritmo de CG4. Adicionalmente, en la Figura 4.7 se puede apreciar que el algoritmo de Tustin respecto a Runge-Kutta obtiene magnitudes mas cercas a uno y menos dispersas, en consecuencia, Tustin preserva mejor la magnitud del cuaternión. Sin embargo, como se reflejo en la Tabla 4.2 Tustin obtiene 18 valores atípicos, por lo cual es un método muy impreciso.

4.4.2. Desempeño de los algoritmos de integración

Con las velocidades angulares dadas por el giroscopio y una orientación inicial dada por el acelerómetro se estima a través de integración numérica la orientación en el intervalo estático final, donde también se determina el desempeño de cada algoritmo mediante el ángulo α formado por el vector de orientación final dado por el acelerómetro y el estimado mediante la integración numérica, el cual idealmente debería ser cero.

Adicionalmente, los ángulos α obtenidos por los algoritmos RK4,CG4 y Tustin se evalúan para las siguientes configuraciones:

1. Medidas de aceleración y giro ajustadas.
2. Medidas de aceleración ajustadas y giroscopio sin ajustar.
3. Medidas de aceleración y giro sin ajustar.

4.4.2.1. Resultados para RK4, CG4 y Tustin

- Runge-Kutta

Los resultados de los ángulos α obtenidos en los tres escenarios mencionados se presentan en la Figura 4.8.

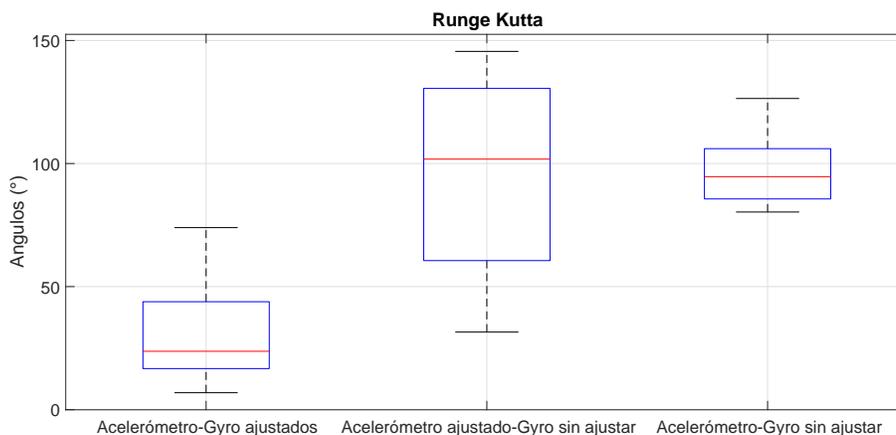


Figura 4.8: Ángulos α obtenido con las medidas sin ajustar y ajustadas RK4.

En la Tabla 4.3 se resumen los valores obtenidos del diagrama de cajas y bigotes.

Acelerometro	Giroscopio	Información del Diagrama de Cajas y Bigotes						
Ajustado	Sin Ajustar	Q1	Q2	Q3	RIC	VMin	VMax	VAt
✓	✓	16.68	23.72	43.85	27.17	6.87	73.96	— — —
✓	✗	60.51	101.79	130.52	69.97	31.51	145.56	— — —
✗	✗	85.61	94.66	106.04	20.4300	80.30	126.43	— — —

Tabla 4.3: Resumen del diagrama de cajas y bigotes para RK4.

- Crouch-Grossman (CG4).

Este algoritmo requiere que la señal física sea muestreada con un tiempo asíncrono, lo que demanda precisión en el hardware de adquisición. En consecuencia, se implementa un interpolador lineal presentado en la Eq. 4.3 para determinar la velocidad angular en los puntos intermedios requeridos por el algoritmo.

$$\omega_{c_i} = (1 - c_i)(\omega(t_n) + c_i(\omega(t_{n+1}))) \quad (4.3)$$

Siendo, c_i las constantes definidas en la sección 2.5.3.

Los resultados obtenidos para cada una de las tres configuraciones evaluadas se presentan en la Figura 4.9.

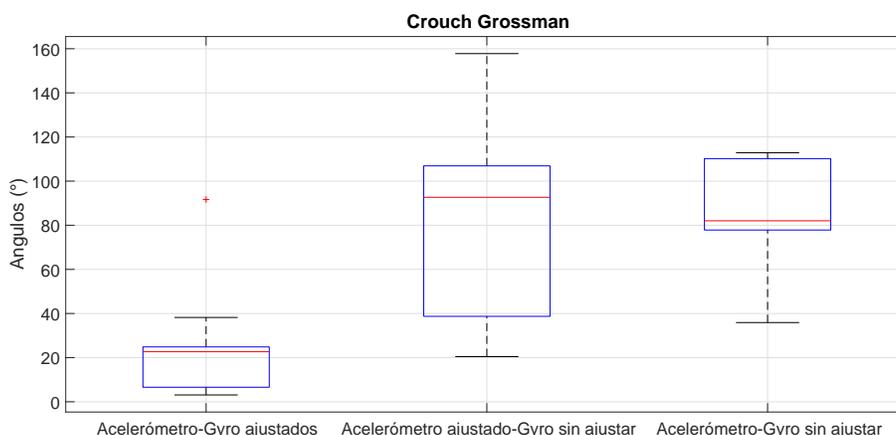


Figura 4.9: Ángulos α obtenido con las medidas sin ajustar y ajustadas CG4.

En la Tabla 4.4 se resumen los valores obtenidos del diagrama de cajas y bigotes.

Acelerometro	Giroscopio	Información del Diagrama de Cajas y Bigotes						
Ajustado	Sin ajustar	Q1	Q2	Q3	RIC	VMin	VMax	VAt
✓	✓	6.59	22.67	24.85	18.26	3.09	38.21	91.65
✓	✗	38.71	92.65	106.90	68.19	20.47	157.78	— — —
✗	✗	77.83	82.01	110.19	32.36	35.89	112.86	— — —

Tabla 4.4: Resumen del diagrama de cajas y bigotes para CG4.

- Tustin

Dado que es un método implícito requiere la solución de una ecuación algebraica no lineal cada vez que se actualiza el cuaternión. En consecuencia, la optimización es lenta y requiere altos recursos computacionales.

Los resultados obtenidos para cada una de las tres configuraciones evaluadas se presentan en la Figura 4.10

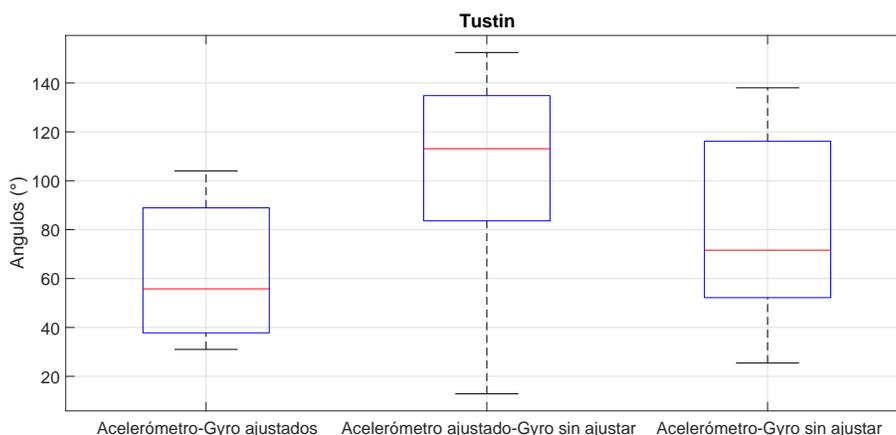


Figura 4.10: Ángulos α obtenido con las medidas sin ajustar y ajustadas Tustin

En la Tabla 4.5 se resumen los valores obtenidos del diagrama de cajas y bigotes.

Acelerometro	Giroscopio	Información del Diagrama de Cajas y Bigotes						
Ajustado	Sin ajustar	Q1	Q2	Q3	RIC	VMin	VMax	VAt
✓	✓	37.80°	55.77°	88.91°	51.11°	31.0092°	104.047°	— — —
✓	✗	83.58°	113.10°	134.82°	51.24°	12.8443°	152.4651°	— — —
✗	✗	52.18°	71.59°	116.14°	63.96°	25.43°	138.01°	— — —

Tabla 4.5: Resumen del diagrama de cajas y bigotes para Tustin

4.4.2.2. Análisis de resultados

Dado el resumen presentado para RK4 en la Tabla 4.3 , CG4 en la Tabla 4.4 y Tustin en la Tabla 4.5, se pueden observar los ángulos α obtenidos entre la orientación de referencia dada por el acelerómetro ajustado con respecto a la orientación estimada mediante la integración de las medidas del giroscopio antes y después de ser ajustadas.

Para RK4 en la Tabla 4.3, se puede observar que el error entre la orientación de referencia y la estimada esta:

- Sin ajustar, entre 80.30 ° - 126.43 ° y en promedio 94.66°.

- Ajustando, entre 6.87° - 73.96° y en promedio 23.72° .

Para CG4 en la Tabla 4.4, se puede observar que el error entre la orientación de referencia y la estimada esta:

- Sin ajustar, entre 35.89° - 112.86° y en promedio 82.01° .
- Ajustando, entre 3.09° - 38.21° y en promedio 22.67° .

Para Tustin en la Tabla 4.5, se puede observar que el error entre la orientación de referencia y la estimada esta:

- Sin ajustar, entre 25.43° - 138.01° y en promedio 71.59° .
- Ajustando, entre 31° - 104.047° y en promedio 55.77°

Los resultados presentados evidencian la importancia de ajustar las medidas antes de usar un dispositivo IMU para determinar la orientación de un cuerpo rígido, además se puede apreciar que el algoritmo CG4 obtiene errores mas pequeños entre la orientación de referencia y la estimada.

Finalmente, se determina la precisión para cada algoritmo empleando el valor del RIC, el cual nos indica la dispersión de los errores obtenidos. Siendo así, el algoritmo CG4 obtuvo la dispersión mas baja con una valor de 18.26° , seguido por el algoritmo de RK4 con un valor de 27.17° y por ultimo Tustin con un valor de 51.11° .

4.4.3. Runge-Kutta (RK4)

En esta sección el algoritmo de integración numérica RK4 se ejecuta para los siguientes escenarios:

1. La velocidad angular se supone constante durante todo el paso de integración.
2. La velocidad angular entre los puntos intermedios es la medida por el giroscopio.

Los resultados obtenidos para ambos escenarios se comparan para verificar la influencia de garantizar la velocidad constante durante el movimiento de la IMU en la obtención de datos.

En la Figura 4.11 se representa la distribución de los resultados.

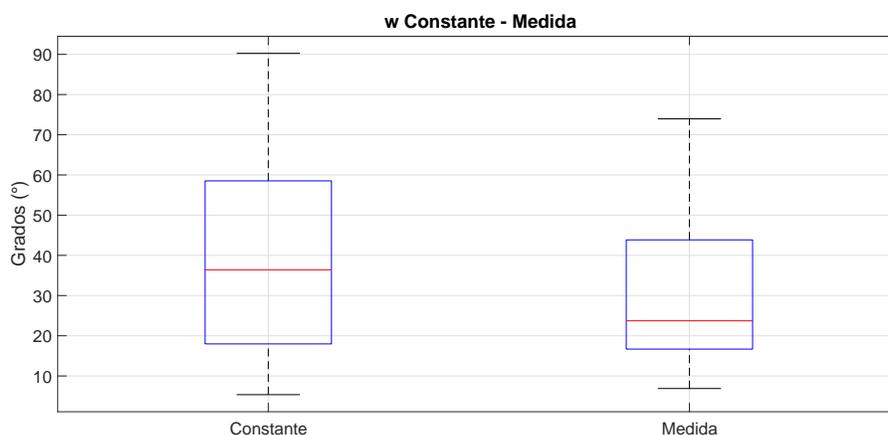


Figura 4.11: Velocidad angular constante y medida.

En la Tabla 4.6 se resumen los valores obtenidos del diagrama de cajas y bigotes.

	ω Constante	ω Medido
Q1	17.97	16.68
Mediana	36.40	23.72
Q3	58.62	43.85
RIC	40.65	27.17
V _{mín.}	5.35	6.87
V _{máx.}	90.25	73.96
V _{Ap}	-----	-----

Tabla 4.6: Resultados con ω constante-medida.

Los datos presentados en la Tabla 4.6 muestran que el RIC y, el valor de la mediana obtenidos es menor cuando se emplea la velocidad angular medida en el punto medio que requiere el método RK4, por tanto, se concluye que no es necesario garantizar la velocidad constante durante la obtención de las medidas de velocidad angular en los experimentos. Para el caso del algoritmo CG4 se obtuvieron resultados similares y para Tustin no fue necesario la evaluación debido que este método no evalúa la función en puntos intermedios.

4.5. Tiempo de ejecución para RK4, CG4 y Tustin.

El tiempo de ejecución de un algoritmo se compone del tiempo transcurrido (*Elapsed time*) y del tiempo de CPU (*CPU time*) el cual hace referencia al tiempo durante el cual la CPU estaba ocupada ejecutando la tarea, en ciertos casos el tiempo de CPU puede ser mayor que el tiempo transcurrido (*Elapsed time*) cuando se utilizan varios sub-procesos en un sistema. Uno de los casos ocurre cuando se usa más de una CPU para completar una tarea.

Para la ejecución de los algoritmos implementados se utilizó un ordenador con las siguientes características técnicas:

- Marca: Toshiba.
- Modelo: P845-S4200.
- Procesador: Core® i5 3ra. Generación (Intel(R) Core(TM) i5-317u CPU @ 1.70GHz)
- Tamaño de la Memoria Ram: 6 Gb.
- Capacidad Disco Duro: 750 Gb.

Para la medición del *Elapsed time* se utilizó la función *tic/toc* de MatLab® y para determinar el *CPU time* se usó la función *cputime* de la misma herramienta software. A continuación se presentan los datos obtenidos.

Algoritmo	Elapsed time (seg.)	CPU time (seg)	Promedio Elased time (min)
RK4.	106.51	155.62	1.77
	104.97	153.58	
	105.38	154.41	
	105.80	154.59	
	107.19	156.67	
CG4	1787.18	1842.91	35.37
	2220.05	2232.00	
	2189.65	2226.73	
	2268.21	2285.78	
	2114.97	2142.40	
Tustin	6772.11	10397.67	118.43
	6757.56	10044.12	
	7202.29	10721.34	
	7386.48	10981.19	
	7409.61	11029.61	

Tabla 4.7: Tiempos de ejecución para los algoritmos.

Con la tabla 4.7 se puede inferir que el algoritmo que tarda más tiempo en la estimación de parámetros es el de Tustin, esto debido a que es un método implícito, es decir, que para estimar el estado posterior se requiere del estado actual y posterior. Además, el algoritmo RK4 tarda alrededor de dos minutos en estimar los parámetros de ajuste, por tanto, si la plataforma fuese alimentada por una batería externa, sería conveniente utilizar algoritmos que requieran poco tiempo en la estimación de los parámetros en el ajuste de las medidas.

Capítulo 5

Conclusiones

Se diseñó y construyó un sistema mecatrónico compuesto por una plataforma modular con tres grados de libertad y una interfaz gráfica de usuario o GUI, para ajustar los parámetros de calibración de una IMU y reducir los errores agregados por el usuario en el procedimiento manual. El sistema se evaluó realizando adquisiciones de datos automatizadas para la MPU 6050 y ajustando sus medidas de aceleración lineal y velocidad angular desde la GUI. La prueba consistió en colocar la IMU en la plataforma y realizar diez trayectorias distintas de desplazamiento en tres estados consecutivos: reposo inicial, movimiento en el espacio, reposo final. A partir de los conjuntos de datos adquiridos se ajustó el acelerómetro empleando la gravedad local como referencia y las lecturas de la aceleración lineal en intervalos estáticos, luego se ajustó el giroscopio usando algoritmos de integración numérica y los valores de la orientación inicial y la velocidad angular entre dos posiciones estáticas consecutivas.

Para determinar la incidencia de los algoritmos de integración numérica en el procedimiento de ajuste de los parámetros de calibración se evaluó el desempeño de tres algoritmos: RK4, CG4 y Tustin. Se empleó como indicador de desempeño el ángulo α , formado por el vector de orientación estimado y el vector de orientación medido en el intervalo estático final. Así, entre más pequeño sea el valor del ángulo α obtenido por el algoritmo, mejor es su desempeño, pues el valor ideal para el indicador es cero. Los resultados obtenidos evidenciaron que los algoritmos RK4 y Tustin no preservaron la unicidad del cuaternión lo que disminuyó directamente su desempeño. Específicamente, los resultados menos satisfactorios fueron los obtenidos con Tustin, esto se debe a que Tustin no evalúa la derivada en puntos intermedios y emplea un paso de integración mayor. En contrapartida, los resultados más favorables fueron los generados con el algoritmo CG4, que preservó la unicidad del cuaternión y obtuvo una

baja dispersión en los ángulos α siendo el algoritmo que mejor se comportó.

La unicidad del cuaternión se obtuvo utilizando el algoritmo CG4 debido principalmente a dos características del método: La primera es que el algoritmo se realiza bajo una restricción, pues el cuaternión q_{k+1} se actualiza desde q_k suponiendo todos los cuaterniones en la superficie de una esfera unitaria en R^4 , la segunda es que se usa la teoría de grupos de Lie, con la cual se puede garantizar que cualquier cuaternión obtenido se encuentre dentro de la esfera unitaria. No obstante, el algoritmo requiere un tiempo de muestreo asíncrono para obtener la variable física medida por lo que se requeriría un mejor procesamiento en el hardware de adquisición, esto aumenta el costo del hardware, torna complejo el esquema de muestreo y conduce a que el algoritmo carezca de interés práctico. En contraste, el algoritmo RK4 es más práctico pues requiere un sencillo esquema de muestreo.

El RK4 evalúa la derivada en los puntos intermedios del paso de integración, por lo que en la evaluación de este método se obtuvieron resultados para dos escenarios: el primero en el que se suponía la velocidad constante y, el segundo en el que se empleó directamente la velocidad medida en dichos puntos. Los resultados mostraron que el algoritmo se comportó mejor cuando se ejecutaba el segundo escenario, por lo tanto, se concluye que no es necesario garantizar la velocidad constante durante la obtención de las medidas de velocidad angular en los experimentos.

Basados en el desempeño y la practicidad de los tres algoritmos evaluados, en este trabajo se considera que RK4 es el más viable, principalmente debido a su practicidad, pues aunque no garantiza la unicidad del cuaternión, su desempeño es aceptable y, la teoría sugiere que es posible mejorarlo significativamente si se emplea un tiempo de muestreo más pequeño en la adquisición de los datos.

La elección de la solución electrónica definitiva para el desarrollo de la plataforma GIM-VEBO cumplió satisfactoriamente los requerimientos básicos para su funcionamiento. Sin embargo, la tarjeta de adquisición que posee un reloj interno de baja oscilación a puesto limitantes en la capacidad de almacenamiento y además presenta un procesamiento lento. Esto impide que se pueda realizar pruebas para cada algoritmo con tiempos de muestreo más pequeños.

Finalmente, el sistema mecatrónico desarrollado permitió que el ajuste de los parámetros de calibración de la IMU se realizará en menos tiempo y con menor esfuerzo. Adicionalmente, los resultados reflejaron la importancia de ajustar los parámetros de calibración de una IMU sin importar el algoritmo, pues la diferencia entre los ángulos α para la IMU sin ajustar

respecto a una IMU ajustada son significativos.

5.1. Trabajos futuros.

- Realizar mejoras y ampliaciones al sistema mecatrónico desarrollado.
- Realizar modificaciones en el Hardware y en el Software para que el sistema almacene los parámetros de calibración directamente en la IMU.
- Validación del ajuste en los parámetros de la IMU empleando como patrón con el fin de estimar la incertidumbre y exactitud de la medida entregada por la IMU ajustada en el proyecto.
- Emplear una IMU ajustada con el sistema mecatrónico implementado en este trabajo, en proyectos que requieran medir velocidad, rotación y fuerzas gravitacionales de un cuerpo.
- Optimizar el protocolo y el algoritmo para otras referencias particulares de IMU, como por ejemplo la IMU contenida en el iPhone 8s [\[48\]](#).

Bibliografía

- [1] M. Karásek, A. J. Koopmans, S. F. Armanini, B. D. Remes, and G. C. De Croon, “Free flight force estimation of a 23.5 g flapping wing MAV using an on-board IMU,” in *IEEE International Conference on Intelligent Robots and Systems*, 2016.
- [2] V. Grabe, H. H. Bulthoff, and P. R. Giordano, “A comparison of scale estimation schemes for a quadrotor UAV based on optical flow and IMU measurements,” in *IEEE International Conference on Intelligent Robots and Systems*, 2013.
- [3] R. Mebarki, J. Cacace, and V. Lippiello, “Velocity estimation of an UAV using visual and IMU data in a GPS-denied environment,” in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2013*, 2013.
- [4] D. Unsal and K. Demirbas, “Estimation of deterministic and stochastic IMU error parameters,” in *Record - IEEE PLANS, Position Location and Navigation Symposium*, 2012.
- [5] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2004.
- [6] D. Tedaldi, A. Pretto, and E. Menegatti, “A robust and easy to implement method for IMU calibration without external equipments,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2014.
- [7] A. Pretto and G. Grisetti, “Calibration and performance evaluation of low-cost IMUs,” *18th International Workshop on ADC Modelling and Testing*, 2014.
- [8] S. Fux., “Development of a planar low cost Inertial Measurement Unit for UAVs and MAVs.” Master’s thesis, ETH Zurich,, 2008.
- [9] M. Tatkow, “Acutronic,” 1973. [Online]. Available: <http://www.acutronic.com/>
- [10] A. Olivares, G. Olivares, J. M. G??rriz, and J. Ram??rez, “High-efficiency low-cost accelerometer-aided gyroscope calibration,” in *Proceedings of the International Symposium on Test and Measurement*, 2009.

- [11] L. E. A. Perez and J. S. E. Garzón, “Diseño, calibración y prueba de una Unidad de Medida Inercial (IMU) inalámbrica para cohetes,” *Ingenium*, 2012.
- [12] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy, “A new multi-position calibration method for MEMS inertial navigation systems,” *Measurement Science and Technology*, 2007.
- [13] J. J. Hall and R. L. Williams, “Case study: Inertial measurement unit calibration platform,” *Journal of Robotic Systems*, 2000.
- [14] P. d. I. U. S. Bolivar, “Grupo de Investigación y Desarrollo en Mecatronica.” [Online]. Available: <http://www.labc.usb.ve/mecatronica/>
- [15] R. Siegwart, “Autonomous Systems Lab.” [Online]. Available: <http://www.asl.ethz.ch/>
- [16] A. Poddar, Shashi and Kumar, Vipin and Kumar, “A Comprehensive Overview of Inertial Sensor Calibration Techniques,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. American Society of Mechanical Engineers, p. 011006, 2017.
- [17] M. Parvis and F. Ferraris, “Procedure for effortless in-field calibration of three-axial rate gyro and accelerometers,” 1995.
- [18] N. Shin, Eun-Hwan and El-Sheimy, “A new calibration method for strapdown inertial navigation systems,” *Z. Vermess*, vol. 127, pp. 1—10, 2002.
- [19] D. Jurman, M. Jankovec, R. Kamnik, and M. Topič, “Calibration and data fusion solution for the miniature attitude and heading reference system,” *Sensors and Actuators A: Physical*, vol. 138, no. 2, pp. 411–420, 2007.
- [20] E. Luhta and K. Roy, “6 - Gimbal Lock,” in *How to Cheat in Maya 2012*, E. Luhta and K. Roy, Eds. Boston: Focal Press, 2012, pp. 137–151. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780240816982500060>
- [21] H. O. Amadi and A. M. J. Bull, “A motion-decomposition approach to address gimbal lock in the 3-cylinder open chain mechanism description of a joint coordinate system at the glenohumeral joint,” *Journal of Biomechanics*, 2010.
- [22] A. W. Watt and Mark, “Advanced Animation and Rendering Techniques: Theory and Practice,” 1992.

- [23] C. M. Cheuk, T. K. Lau, K. W. Lin, and Y. Liu, "Automatic Calibration for Inertial Measurement Unit," *12th International Conference on Control, Automation, Robotics & Vision*, 2012.
- [24] W. T. Fong, S. K. Ong, and A. Y. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," *Measurement Science and Technology*, 2008.
- [25] M. S. Andrie and J. L. Crassidis, "Geometric Integration of Quaternions," *Journal of Guidance, Control, and Dynamics*, 2013.
- [26] Solidworks®, "CAD 3D," 2015.
- [27] Mathworks®, "Matlab," 2016.
- [28] A. Chatfield, *Fundamentals of high accuracy inertial navigation*, 1997.
- [29] "https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/." [Online]. Available: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- [30] K. R. Britting, "Inertial navigation systems analysis." 1971.
- [31] C. Jekeli, "Inertial Navigation Systems with Geodetic Applications," *De Gruyter*, 2001.
- [32] I. Skog and H. Peter, "Calibration of a MEMS inertial measurement unit," *International Measurement Confederation*, 2006.
- [33] K. S. & T. Duff, "Matrix animation and polar decomposition," 1994.
- [34] S. W. R. Hamilton., "Elements of Quaternions," vol. volume 1-2, no. Longmans, Green and Co., 1899.
- [35] S. W. R. Hamilton, "Lectures on Quaternions," no. Hodges and Smith, 1853.
- [36] K. Shoemake., "Animating rotation with quaternion curves, Computer Graphics," 1985.
- [37] K. Shoemake, "Quaternions," 1994.
- [38] A. Witkin and D. Baraff, "Physically Based Modeling : Principles and Practice," *Differential Equations*, 1997.
- [39] B. M. Saunders., "Fast Animation Dynamics," PhD thesis, University of Cambridge Computer Laboratory, 2000.

- [40] S. Card, X. Parc, J. Grudin, J. Nielsen, and N. N. Group, “The Morgan Kaufmann Series in Interactive 3D Technology,” *Real-Time Collision Detection*, 2005.
- [41] S. a. Whitmore, “Closed-Form Integrator for the Quaternion (Euler Angle) Kinematics Equations,” *Ntrs.Nasa.Gov*, 2000.
- [42] G. C. Francisco Solís, “Discretización de ecuaciones diferenciales no lineales unimodales,” *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería.*, vol. 19, pp. 197–210, 2003.
- [43] B. O. Z. Jackiewicz, A. Marthinsen, “Construction of Runge-Kutta Methods of Crouch-Grossman Type of High Order,” *Advanced Computational Mathematics*, vol. 13, pp. 405–415, 2000.
- [44] B. O. Marthinsen, “Runge-Kutta Methods Adapted to Manifolds and Based on Rigid Frames,” pp. 116–142, 1999.
- [45] ABS, “Acrilonitrilo butadieno estireno.” [Online]. Available: <https://ultimaker.com/download/67619/TDSABSV3.011-spa-ES.pdf>
- [46] D. G. Gonzales and C. Martinez, “Aplicación del método morphol,” in *Seminario de Prospectiva y Pensamiento Estratégico*, vol. Instituto, 2014.
- [47] M. I. Products, “SensorsONE.” [Online]. Available: <https://www.sensorsone.com/local-gravity-calculator/{#}latitude>
- [48] “[https://www.apple.com/us_kiosk_423720/shop/buy-iphone/iphone-8.](https://www.apple.com/us_kiosk_423720/shop/buy-iphone/iphone-8)” [Online]. Available: https://www.apple.com/us_{_}kiosk_{_}423720/shop/buy-iphone/iphone-8

Anexos

Anexo A

Análisis estático-movimiento

A través de la Ingeniería Asistida por Computadora (CAE, *Computer assisted engineering*) se puede modelar cualquier pieza o sistema con una precisión prácticamente real. Ayudándose de análisis estáticos, se puede prever su comportamiento bajo condiciones reales de trabajo, sin necesidad de prototipos. Este tipo de análisis reduce el coste de las pruebas de campo ya que puede simular previamente el comportamiento del modelo, y permite mejorar el producto rápidamente mediante el rediseño del modelo de las piezas de forma virtual antes de la fabricación.

A.1. Análisis estático.

Solidworks® cuenta con el toolbox *Simulation* que nos permite aplicar cargas a una pieza o sólido, el efecto de las cargas se transmite a través del sólido o pieza y generan una deformación en su estructura. El análisis estático lineal calcula los valores de desplazamientos, deformaciones unitarias, tensiones y fuerzas de reacción generados por las cargas aplicadas.

El análisis estático lineal realiza las siguientes suposiciones:

- Suposición estática: La fuerza se aplican lenta y gradualmente hasta alcanzar el valor final de su magnitud, a partir de este punto las cargas permanecerán constantes. Debido a esto se puede despreciar las fuerzas inerciales y de amortiguamiento, ya que las velocidades y aceleraciones son pequeñas.

- **Linealidad:** Las fuerzas aplicadas son directamente proporcionales a las deformaciones, tensiones y desplazamientos generados en el sólido. Además las cargas son constantes en magnitud, dirección y distribución, por tanto no cambian durante la deformación del modelo.

Para la interpretación del análisis estático en este caso de estudio se presentaran tres gráficas: tensiones, desplazamiento y deformación unitaria. A continuación se hace una breve descripción de cada propiedad física.

1. **Tensión:** Fuerza aplicada que intenta estirar el material a lo largo de su línea de acción.



Figura A.1: Cuerpo sometido a Tensión

2. **Desplazamiento:** Cambio de posición de un cuerpo en dos instantes.

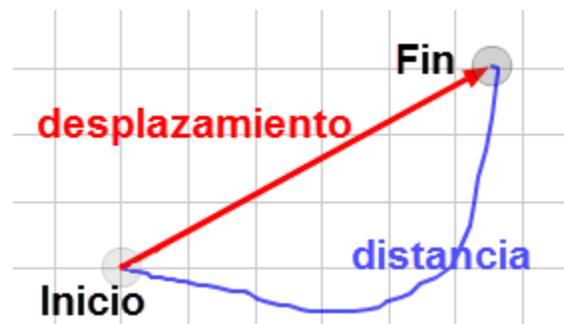


Figura A.2: Desplazamiento

3. **Deformación Unitaria:** Relación existente entre desplazamiento total (δ) y estado inicial del elemento (L), como se ilustra la Figura A.3. En este caso de estudio fue necesario implementar la deformación unitaria para determinar cuan resistente es el modelo.

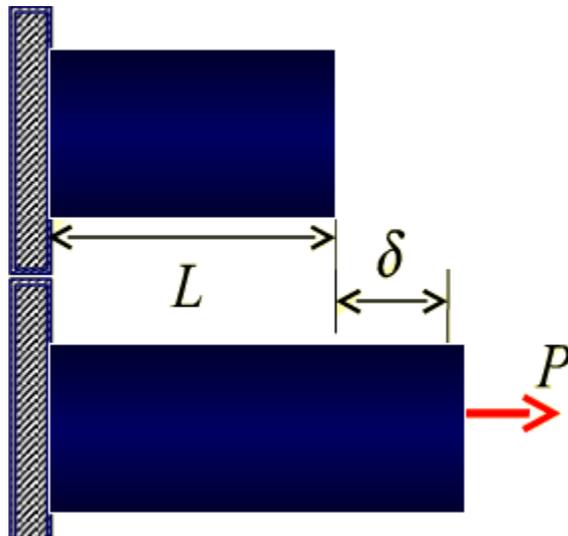


Figura A.3: Deformación unitaria

Para una mejor interpretación de los resultados se muestra la ecuación general de la deformación unitaria.

$$\epsilon = \frac{\delta}{L} \quad (\text{A.1})$$

En el análisis estático Solidworks entrega un resultado *ESTRN A.5* (Deformación unitaria equivalente) la cual esta definida por:

EPSX	Deformación unitaria normal X
EPSY	Deformación unitaria normal Y
EPSZ	Deformación unitaria normal Z
GMXY	Tensión cortante en dirección Y en plano YZ
GMXZ	Tensión cortante en dirección Z en plano YZ
GMYZ	Tensión cortante en dirección Z en plano XZ
E	Deformación unitaria promedio
E1	Deformación unitaria normal en la primera dirección principal
E2	Deformación unitaria normal en la segunda dirección principal

Tabla A.1: Componentes deformación unitaria.

Donde:

$$E = \frac{(EPSX + EPSY + EPSZ)}{3} \quad (\text{A.2})$$

$$E1 = \frac{(EPSX - E)^2 + (EPSY - E)^2 + (EPSZ - E)^2}{2} \quad (\text{A.3})$$

$$E2 = \frac{(GMXY)^2 + (GMXZ)^2 + (GMYZ)^2}{4} \quad (\text{A.4})$$

Por tanto:

$$ESTRN = 2\sqrt{\frac{(E1 + E2)}{3}} \quad (\text{A.5})$$

En los experimentos efectuados se aplicó una aceleración gravitacional de $9,81m/s^2$ y una fuerza externa de 2 Kgf.

A.1.1. Base

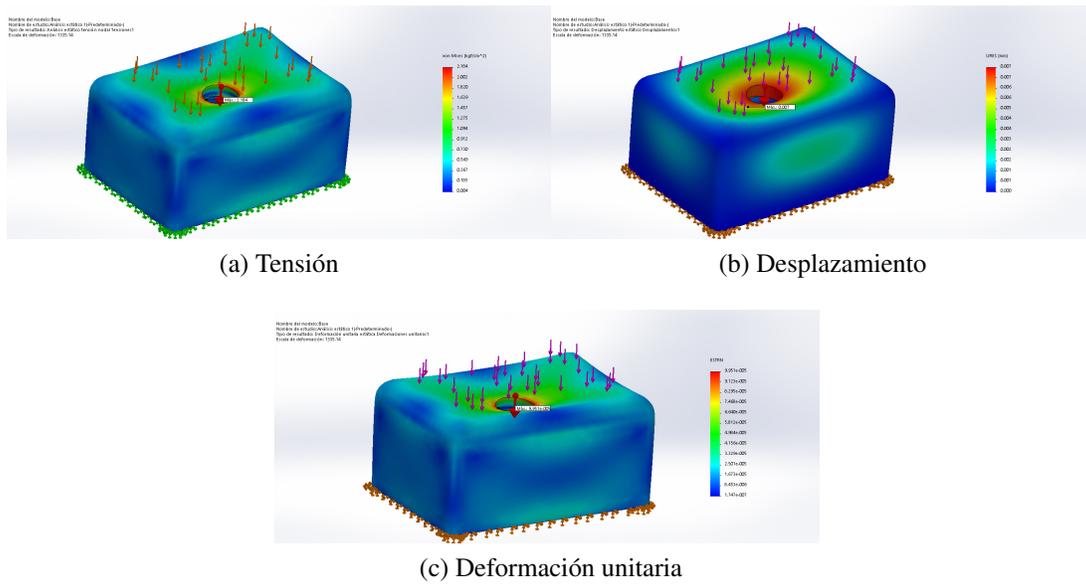


Figura A.4: Base

Los resultados del análisis estático lineal ejecutado en la pieza *base*, evidencian una deformación considerable en la parte donde se sujeta el servomotor, debido a que el modelo posee un vaciado para la sujeción del servomotor.

A.1.2. Brazo 1.

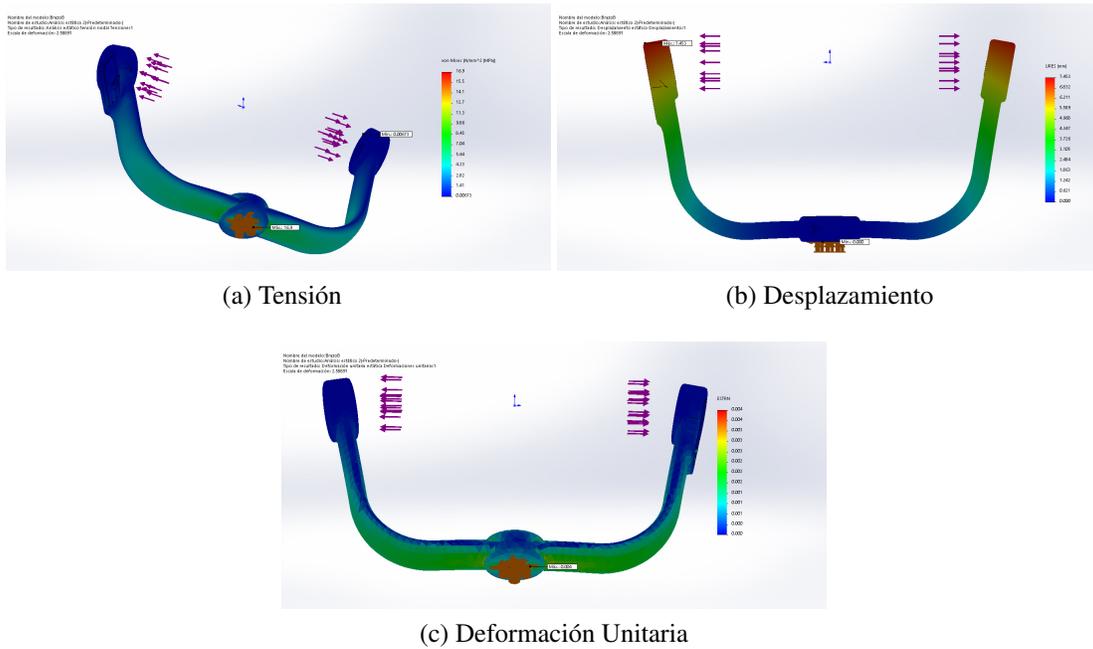


Figura A.5: Brazo 1

En el *brazo 1* se evidencia que el punto con mayor tensión es una de las sujeciones al servomotor Dynamixel, el mayor desplazamiento se presenta en los extremos superiores de los brazos, justo en los puntos donde se aplicó la fuerza externa.

A.1.3. Brazo 2

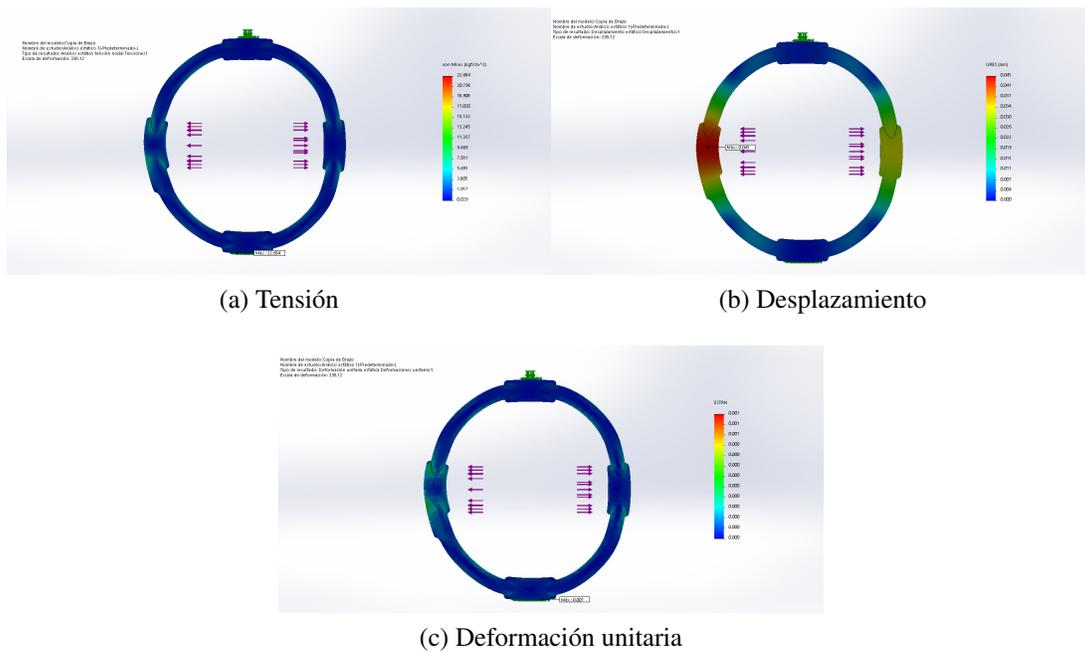


Figura A.6: Brazo 2.

Para el análisis estático lineal del *brazo 2* se aprecia que el valor de $ESTRN$ es muy pequeño, por tanto, la deformación para este modelo son pequeñas en comparación con las otras piezas.

A.1.4. Brazo 3

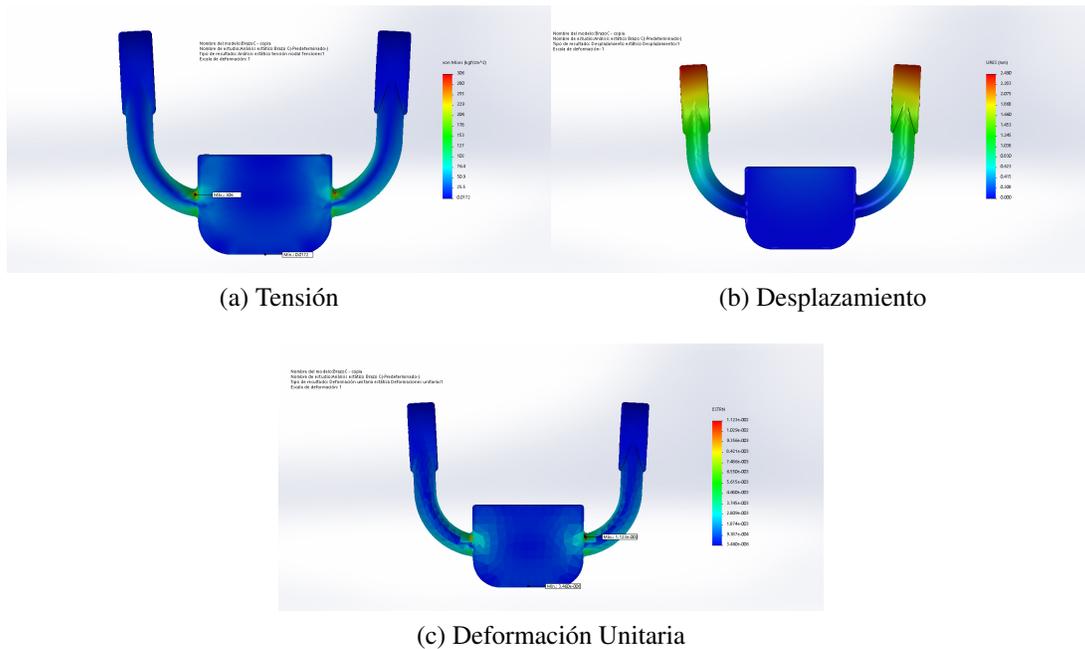


Figura A.7: Brazo 3.

En el *brazo 3* se aprecia un *ESTRN* considerable en un nodo del brazo, por tanto, en caso de que suceda un evento con condiciones similares a las del experimento este será el punto de ruptura. El desplazamiento del *brazo 3* es similar al *brazo 1* solo que con valores de desplazamiento mas pequeños.

Con lo anterior se determino que el diseño de la plataforma y el material seleccionado es una opción viable, ya que las condiciones presentadas en los experimentos poseen una gran magnitud de fuerza centrífuga, y la GIMVEBO no será capaz de generar un evento similar.

A.2. Estudio de movimiento.

Se evaluó el comportamiento y movimiento del modelo durante un ciclo operativo empleando el toolbox *SOLIDWORKS Motion*. El funcionamiento de este toolbox permite determinar cuando se presenta alguna eventualidad durante un experimento y notifica al usuario con el problema e información de ayuda.

Para este caso de estudio se realizó un ensamble general de toda la plataforma, se definieron las relaciones de posición y se incluyeron tres motores para generar el movimiento, además se tuvo en cuenta la aceleración gravitacional y una fuerza externa. En la Figura A.8 se exponen los resultados del estudio de movimiento.

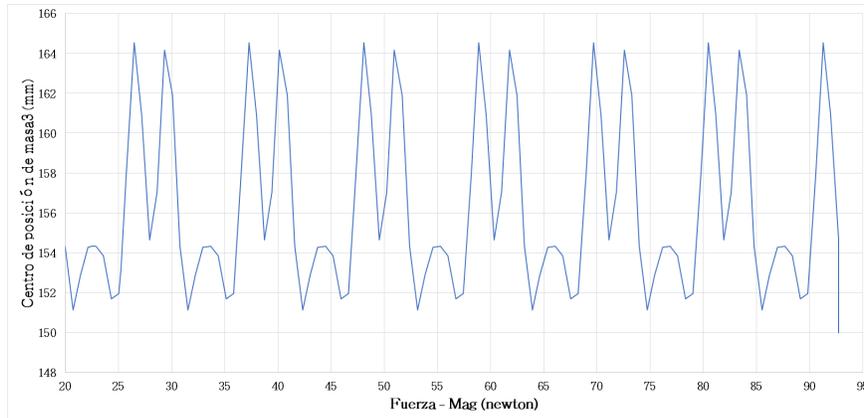


Figura A.8: Análisis de movimiento

La Figura A.8 hace referencia al desplazamiento en milímetros (mm) del centro de masa del *Brazo 3* respecto a la fuerza centrífuga o de reacción. Al ser movimientos cíclicos el desplazamiento del centro de masa del *brazo 3* se repite en un determinado tiempo. En el momento en el que la fuerza externa supera el valor máximo soportado por la plataforma, el centro de masa del *brazo 3* cambia abruptamente y tiende a cero, por tanto indica que la pieza *brazo 3* de la GIMVEBO se desensambló.

Anexo B

Manual de usuario

El presente manual tiene como propósito guiar al usuario en el proceso de estimación de parámetros utilizando la plataforma GIMVEBO. El manual se divide en dos secciones: en la primera sección se describe la instalación y configuración de las herramientas software, en la segunda sección se detalla el procedimiento de estimación de parámetros utilizando la aplicación software y la plataforma. Es importante aclarar que el desarrollo software del presente trabajo se realizó utilizando el sistema operativo Windows 10.

B.0.1. Instalación de herramientas software.

A continuación se da una descripción general de las páginas de descarga, los pasos para la descarga e instalación.

B.0.1.1. ARDUINO IDE

Un entorno de desarrollo integrado (IDE del inglés Integrated Development Environment), es una aplicación que facilita al usuario el desarrollo de software. Para descargar e instalar la aplicación *Arduino IDE* se debe seguir los siguientes pasos:

1. Obtener la última versión de la aplicación, en este caso se realiza la descarga desde *Microsoft Store*, también lo puede hacer desde su página de [descarga](#). Para acceder a *Microsoft Store* presione la tecla Windows, escriba *Microsoft Store* y ejecútela.

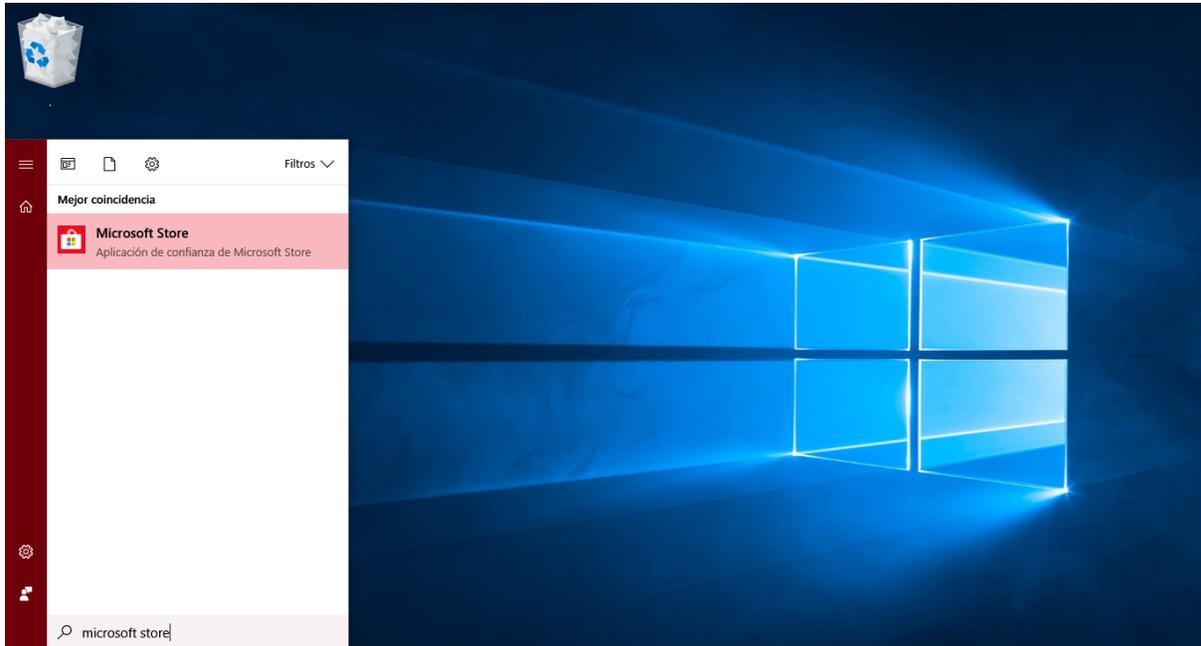


Figura B.1: *Microsoft Store*

2. En el buscador de la aplicación escribir *Arduino IDE*.

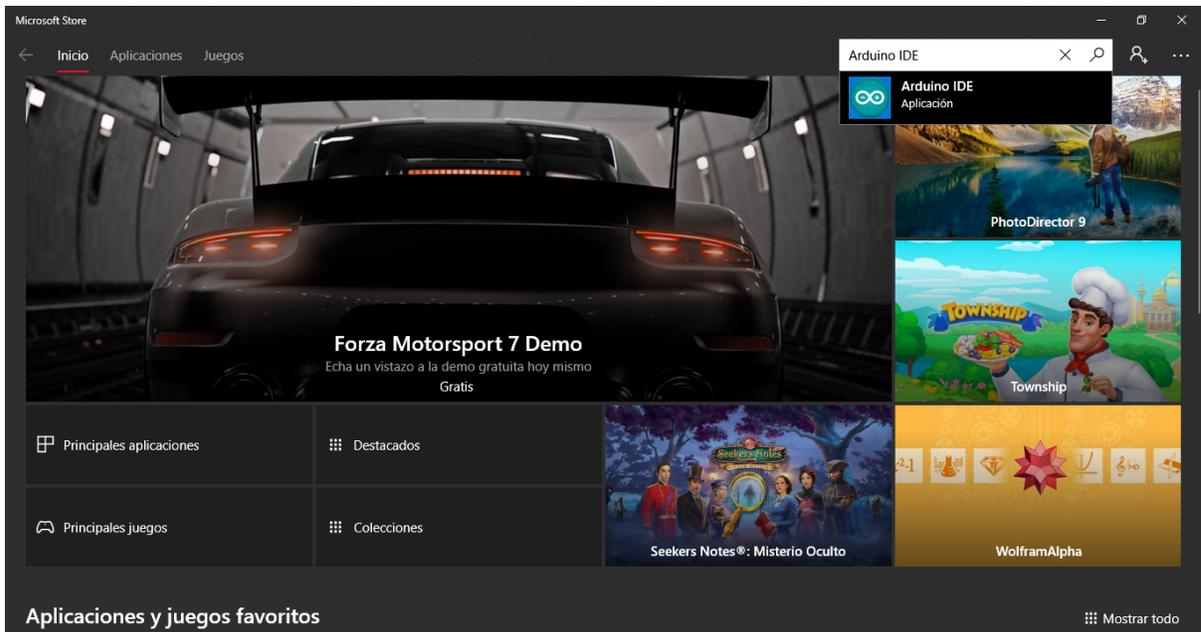


Figura B.2: *Microsoft Store - Arduino IDE*.

3. Clic en la opción Obtener y la aplicación se instalará.

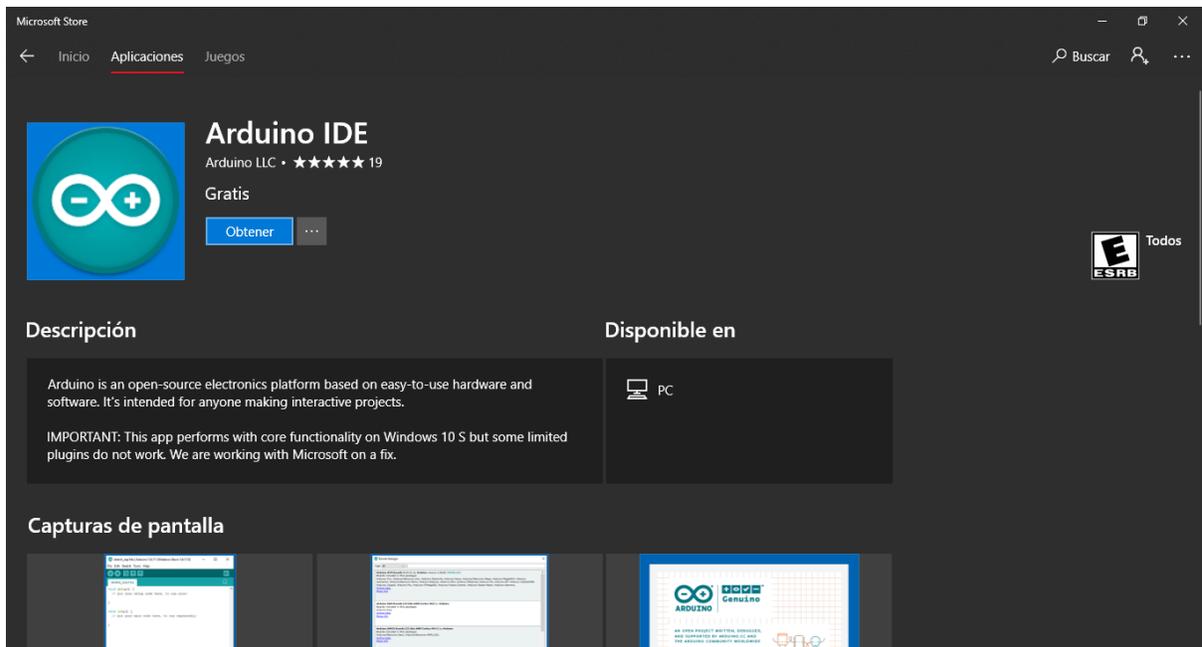


Figura B.3: *Arduino IDE*.

4. Incluir las librerías *I2Cdev.h* y *MPU6050.h*.

B.0.1.2. Robotics_OpenCM IDE

Es la aplicación que permite desarrollar software para los elementos de la marca Robotics[®], a continuación se hace una breve descripción sobre su descarga e instalación:

1. Ingresar a la página principal **Robotics**, seleccione y descargue la opción para sistema operativo Windows.

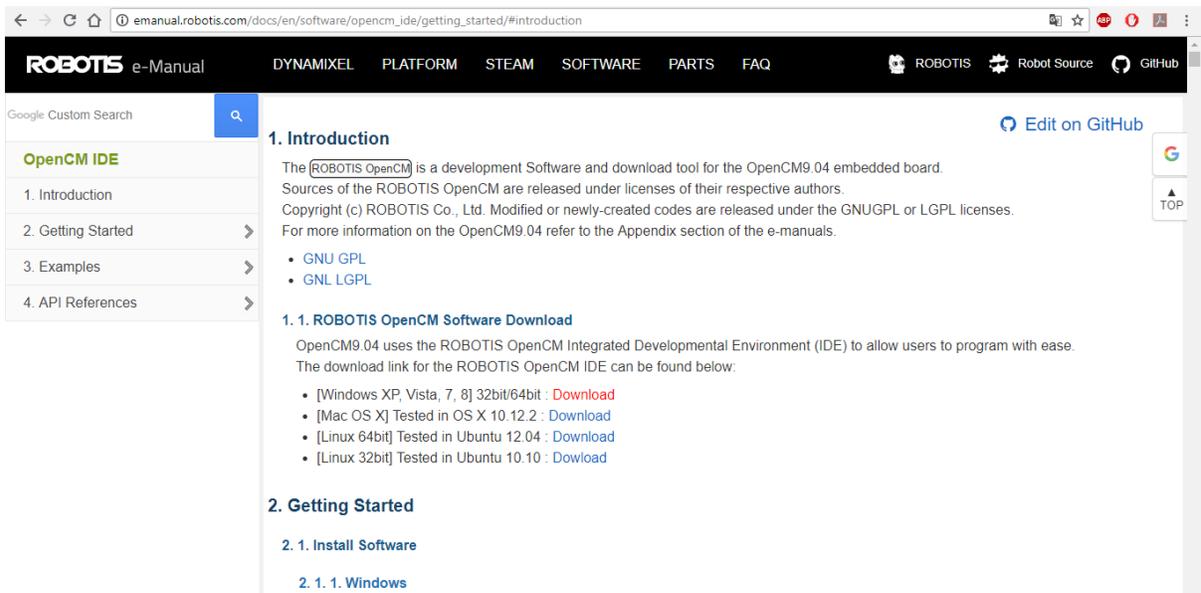


Figura B.4: Pagina descarga Robotics.

2. Descomprimir el archivo descargado y ejecute el archivo *ROBOTIS_OpenCM*.

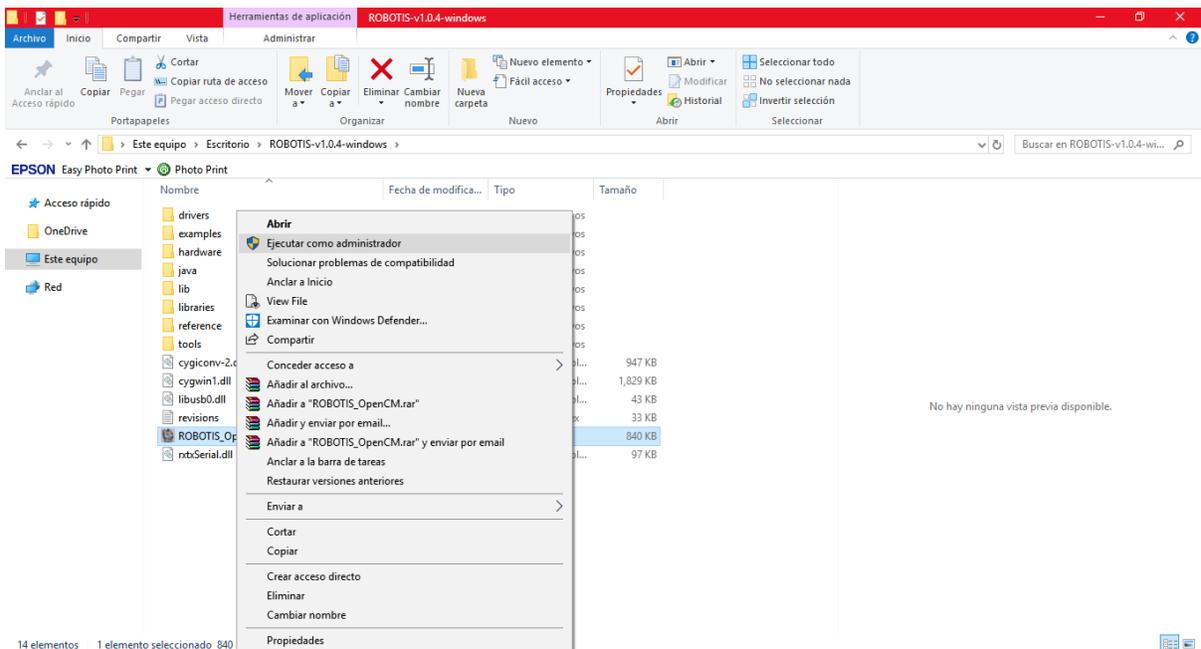


Figura B.5: Contenido archivo descargado.

B.0.2. Procedimiento de estimación de parámetros.

Antes de iniciar el procedimiento asegúrese de haber identificado las partes de la plataforma GIMVEBO y las aplicaciones software a utilizar.

1. Fijar la base de la plataforma GIMVEBO a una estructura plana y estable, para ello utilice las cuatro perforaciones ubicadas en cada esquina de la base.
2. Conectar el adaptador de la plataforma GIMVEBO a la red eléctrica.
3. Conectar el puerto USB *micro-b* de la GIMVEBO y arduino pro-micro a un puerto USB del ordenador donde se encuentra instaladas las aplicaciones software.
4. Verificar que la plataforma y la tarjeta Arduino se encuentren energizadas.
5. En el ordenador diríjase a *Inicio > clic derecho > Administrador de dispositivos > Puertos de comunicación* y verifique que estén activos el puerto de la OpenCM y el de Arduino.
6. En la aplicación Arduino IDE cargue el sketch *Adquisicion.ino*.
7. Configure en la barra de herramientas opción *Herramientas > Placa* la opción *Arduino/Genuino Micro*
8. Seleccione el puerto de comunicación *Herramientas > Puerto > COM(11)Arduino ProMicro* (el número asignado a cada puerto de comunicación COM puede variar).
9. En la aplicación Robotics_OpenCM IDE cargue el sketch *Control.ino*.
10. Configure en la barra de herramientas opción *Herramientas > Tarjeta* la opción *ROBOTICS OpenCm9.04*.
11. Seleccione el puerto serial de comunicación asignado a la tarjeta OpenCM9.04 *Herramientas > Puerto Serial > COM(12)*.
12. Ejecute la aplicación Gimvebo SW.
13. En la opción puerto serial establezca el mismo COM asignado a la arduino Pro-micro y la tarjeta OpenCM9.04.
14. Configure el número de aceleraciones lineales a obtener y el numero de experimentos que desea para la prueba.

15. En la siguiente ventana haga clic sobre el icono  para iniciar la prueba de adquisición.
16. Presione el botón de ajuste de medias para estimar los parámetros del acelerómetro y giroscopio empleando los algoritmos de integración numérica.

Anexo C

Características eléctricas, electrónicas, mecánicas y económicas

C.0.0.1. Fuentes de alimentación y potencia.

Fuentes de alimentación			
Tipo	Switching power adaptor HW-050200U33	Regulador de impulso ajustable Pololu 2.5-9.5V	Puerto serial (Computador)
Tensión de alimentación	100-240 V - 50/60 Hz	1.5 V - 16 V	10.8 V DC 4200 mAh
Tensión de salida	5 V DC	2.5 V - 9.5 V (ajustable)	5 V DC
Corriente Máxima	2 A	80%-90% corriente entrada	500 mA
Especificaciones económicas			
Costo (COP)	\$25.000	\$43000	X

Tabla C.1: Características fuentes de alimentación.

C.0.0.2. Captura

Sensor	
Marca	Invensense
Tipo	Unidad de procesamiento de movimiento
Modelo	MPU6050/GY-521
Tecnología	MEMS Sistemas Microelectromecánicos
Especificaciones Mecánicas	
Dimensiones	21.2 mm x 16.4 mm x 3.3 mm
Tecnología de fabricación	CMOS
Peso	2,1 g
Rango Acelerómetro	+/- 2 g, +/- 4 g, +/- 8 g, +/- 16 g donde (g = gravedad)
Rango giroscopio	+/- 250, +/- 500, +/- 1000, +/- 2,000 ° / s
Especificaciones Eléctricas y de Temperatura	
Voltaje de operación	3.3 VDC - 5 VDC
Corriente nominal	3.9 mA
Convertor AD	16 bits
Temperatura de operación	-40 °C - 85 °C
Especificaciones de Comunicación	
Protocolos de comunicación	I2C/SPI
Especificaciones económicas	
Precio a la fecha (COP) incluido IVA	\$15000

Tabla C.2: Características MPU6050

C.0.0.3. Movimiento

Motores				
Marca	Robotics	Robotics	Maytech	Casun
Modelo	Dynamixel XL320	Dynamixel AX-12	Brushless A2212/13T- 1000-Kv	Paso a paso Nema 17
Tipo	Servomotor	Servomotor	Motor Eléctrico sin escobillas	Motor de pasos de rotor de imán permanente
Especificaciones Mecánicas				
Material de cobertura	Plástico	Plástico	Aluminio	Acero
Dimensiones	24 mm x 36 mm x 27mm	32mm x 50 mm x 40 mm	40 mm x 28 mm	42.3 mm x 42.3 mm x 40 mm
Peso	16.7 g	54.6 g	73 g	57 g
Torque	0.39 Nm	1.5 Nm	3 Nm	450 Nm
Rango de operación	300°	300°	360°	360°
Precisión	0.29°	0.29°	1.9°	1.8°
Especificaciones Eléctricas y de Temperatura				
Tensión de alimentación	6 - 8.4V (Recomendado : 7.4V)	9 - 12V (Recomendado 11.1V)	5.6 - 16.8 V	3.75 - 12 V
Temperatura de operación	-5 °C - +70 °C	-5 °C - +70 °C	-2 °C - +50 °C	-6 °C - +60 °C
Especificaciones de Comunicación				
Comunicación	Half duplex Asynchronous Serial Communication	Half duplex Asynchronous Serial Communication	serial USART AVR	serial USART AVR
Especificaciones Económicas				
Precio a la fecha en (COP) incluido IVA	\$ 120.000	\$215.000	\$124.974	\$68.000

Tabla C.3: Características motores.

C.0.0.4. Control

Tarjetas de control para actuadores			
Marca	Robotics	RobotDyn	Arduino
Modelo	OpenCM	Mega 2560 mini	Arduiono uno
Especificaciones Mecánicas			
Dimensiones	66.5 mm x 27 mm	52 mm x 38 mm	68.6 mm x 63.4 mm
Tecnología de fabricación	CMOS	CMOS	CMOS
Conectores	USB micro tipo B	USB micro tipo B	USB Tipo B
Especificaciones Eléctricas			
Tensión de alimentación	5 V - 16 V	7 V - 12 V	7 V - 12 V
Procesamiento de bits	16	8	8
Memoria de programa	FLASH	FLASH	FLASH
Memoria de datos	RAM 20kbytes EEPROM 118kbytes	RAM 8 kbytes EEPROM 4 kbytes	RAM 2 kbytes EEPROM 1 kbytes
Frecuencia de oscilador	72 MHz	16 MHz	16 MHz
Especificaciones de comunicación			
USART	3	1	1
TTL BUS	4	X	X
Especificaciones Económicas			
Precio a la fecha (COP) incluido IVA	\$106.000	\$45.000	\$66.000

Tabla C.4: Características tarjetas de control para actuadores

C.0.0.5. Almacenamiento

Tarjetas adquisición y almacenamiento de datos		
Marca	Sparkfun	Arduino
Modelo	pro micro	nano
Especificaciones mecánicas		
Dimensiones	18 mm x 36 mm	19 mm x 46 mm
Tecnología de fabricación	CMOS	CMOS
Conectores	USB micro tipo B	USB micro tipo B
Especificaciones Eléctricas		
Tensión de alimentación	6 V - 16 V	7 V - 12 V
Procesamiento de bits	8	8
Memoria de programa	FLASH	FLASH
Memoria de datos	RAM 1.5kbytes EEPROM 1kbytes	RAM 1kbytes EEPROM 512kbytes
Frecuencia de oscilador	16 MHz	16 MHz
Especificaciones de comunicación		
USART	1	1
I2C	1	1
Especificaciones Económicas		
Precio a la fecha (COP) incluido IVA	\$25.000	\$68.500

Tabla C.5: Características tarjetas adquisición y almacenamiento de datos

C.0.0.6. Trasmisión.

Transmisión		
Tipo	Radio Frecuencia HC11	Cableado
Comunicaciones	Half duplex	USB serial Half duplex
Dimensiones	13.5mm x 28.2mm	X
Especificaciones eléctricas		
Tensión de alimentación	3.3 V - 5 V	5 V
Consumo de corriente en reposo	80 uA	X
Consumo de corriente activo	3.5 mA	0.1 mA
Especificaciones Económicas		
Precio a la fecha (COP) incluido IVA	\$15.000	\$10.000

Tabla C.6: Características transmisión.