



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN ELECTRÓNICA,
TELECOMUNICACIONES Y REDES

**“IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN
INERCIAL, PARA MEJORAR LA PRECISIÓN DE
POSICIONAMIENTO DE UN PROTOTIPO GPS EN UNA
TRAYECTORIA DENTRO DE LA ESPOCH”**

TRABAJO DE TITULACIÓN: PROPUESTA TECNOLÓGICA
Para optar al Grado Académico de:
INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES

AUTORES: LORENA ELIZABETH CUENCA SÁENZ
JORGE ALFONSO LEÓN CANDO
TUTORA: ING. MÓNICA ZABALA HARO, MSC.

Riobamba-Ecuador
2017

©2017, Lorena Elizabeth Cuenca Sáenz & Jorge Alfonso León Cando

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA, TELECOMUNICACIONES Y REDES

El tribunal del trabajo de titulación certifica que: El trabajo de titulación: “IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN INERCIAL, PARA MEJORAR LA PRECISIÓN DE POSICIONAMIENTO DE UN PROTOTIPO GPS EN UNA TRAYECTORIA DENTRO DE LA ESPOCH”, de responsabilidad de la señorita Lorena Elizabeth Cuenca Sáenz y el señor Jorge Alfonso León Cando, ha sido revisado de forma minuciosa por los miembros del tribunal del trabajo de titulación, quedando autorizado su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna E. DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Franklin Moreno. DIRECTOR DE LA ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES	_____	_____
Ing. Mónica Zabala Haro, Msc. DIRECTORA DEL TRABAJO DE TITULACIÓN	_____	_____
Ing. Raúl Lozada. MIEMBRO DEL TRIBUNAL	_____	_____

“Nosotros, Lorena Elizabeth Cuenca Sáenz y Jorge Alfonso León Cando, somos responsables de las ideas, doctrinas y resultados, expuestos en el trabajo de titulación, y el patrimonio intelectual del Trabajo de Titulación pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**.

Lorena Elizabeth Cuenca Sáenz

Jorge Alfonso León Cando

DEDICATORIA

Dedico este trabajo a Dios, por haberme dado su amor, su fortaleza para seguir adelante y haberme dado vida y salud para lograr mis metas.

A mi madre Martha Sáenz Ruiz, por su apoyo incondicional, por sus consejos y por su motivación para alcanzar mis metas.

A mi padre Victor Cuenca, por su ejemplo de perseverancia que me ha enseñado a no rendirme, por su gran esfuerzo y apoyo para realizar mis metas.

Lorena Cuenca Sáenz

Dedico el presente trabajo a mis padres, esposa y familia quienes me brindaron su amor y calidez de hogar en todo momento, confiando en mí durante todo mi carrera estudiantil, siendo una gran fuente de confianza e inspiración para superarme día a día y llegar a cumplir la meta de ser un profesional.

Jorge León Cando

AGRADECIMIENTO

Agradezco a Dios por sus bendiciones, por su bondad y por la fortaleza para alcanzar mis objetivos, a mis padres por sus consejos y motivación para no rendirme en alcanzar mis objetivos.

A la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, en especial a la Escuela de Ingeniería en Electrónica, telecomunicaciones y Redes por haberme dado la oportunidad de estudiar y formarme como un profesional.

Agradezco a la Ing. Mónica Zabala e Ing. Raúl Lozada por su asesoramiento y dedicación, quienes con sus conocimientos, paciencia y experiencia hemos logrado culminar el presente documento.

Lorena Cuenca Sáenz

Agradesco a Dios, por llenarme de bendiciones durante todo el transcurso de mi vida estudiantil, brindándome salud y vida para día a día dar lo mejor de mí para cumplir una meta muy importante en vida.

A mis padres, esposa, hijos, hermanos, sobrinos quienes estuvieron junto a mí brindándome su apoyo de manera incondicional, siendo una gran fuente de inspiración en todo momento.

Especialmente a la Ing. Mónica Zabala e Ing. Raúl Losada, docentes y amigos que han sido un gran aporte e inspiración profesional, quienes nos ayudaron con sus conocimientos para poder desarrollar de la mejor manera el presente trabajo, a todos muchas gracias.

Jorge León Cando

TABLA DE CONTENIDO

RESUMEN	xvi
SUMMARY	xvii
INTRODUCCIÓN	1

CAPÍTULO I

1. MARCO TEÓRICO	6
1.1. Navegación.....	6
1.2. Navegación Inercial.....	6
<i>1.2.1. Sistemas de coordenadas.....</i>	<i>7</i>
<i>1.2.2. Sistemas de referencia usuales en navegación</i>	<i>7</i>
<i>1.2.3. Errores del Sistema de Navegación Inercial (INS).....</i>	<i>8</i>
1.3. Sistema de Posicionamiento Global (GPS).....	8
<i>1.3.1. Segmentos del GPS.</i>	<i>9</i>
<i>1.3.2. Segmento Espacial</i>	<i>9</i>
<i>1.3.3. Segmento de Control.....</i>	<i>11</i>
<i>1.3.4. Segmento del Usuario</i>	<i>12</i>
<i>1.3.5. Errores GPS</i>	<i>13</i>
1.4. GPS / INS	13
<i>1.4.1. Arquitecturas de integración</i>	<i>14</i>
1.5. Fusión sensorial.....	14
1.6. Sensores.....	15
<i>1.6.1. Clasificación.....</i>	<i>15</i>
<i>1.6.2. Características.....</i>	<i>16</i>
<i>1.6.2.1. Características estáticas.....</i>	<i>16</i>
<i>1.6.2.2. Características dinámicas.....</i>	<i>17</i>

1.7.	Unidad de Madición Inercial (IMU).....	17
1.8.	Grados de libertad DOF	18
1.9.	Giroscopio	19
1.10.	Magnetómetro	20
1.11.	Acelerómetro	20
1.12.	Arduino	21
1.13.	El módulo GY-NEO6MV2	23
1.14.	Interfaz Hombre Máquina (HMI)	24
1.15.	Comunicación Inalámbrica con Arduino.....	24
1.16.	Filtro Kalman	25
1.16.1.	<i>Algoritmo del filtro de kalman.....</i>	26
1.17.	Software y Equipos de Monitoreo	28
1.18.	Diferencial de GPS	29
1.18.1	<i>Arquitectura DGPS</i>	29
1.18.1.1	<i>Precisión y limitaciones de DGPS.....</i>	30
CAPÍTULO II		
2.	MARCO METODOLÓGICO.....	31
2.1.	Selección de Componentes.....	31
2.1.1.	<i>Selección del microcontrolador</i>	32
2.1.1.1.	<i>Arduino MEGA</i>	33
2.1.2.	<i>Selección módulo GPS.....</i>	35
2.1.2.1.	<i>Gps neo 6m v2.....</i>	36
2.1.3.	<i>Selección del sensor inercial.....</i>	39
2.1.3.1.	<i>IMU MPU-9250</i>	41
2.2.	Integración GPS – IMU	47
2.2.1.	<i>Configuración del acelerómetro, giroscopio y magnetómetro.....</i>	54
2.2.2.	<i>Calibración de la IMU</i>	55
2.2.2.1.	<i>Requisitos de instalación</i>	56
2.2.2.2.	<i>Configuraciones de codificación.....</i>	56
2.2.2.3.	<i>Ejecución de la aplicación de calibración.....</i>	57
2.2.3.	<i>Programación del Arduino Mega 2560.....</i>	60
2.2.4.	<i>Aplicación de interpretación de datos</i>	63
2.2.4.1.	<i>Filtro de Kalman.....</i>	65
2.2.4.2.	<i>Algoritmo del filtro de kalman</i>	65

2.3.	Interfaz Gráfica Ljlab	69
2.4.	Propuesta de Mejora del Dispositivo - Soporte de Conexión Inalámbrica	70
2.3.1	<i>Codificación Arduino IDE</i>	71
2.3.2.	<i>Configuración de la aplicación en Processing</i>	75
2.3.3.	<i>Configuración de los módulos Xbee Serie 1</i>	78
CAPITULO III		
3.	Resultados	80
3.1.	Precisión y Fusión Sensorial	80
3.2.	Filtros (MADGWICK Y KALMAN)	81
3.3.1.	<i>Arquitectura de acoplamiento</i>	83
3.3.	Validación de Resultados Mediante Métodos Estadísticos	83
3.4.	Recorrido a través de la Epoch	84
3.5.	Comparación de Resultados Mediante Diseño Experimental	84
3.6.	Ejecución del Experimento	88
3.7.	Interpretación de Resultados	94
3.8.	Ventajas de la Metodología Propuesta	103
3.9.	Aplicaciones	105
CONCLUSIONES		106
RECOMENDACIONES		107
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

	Páginas
Tabla 1-1. Generación de satélites.....	10
Tabla 2-1. Clasificación de los sensores.....	16
Tabla 1-2. Comparación de las placas Arduino.....	33
Tabla 2-2. Comparación Módulos GPS.....	35
Tabla 3-2. Comparación sensores inerciales.....	39
Tabla 4-2. Comparación dispositivos inalámbricos.....	54
Tabla 1-3. Coordenadas de referencia de la ruta.....	86
Tabla 2-3. Coordenadas medidas.....	88
Tabla 3-3. Error absoluto de posicionamiento en centímetros.....	90
Tabla 4-3. Error medio en centímetros del factor principal y factor de bloque.....	91
Tabla 5-3. Estructura ANOVA para DBCA.....	92
Tabla 6-3. Totales por tratamiento y por bloque.....	92
Tabla 7-3. ANOVA.....	94
Tabla 8-3. Comparación de precios.....	104

ÍNDICE DE FIGURAS

	Páginas
Figura 1-1. Formación satelital.	10
Figura 2-1. Estaciones de segmento de control.....	11
Figura 3-1. Equipos GPS.....	12
Figura 4-1. INS con mediciones de carácter interoceptivos fusionadas con mediciones exteroceptivas.....	15
Figura 5-1. Unidad de medición inercial MPU 9150.....	18
Figura 6-1. Giroscopio.....	19
Figura 7-1. Acelerómetro.....	21
Figura 8-1. Placa Arduino UNO.....	23
Figura 9-1. Módulo GY-NEO6MV2.....	23
Figura 10-1. Interfaz hombre máquina.....	24
Figura 11-1. Conexión módulo XBee con Arduino mediante Shield.....	25
Figura 1-2. Metodología para el desarrollo del prototipo.....	31
Figura 2-2. Arduino MEGA.....	31
Figura 3-2. Módulo GPS.....	34
Figura 4-2. Conexión Arduino UNO – Módulo NEO 6M.....	37
Figura 5-2. Coordenadas obtenidas con NEO6M.....	37
Figura 6-2. Coordenadas obtenidas con NEO6M, Google Maps ESPOCH.....	38
Figura 7-2. Conexión MPU 9250.....	38
Figura 8-2. Ejes MPU 9250.....	41
Figura 9-2. Ejes acelerómetro MPU 9250.....	42
Figura 10-2. Ejes giroscopio MPU 9250.....	42
Figura 11-2. Reposo acelerómetro MPU-9250.....	43
Figura 12-2. Movimiento acelerómetro MPU-9250.....	43
Figura 13-2. Reposo giroscopio MPU-9250.....	44
Figura 14-2. Movimiento acelerómetro MPU-9250.....	44
Figura 15-2. Ejes del magnetómetro AK8963 MPU-9250.....	45
Figura 16-2. Ejes magnetómetro MPU 9250.....	45
Figura 17-2. Reposo magnetómetro MPU-9250.....	46
Figura 18-2. Movimiento magnetómetro MPU-9250.....	46
Figura 19-2. Conexión dispositivos.....	47
Figura 20-2. Selección de IMU.....	49
Figura 21-2. Obtención de la declinación magnética.....	50
Figura 22-2. Datos de sensores con filtro Madgwick Gradient Descent.....	51
Figura 23-2. Datos de sensores con filtro Madgwick implementation of Mahoney DCM.....	51

Figura 24-2. Datos de sensores con filtro DCM Implementation.....	52
Figura 25-2. Datos de sensores con filtro AHRS de Sebastián Madgwick.	52
Figura 26-2. Datos de sensores con filtro AHRS de Sebastián Madgwick.....	53
Figura 27-2. Aplicación de calibración en Python.....	58
Figura 28-2. Orden de movimientos para calibración.....	59
Figura 29-2. Gráficas de calibración.....	60
Figura 30-2. Gráficas de interfaz.....	68
Figura 31-2. Interfaz LJLAB.....	70
Figura 32-2. XCTU Configuración de canal e ID de comunicación.....	78
Figura 33-2. XCTU Configuración del protocolo de comunicación.....	78
Figura 34-2. Gráficas de calibración.....	78
Figura 35-2. Energy Shield2.....	79
Figura 36-2. Propuesta de dispositivo.....	79
Figura 1-3. Tight coupling modificado.	83
Figura 2-3. Trayectoria y coordenadas de puntos evaluados dentro de la ESPOCH.....	84
Figura 3-3. Dispersión de puntos LJLAB.....	102
Figura 4-3. Dispersión de puntos en trayectoria LJLAB.....	103

ÍNDICE DE GRÁFICOS

Gráfico 1-3. Dispersión de puntos GPS (40 m/min)	96
Gráfico 2-3. Dispersión de puntos GPS (160 m/min)	96
Gráfico 3-3. Dispersión de puntos GPS (280 m/min)	97
Gráfico 4-3. Dispersión de puntos GPS+IMU (40 m/min).....	97
Gráfico 5-3. Dispersión de puntos GPS+IMU (160 m/min).....	98
Gráfico 6-3. Dispersión de puntos GPS+IMU (280 m/min).....	98
Gráfico 7-3. Comparación de dispersión de puntos (GPS, GPS+IMU 40 m/min).....	99
Gráfico 8-3. Comparación de dispersión de puntos (GPS, GPS+IMU 160 m/min).....	99
Gráfico 9-3. Comparación de dispersión de puntos (GPS, GPS+IMU 280 m/min).....	100
Gráfico 10-3. Dispersión de puntos (GPS, GPS+IMU).....	100
Gráfico 11-3. Dispersión de puntos en trayectoria (GPS, GPS + IMU 40 m/min).....	101
Gráfico 12-3. Dispersión de puntos en trayectoria (GPS, GPS + IMU 160 m/min).....	101
Gráfico 13-3. Dispersión de puntos en trayectoria (GPS, GPS + IMU 280 m/min).....	102

ÍNDICE DE ECUACIONES

	Páginas
Ecuación 1-1. Ecuación principal del algoritmo de Kalman.....	26
Ecuación 2-1. Ecuación de la dinámica de las variables de estado.....	27
Ecuación 3-1. Ecuación de predicción.....	27
Ecuación 4-1. Ecuación de predicción.....	27
Ecuación 5-1. Ecuación de corrección.....	28
Ecuación 1-2. Ecuación principal del algoritmo de Kalman.....	65
Ecuación 2-2. Ecuación de la dinámica de las variables de estado.....	66
Ecuación 3-2. Ecuación de predicción.....	66
Ecuación 4-2. Ecuación de predicción.....	67
Ecuación 5-2. Ecuación de corrección.	67
Ecuación 6-2 Fórmula de obtención de ángulos de posición.....	69
Ecuación 1-3. Distancia entre dos puntos geográficos.....	91
Ecuación 2-3. Suma de cuadrados totales.....	93
Ecuación 3-3. Suma de cuadrados del tratamiento.....	93
Ecuación 4-3. Suma de cuadrados del bloque.	93
Ecuación 5-3. Suma de cuadrados del error.....	94

ÍNDICE DE ANEXOS

ANEXO A. Programación módulo NEO 6M

ANEXO B. Programación acelerómetro MPU 9250

ANEXO C. Programación giroscopio MPU 9250

ANEXO D. Programación magnetómetro MPU 9250

LISTA DE ABREVIATURAS

INS	Sistema de navegación inercial
GPS	Sistema de posicionamiento global
IMU	Unidad de medición inercial
MPU	Unidad de proceso múltiple
DOF	Grados de libertad
DC	Corriente Continua
V	Voltaje
I	Corriente
SC	Sumatoria de cuadrados
Bd	Baudios
HMI	Interfaz hombre máquina

RESUMEN

En el trabajo de investigación se implementó un sistema de navegación inercial, para mejorar la precisión de posicionamiento de un prototipo GPS en una trayectoria dentro de la Escuela Superior Politécnica de Chimborazo (ESPOCH), se utilizó el sistema de navegación inercial (INS) y sistema de posicionamiento global (GPS), para realizar el acoplamiento aplicando conocimientos científicos en el desarrollo de tecnologías de navegación y posicionamiento. Para el desarrollo del proyecto se calibraron los dispositivos en Python y su interfaz desarrollada en Processing, además se usó la arquitectura Tight Coupling en la cual se utilizó el Filtro de KALMAN para el acoplamiento de estos dos sistemas; en la cual se obtuvo las mejoras necesarias en el posicionamiento, además se modificó la arquitectura añadiendo el filtro de MADGWICK estabilizando de manera más rápida y precisa la unidad de medición inercial (IMU). Se desarrolló el dispositivo que fusiona las señales GPS e IMU y el software LJLAB en el cual se mostró las mediciones mejoradas (GPS/IMU) con sus respectivas gráficas. Para el análisis se usó el método experimental ANOVA y se logró demostrar una reducción en el error de posicionamiento en un 93% aproximadamente en comparación con el error que presentaba el dispositivo GPS por sí solo. Adicionalmente la propuesta representa un ahorro del 94% en cuanto a costo de adquisición, en comparación con los dispositivos actualmente presentes en el mercado. Se recomienda utilizar nuevas arquitecturas de acoplamiento de fusión sensorial con los dispositivos modernos que cada vez poseen mayor tecnología, lo cual reducirá más el error existente en las mediciones.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TELECOMUNICACIONES>, <SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)>, <UNIDAD DE MEDICIÓN INERCIAL (IMU)>, <SISTEMA DE NAVEGACIÓN INERCIAL>, <FUSIÓN SENSORIAL>, <SENSOR>, <MONITOREO EN TIEMPO REAL > <INTERFAZ HOMBRE MÁQUINA (HMI)>.

SUMMARY

This research implemented an inertial navigation system to improve the positioning precision of a GPS prototype in a path inside Escuela Superior Politécnica de Chimborazo (ESPOCH), using the Inertial Navigation System (INS) and Global Positioning System (GPS), to develop the coupling applying scientific knowledge in the development of navigation and positioning technologies. For the progress of the project the devices were calibrated in Python and its interface growth in Processing, also the Tight Coupling structural design was used in which the KALMAN Filter was used for the coupling of these two systems; in which the necessary improvements in positioning were obtained, and the structural design was modified by adding the MADGWICK Filter to the Inertial Measurement Unit (IMU). The device that fuses the GPS and IMU signals and the LJLAB software was developed in which the improved measurements (GPS / IMU) were shown with their respective graphs. For the analysis, the experimental method ANOVA was used and it was possible to demonstrate a reduction in the positioning error approximately 93% when compared to the error presented by the GPS device alone. Additional, the proposal represents a 94% about acquisition cost, compared to the devices currently on the market. It is recommended to use new sensory fusion coupling structural design with modern devices that increasingly have more technology, which will further reduce the error in measurements.

KEYWORDS: <ENGINEERING TECHNOLOGY AND SCIENCES>, <TELECOMMUNICATIONS>, <GLOBAL POSITIONING SYSTEM (GPS)>, <INERTIAL MEASUREMENT UNIT (IMU)>, <INERTIAL NAVIGATION SYSTEM>, <SENSORIAL FUSION>, <SENSOR>, <REAL TIME MONITORING>, <HUMAN MACHINE INTERFACE (HMI)>.

INTRODUCCIÓN

A pesar de los avances tecnológicos respecto a los Sistemas de Posicionamiento Global se presentan dos desventajas principales que ocurren al trabajar continuamente con dispositivos GPS (*Global Positioning System, por sus siglas en inglés*) que son los bloqueos de la señal debido a obstrucciones y la baja potencia de las señales. La combinación del GPS con un sistema autónomo de navegación inercial (INS) proporciona una solución ideal, que no sólo puede abordar la debilidad del sistema, sino también el error INS (*Inertial Navigation System, por sus siglas en inglés*) que asciende con el tiempo cuando funciona por sí solo. El sistema integrado puede proporcionar una solución de posición y velocidad incluso durante una interrupción del GPS, aunque sea durante un período limitado.

Con el presente trabajo de titulación se pretende dar una opción tecnológica en cuanto a los sistemas de posicionamiento, lo cual se consigue mediante la implementación de un sistema de navegación inercial INS en un prototipo GPS, con ayuda de fusión sensorial entre los sensores del sistema de navegación inercial, acelerómetro giroscopio y magnetómetro y el GPS, con lo cual se mejorará su precisión a un costo relativamente bajo en comparación con otros dispositivos sofisticados que cumplen con esta función pero a un precio de adquisición superior. El diseño de algoritmos también es una parte fundamental del proyecto que permitirán cumplir con el objetivo de integrar los dos dispositivos en uno solo.

La arquitectura que se emplea es una configuración de un sistema integrado GPS / INS en tiempo real, los datos se introducen con el filtro en donde se estima los errores INS que luego se usan para corregir la solución INS. La solución corregida se envía a un ordenador de campo en el que se ejecuta un sistema de información geográfica. La solución puede ser trazada en un mapa o visualizada en la plataforma GIS. Los resultados obtenidos se comprobarán mediante ensayos de prueba en una trayectoria dentro de la ESPOCH para hacer un análisis de resultados comparando el método inicial con el propuesto.

ANTECEDENTES

En las últimas décadas se han realizado diversas investigaciones en el campo de la radionavegación con el propósito de mejorar la utilidad del Sistema de Posicionamiento Global (GPS), los sistemas de navegación modernos procuran integrar dos o más tecnologías que les permitan obtener datos con mayor precisión en relación al posicionamiento de un cuerpo. El GPS

por si solo es capaz de proporcionar información exacta y continua sobre la posición para corregir estos errores se considera implementar un sistema de navegación inercial (INS) el cual proporciona una solución a la navegación en todo momento basada en la determinación de los cambios de velocidad y cambios de posición angular utilizando como instrumento una IMU (*Inertial Measurement Unit, por sus siglas en inglés*) la cual se conforma por sensores inerciales como giroscopios, acelerómetros y magnetómetros tridimensional, pero debido a la dependencia que existe con la constelación satelital, la precisión y la disponibilidad de este sistema se ve afectada por varios tipos de errores, algunos de estos errores son independientes del receptor GPS causados generalmente por las condiciones atmosféricas o errores de órbita, como también se presentan varios errores que dependen del entorno local, por ejemplo la interferencia en zonas urbanas en donde la señal de los satélites es susceptible, por lo que el funcionamiento autónomo de este sistema sigue siendo muy limitado.

Esta técnica de navegación se basa en el seguimiento de la posición anterior, la velocidad y el uso de los cambios medidos para actualizar sus valores, así en cualquier momento la solución de posicionamiento depende de las mediciones anteriores.

La incorporación del sistema de navegación inercial en conjunto con el sistema de posicionamiento global ha progresado ampliamente, de tal manera que a continuación se mencionan varias investigaciones relacionadas con el presente trabajo de titulación, las mismas que tienen campo de aplicación en el sector aeronáutico con aplicaciones para el control de vehículos no tripulados, además del interés en el campo de la radionavegación lo que ha permitido desarrollar métodos y técnicas con el propósito de mejorar la fiabilidad de los equipos receptores GPS.

El trabajo denominado “*GPS/INS Combination for a Beam Tracking System*” realizado por Sheng Zhang en la Universidad de Gavle en septiembre del año 2011, tiene como objetivo utilizar un algoritmo para mejorar la inexactitud del GPS en las carreteras, el estudio se divide en dos partes, la primera cual describe dos tipos de algoritmos que servirán de filtro para combinar los sistemas en una sola estimación de tiempo, velocidad y posición, para posteriormente decir cuál es el más apropiado para el proyecto. La segunda parte consiste en cargar el algoritmo seleccionado en la tarjeta ArduPilotMega siendo programado en el software arduino y a su vez realizar pruebas con el sistema GPS / INS integrado para estimar el rendimiento del mismo. Como conclusiones del trabajo el algoritmo de DCM no fue adecuado, por lo tanto, el método de Euler

Angle es más eficaz, los resultados de la simulación muestran que mediante la implementación del algoritmo permite mejorar la capacidad de seguimiento real del sistema.

El proyecto denominado “*Diseño e implementación de un sistema de navegación inercial tipo strapdown para estimar la posición de un robot móvil, aplicable a un prototipo de autopiloto de UAV*” realizado por Jorge David Guallichico y Carlos Adrián Utreras en la Escuela Politécnica Nacional en el año 2013, se desarrolla un prototipo con la finalidad de medir la actitud y el desplazamiento de un robot móvil para aplicarlo en un vehículo auto pilotado. El primer capítulo consiste en explicar los conceptos básicos que son necesarios para entender el contexto del proyecto, resaltando conceptos como sensores inerciales, filtro de Kalman etc., en el segundo capítulo se describe los lineamientos para diseñar y construir el hardware empleado en donde se hace uso de un sensor LSM303DLM, un sensor 6dof digital, un microcontrolador MBED, un módulo LS20031 y dispositivo XBEE PRO. El capítulo tres consiste en el desarrollo del sistema de navegación inercial explicando la programación para el microcontrolador utilizando el software LABVIEW. El cuarto capítulo presenta la interfaz que permite controlar el prototipo elaborado. El último capítulo contiene información acerca de las pruebas realizadas con el equipo, además de exponer los inconvenientes y beneficios obtenidos. Como resultado se obtuvo un INS con filtro de kalman extendido capaz de competir con los sistemas comerciales.

El trabajo de fin de grado en Ingeniería Aeroespacial realizado por Manuel Bravo Escudero con el título “*Aplicación electrónica para UAV: integración de IMU y GPS*” en la Universidad de Sevilla en el año 2015, tiene como objetivo estudiar las características de los dispositivos que forman parte del prototipo UAV expuesto por el autor, inicialmente se realiza un estudio de mercado de los sistemas UAV para ilustrar los equipos con mayor acogida en la localidad, además describe los elementos que conforman la unidad de medida inercial la cual posee diez grados de libertad, es decir, seis grados por parte del acelerómetro y giroscopio L3DG20H, tres grados por el magnetómetro LSM303DLHC y un grado por el sensor de temperatura y presión BMP180, además se realiza una descripción de las librerías de arduino utilizadas para los sensores. Respecto al receptor GPS el autor selecciona el modelo ultimate Breakout, que de manera similar describe los códigos necesarios para su funcionamiento. Como resultado se obtiene la integración de los dispositivos mediante el uso de una placa Arduino Due.

El estudio realizado por Muhammad Khalaf en junio del año 2016 denominado “**Optimization of an Integrated GPS/INS Navigation System for Optimal Control of UAV Horizontal Motion**” tiene como objetivo integrar sensores inerciales al sistema GPS para mejorar la navegación de un

vehículo no tripulado desarrollando una interfaz en el software Labview, utilizando dispositivos electrónicos como la IMU MPU 6050 y el GPS NEO-6M V2, los resultados de la simulación y verificación experimental de los algoritmos desarrollados confirman la eficacia del método propuesto para obtener una buena estimación UAV (*Unmanned Aerial Vehicle, por sus siglas en inglés*) incluso cuando los ruidos de los sensores aumentan.

FORMULACIÓN DEL PROBLEMA

¿Podrá la implementación de un sistema de navegación inercial, mejorar la precisión de posicionamiento de un GPS en la navegación a través de una trayectoria?

JUSTIFICACIÓN TEÓRICA

La aplicación de conocimientos científicos en el desarrollo de alternativas tecnológicas es de gran importancia en la actualidad, mediante el presente proyecto se busca desarrollar una nueva alternativa en cuanto a sistemas de navegación se refiere, con la aplicación de ciencias como el análisis matemático, programación y electrónica principalmente, lo cual hará un aporte favorable al repositorio investigativo de la ESPOCH en pro del desarrollo de conocimientos tecnológicos. En tal virtud con la aplicación de varias áreas de la ciencia se consigue generar nuevos conocimientos que con seguridad serán de utilidad para la formación de los estudiantes de las distintas instituciones educativas e investigativas, al ser un tema no explorado hasta la actualidad en nuestro país.

JUSTIFICACIÓN APLICATIVA

Con el desarrollo del proyecto se logra proveer de una alternativa en cuanto a sistemas de navegación se refiere, a pesar de que con el desarrollo tecnológico ya existen dispositivos sumamente sofisticados en cuanto a posicionamiento geográfico, su elevado costo es un limitante en cuanto a su adquisición, por tal motivo la implementación del proyecto se justifica al proporcionar un dispositivo que pueda cumplir con la misma funcionalidad con precisión, eficiencia y eficacia al menor costo posible.

OBJETIVOS

Objetivo general

- Implementar un sistema de navegación inercial, para mejorar la precisión de posicionamiento de un prototipo GPS en una trayectoria dentro de la ESPOCH.

Objetivos específicos

- Analizar el sistema de posicionamiento global GPS e investigar el funcionamiento, arquitectura y sistemas de referencia de INS
- Seleccionar los componentes del prototipo a implementar que cumplan con características de funcionalidad requeridas y bajo costo de adquisición.
- Diseñar y construir el prototipo GPS para acoplarlo con los sistemas de navegación inercial.
- Examinar las mediciones individuales de cada sistema para sus respectivas interpretaciones
- Validar el funcionamiento del sistema mediante pruebas de campo, en una trayectoria dentro de la ESPOCH.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Navegación

La navegación se define como el método que permite determinar la posición y el rumbo de un cuerpo en movimiento con respecto a una referencia conocida a través de la geometría, la astronomía o las señales de radio además permite planificar un curso de un lugar a otro para evitar y prevenir obstáculos o colisiones.

1.2. Navegación inercial

La navegación inercial basa sus fundamentos en los principios de la cinemática que partiendo de un punto de posición inicial logra calcular posiciones futuras en todo momento, siempre y cuando conozca la velocidad, orientación y aceleración relativas.

Entre las ventajas de la navegación inercial se tiene que no necesita de estaciones externas para el cálculo del posicionamiento, tal como ocurre con los GPS, los cuales hacen uso de satélites. La precisión de este sistema también es de destacar debido a que la información producida hace uso de una tasa muy alta de muestreo. La navegación inercial es de funcionamiento continuo, una solución de navegación de alto ancho de banda (por lo menos 50 Hz), un bajo ruido a corto plazo y la provisión de medidas de actitud, angular y de aceleración, así como la posición y la velocidad.

También posee ciertas desventajas, las cuales radican en que necesita datos iniciales de información como posición y orientación. Después de la etapa de toma de mediciones iniciales procede al cálculo de las posiciones mediante la información de velocidad y aceleración tanto lineal como angular, información extraída por sensores, no obstante, esta toma continua de mediciones produce errores que se van acumulando conforme avanza el tiempo. Otros inconvenientes son la degradación en la precisión de la navegación con el tiempo y el coste.

La precisión de este tipo de navegación se degrada con el tiempo, debido a los errores acumulativos causados por offsets, biases, factores de escala y no linealidades presentes en los sensores inerciales. (Grewal.M, 2007,p.2)

1.2.1. Sistemas de coordenadas.

En los sistemas de navegación inercial es común ver la aplicación de un MDT que por sus siglas significa modelo digital del terreno, sin embargo, para poder establecer las coordenadas del MDT es necesario hacerlo con respecto a un punto de referencia, es decir un sistema de coordenadas.

La selección del sistema de coordenadas es de suma importancia debido a que cada componente del sistema sensorial tiene su forma propia de proporcionar la información de posición.

Hay gran variedad de sistemas de coordenadas como los ortogonales, dextrógiros y cartesianos y las diferencias entre ellos radican en el punto de referencia que adoptan como origen, la orientación de cada uno de sus ejes y finalmente en el desplazamiento entre cada uno de sus planos. La selección de ellos dependerá del movimiento en el que se enfoque el estudio.

1.2.2. Sistemas de referencia usuales en navegación

Con la finalidad de comprender los sistemas de navegación es importante conocer acerca de los principales sistemas de referencia, los cuales a continuación se los menciona: Sistema inercial geocéntrico ECI (*Earth Centered Inertial, por sus siglas en inglés*), Sistema de Ejes Tierra ECEF (*Earth Centered, Earth Fixed, por sus siglas en inglés*) y Sistema de ejes horizonte local LLS (*Local Level System, por sus siglas en inglés*).

ECI es un sistema de referencia que se encuentra nominalmente ubicado en el centro de masa terrestre y con orientación respecto al eje de giro de la Tierra. Los ejes X y Y se encuentran dentro del plano ecuatorial, estos ejes no giran con la Tierra, el eje Y siempre se encuentra a noventa grados por delante del eje x en la dirección de rotación. El eje Z siempre apunta a lo largo del eje de rotación de la Tierra desde el centro hacia el polo norte,

ECEF tiene su punto de origen en el centro del elipsoide que modela la superficie de la Tierra, el eje X apunta desde el centro hasta la intersección del ecuador con el meridiano de referencia o el

meridiano cero convencional, que define 0 grados de longitud. El eje Y completa el conjunto ortogonal derecho, apuntando desde el centro hasta la intersección del ecuador con el meridiano este de noventa grados. El eje Z siempre apunta a lo largo del eje de rotación de la Tierra desde el centro hasta el Polo Norte.

LLS es el sistema de referencial en donde el punto de origen es el centro de masa del cuerpo usuario o vehículo. El eje X es la proyección en el plano ortogonal al eje Z de la línea desde el usuario hasta el polo norte., el eje Y siempre apunta hacia el este, el eje z, se encuentra perpendicular a la superficie del elipsoide de referencia apuntando aproximadamente hacia el centro de la Tierra.

1.2.3. Errores del Sistema de Navegación Inercial (INS)

Por lo general se presentan tres tipos de errores cuando se refiere a sistemas de navegación inercial, estos son errores en las medidas de acelerómetro y giroscopio, los errores de inicialización y las limitaciones de la velocidad de iteración en las ecuaciones de navegación. El error de inicialización de velocidad y error producido por las ecuaciones de navegación integran los sesgos de acelerómetro y giroscopio los cuales puede producir errores respecto a la posición, las limitaciones de ecuaciones de navegación y ruido de giro tienen un efecto acumulativo en los errores de la solución de navegación.

1.3. Sistema de Posicionamiento Global (GPS)

El Sistema de Posicionamiento Global es un sistema de radio navegación desarrollado por el departamento de defensa de los Estados Unidos de América que en sus inicios fue utilizado con fines militares, pero en la actualidad se ha convertido en un servicio gratuito de uso civil a razón de sus múltiples beneficios. Este sistema concede al usuario información específica acerca del posicionamiento, navegación y cronometría en un punto determinado.

En cuanto a la característica de posicionamiento, hace referencia a la ubicación de un objeto en el globo en donde el sistema cumple la función de determinar las coordenadas espaciales como longitud, latitud y altitud de un punto respecto a un sistema de referencia global, teniendo la capacidad de localizar la posición del objeto que posea un receptor GPS en cualquier lugar de la superficie de la Tierra sin importar las condiciones climáticas durante las 24 horas. Respecto a la característica de navegación el sistema permite dar seguimiento a los objetos estáticos o en

movimiento en el campo terrestre, aéreo y marítimo proporcionando datos relevantes relacionados a la trayectoria de desplazamiento precisando un punto de origen y un punto de destino. Por último, la característica de cronometría se refiere a la capacidad de anticipar y definir el tiempo aproximado en recorrer una trayectoria de principio a fin.

La posición y velocidad de un vehículo puede ser obtenida mediante un GPS, el cual requiere de una constelación de satélites en órbita para su funcionamiento. Esta técnica de navegación permite obtener mediciones aproximadamente cada 01 segundo y error limitado de 03 a 05 metros para uso civil”. (P. Aggarwal, 2010)

1.3.1. Segmentos del GPS.

El Sistema de Posicionamiento Global (GPS) se divide en tres segmentos los cuales se mencionan a continuación. En primer lugar, el segmento espacial el cual está compuesto por una red de satélites que orbitan el globo terrestre, los mismos que están distribuidos estratégicamente para lograr una cobertura a nivel mundial y emitir señales de radio hacia toda la superficie, el segundo es el segmento de control el cual se refiere a las estaciones terrestres que monitorizan y registran las señales recibidas por los satélites y por último el segmento del usuario el cual hace mención a los equipos que tienen la capacidad de recibir la señal que envían los satélites.

1.3.2. Segmento Espacial

Este segmento se conforma por un grupo de satélites que envían señales de radio desde el espacio denominados constelación NAVSTAR en donde el Sistema de Posicionamiento Global tiene el compromiso de mantener en operación al menos veinte y cuatro satélites, la Fuerza Aérea en los últimos años ha conservado en órbita siete satélites adicionales los cuales sirven de ayuda para mejorar el rendimiento de la constelación principal.

Como características de la constelación principal se puede mencionar que los veinte y cuatro satélites operan normalmente a 20.200 Kilómetros de altitud, los mismos que se distribuyen uniformemente en cada uno de los seis planos orbitales, además se considera una inclinación respecto al plano ecuatorial de cincuenta y cinco grados, cada satélite rota la tierra dos veces por día teniendo un tiempo de sidéreo de 12 horas. Los satélites GPS transfieren dos tipos de señales de radio las cuales son de baja potencia, denominadas señales L1 y L2, ambas señales se componen de tres parámetros que brindan información importante acerca del estado de los

satélites al segmento de control, estos parámetros son los datos de efemérides de satélite, códigos pseudoaleatorios y los datos de almanaque.

El primer parámetro son los datos de efemérides del satélite que permiten conocer la posición de todos los equipos espaciales en el plano orbital a cualquier instante. El código pseudoaleatorio identifica cual fue el satélite que transmite la señal. Y por último los datos de almanaque cuya información predice el estado de los satélites además de entregar la fecha y hora actual mediante el reloj atómico incorporado en el satélite.

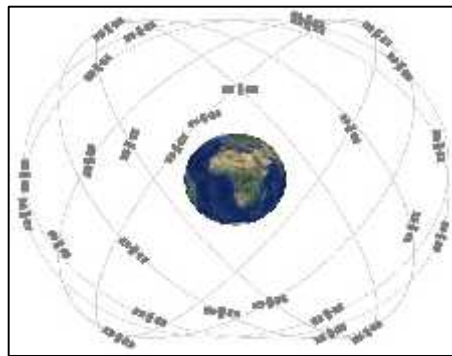


Figura 1-1. Formación satelital
 Fuente: (GPS.gov, 2017), url: <https://goo.gl/Dd514h>

La constelación principal del GPS actualmente opera satélites antiguos y nuevos, el reemplazo de los equipos espaciales se debe principalmente a la vida útil de los satélites, que causas como el agotamiento de combustible, las averías en el sistema eléctrico, el daño en los paneles solares son las más usuales, además los avances tecnológicos respecto a su diseño han permitido mejorar el estado de los satélites. En el transcurso de los años han existido varias generaciones de satélites, en un inicio se consideraba un promedio general cercano a los siete años y medio, pero hoy en día con los avances tecnológicos asciende a los quince años de vida útil. A continuación, se resume las generaciones de satélites en la siguiente tabla.

Tabla 1-1. Generación de satélites

Satélites	Estado	Fecha de lanzamiento	Número de satélites
Block I	-	1978 – 1985	10
Block II	-	1985 – 1990	9
Block II A	0 operativa	1990 – 1997	19

Block II R	12 operativa	1997 – 2004	12
Block II R-M	7 operativa	2005 – 2009	8
Block II F	12 operativa	2010 – 2016	12
Block III	En producción	Lanzamiento 2017	Previsto 32

Fuente: (GPS.gov, 2017), url: <https://goo.gl/Dd514h>

1.3.3. Segmento de Control

El segmento de control del GPS es una red global que integra a todas las instalaciones, estaciones y equipos distribuidos en la superficie terrestre los cuales se encargan de monitorear los satélites del segmento espacial, permanentemente supervisan el estado de los equipos y las señales que transmiten los satélites.

Las estaciones de control incorporan modelos los cuales determinan y predicen las órbitas calculando los datos de ajuste y las posiciones de los satélites (efemérides), además sincronizan y corrigen los relojes con el tiempo GPS. Las estaciones del segmento de control se encuentran localizadas estratégicamente cerca del plano ecuatorial, este segmento consta de una estación principal ubicada en Colorado Springs (U.S.A.), una estación de control maestro alternativa que se encuentra en California (U.S.A.), dieciséis estaciones de monitoreo y once antenas de mando. La siguiente figura identifica la ubicación de las principales instalaciones.



Figura 2-1. Estaciones de segmento de control

Fuente: (GPS.gov, 2017), url: <https://goo.gl/Dd514h>

La estación principal de control que se encuentra ubicada en el estado de Colorado en la base de la Fuerza Aérea Schriever en este lugar se calcula los datos y parámetros de las órbitas de cada uno de los satélites y los relojes de las demás estaciones transmitiendo esta información a las antenas terrestres que son receptoras y emisoras de ondas de radio por el medio denominado banda S.

Las estaciones de monitoreo tienen la función de identificar las distancias de cada uno de los satélites y transmitir esta información junto con los datos meteorológicos a la estación principal de control, las estaciones de monitoreo cuentan con receptores GPS que trabajan con dos frecuencias L1/L2 además de un reloj de cesio que poseen muy alta precisión.

1.3.4. Segmento del Usuario

Este segmento es el conjunto de equipos receptores GPS, software y tecnología que son utilizados para registrar las señales emitidas por los satélites y que calculan los datos de posición en base a la velocidad de la luz y el tiempo que tarda en recibir la señal.



Figura 3-1. Equipos GPS

Fuente: (Brandão, 2016), url: <https://goo.gl/XaXMWw>

El receptor calcula las coordenadas en los planos X, Y, Z además de proporcionar el tiempo actual. Para el cálculo de los parámetros de longitud, latitud, altitud y obtención de la hora se requiere como mínimo de cuatro satélites visibles. Estos equipos integran una antena que les permite captar la señal de radio además de un receptor con la capacidad de dar seguimiento, visualización, procesamiento, registro y presentación de los datos.

1.3.5. Errores GPS

Generalmente los sistemas de posicionamiento global están sujetos principalmente a dos tipos de error: en relación con el tiempo y al ruido es decir los errores del reloj satelital y los errores de propagación de la ionosfera y la troposfera los mismos que pueden ser parcialmente corregidos por el equipo receptor gps que el usuario utilice. Los errores antes de la aplicación de correcciones se conocen como errores crudos y los que quedan después del proceso de corrección se conocen como errores residuales. Los errores de seguimiento no se pueden corregir, sólo se suavizan. Los errores debidos a la trayectoria múltiple junto con el error de predicción de efemérides afectan a la solución de navegación a través del cálculo de la posición y velocidad del satélite.

1.4. GPS / INS

El sistema de posicionamiento global (GPS) tiene como ventaja el proporcionar una alta precisión de posición a largo plazo con errores limitados a unos pocos metros, la velocidad de salida es baja, normalmente alrededor de 10 Hz. Las señales también están sujetas a obstrucción e interferencia, por lo que no se puede confiar en los receptores GPS para obtener una solución de navegación continua, en cambio el sistema de navegación inercial proporciona salida de alto ancho de banda en al menos 50 Hz y exhibe bajo ruido a corto plazo. También proporciona medidas efectivas de la velocidad angular y la aceleración, así como la posición y la velocidad. Sin embargo, la precisión de una solución de navegación inercial se degrada con el tiempo a medida que los errores de los instrumentos de inercia se integran a través de las ecuaciones de navegación.

Los beneficios del sistema de navegación inercial (INS) y del sistema de posicionamiento global (GPS) se complementan, por lo que, al integrarlos, las ventajas de ambas tecnologías se combinan para ofrecer una solución de navegación completa continua, de alto ancho de banda, con una alta precisión a largo y corto plazo. La integración es adecuada para aplicaciones de navegación inercial establecidas, tales como barcos o aviones de pasajeros además otro uso práctico de la navegación inercial es la aplicación de sensores inerciales de bajo nivel táctico, convirtiendo así a la integración INS/GPS en una solución de navegación adecuada para aviones ligeros, helicópteros, vehículos aéreos no tripulados.

1.4.1. Arquitecturas de integración

La arquitectura de un sistema de navegación integrado INS/ GPS varía en tres aspectos principales que son: primero el cómo se aplican las correcciones a la solución de navegación inercial, segundo que clase de equipo GPS se utiliza para obtener las mediciones y tercero la integración del equipo receptor GPS con el INS además de aplicar un algoritmo de fusión. Estos son en gran medida independientes entre sí, los términos tales como bajo acoplamiento, estrechamente acoplado, acoplado ultraestrecho se utilizan para definir las arquitecturas de integración antes mencionadas.

1.5. Fusión sensorial

Es importante tener en cuenta que cualquier algoritmo de fusión de datos solo tendrá un funcionamiento eficaz en el caso de cumplirse ciertas circunstancias consistentes dependiendo de las hipótesis simplificadoras en las cuales esté basado su diseño. En tal virtud la selección del algoritmo adecuado conforme a las necesidades es crucial en la calidad de los resultados posteriormente obtenidos.

Cada algoritmo hace uso de la estadística en donde se establece un grado de dependencia o correlación, y necesita conocer por completo las características y funcionalidades de cada sensor, por este motivo realizar una evaluación correcta de cada sensor en cuanto a parametrización y performance es de gran utilidad para obtener estimaciones confiables. Con el avance continuo de la tecnología, se han podido establecer algoritmos cada vez más eficaces y confiables, esto ha permitido que se pueda unir distintas clases de sensores con el fin de reducir el error acumulativo de las mediciones de los mismos, tales clases son: los sensores interoceptivos que no necesitan de estaciones externas como acelerómetros, giroscopios, magnetómetros, etc. los sensores exteroceptivos como GPS, radar, cámara entre otros, que a pesar de proporcionar información de manera directa con errores reducidos, los instantes con los que los proveen son discretos y retardados lo cual es una gran desventaja para usos de navegación en tiempo real.

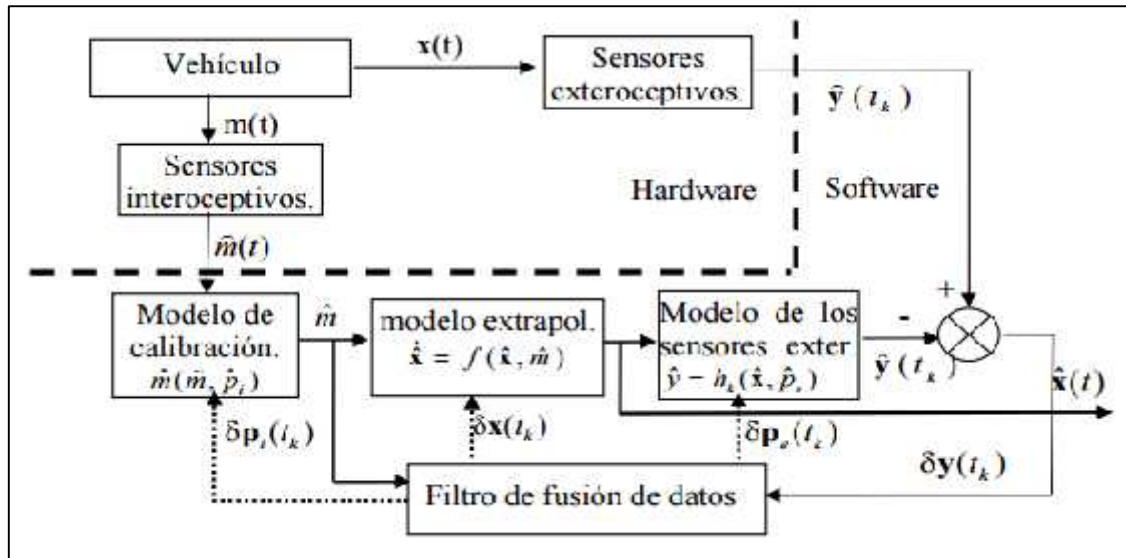


Figura 4-1. INS con mediciones de carácter interoceptivos fusionadas con mediciones exteroceptivas.

Fuente: (España, 2012), p.22, url: <https://goo.gl/CWYCXZ>

1.6. Sensores

Un sensor es un objeto o dispositivo que posee la capacidad de identificar magnitudes tanto físicas como químicas y transformarlas en variables medibles, es decir que el propósito de un sensor es responder a una propiedad de entrada y convertirla en una señal eléctrica cuantificable que sea compatible con circuitos electrónicos.

Existen dos tipos de sensores: digitales y analógicos, en donde el voltaje de salida del sensor digital varía en un rango comprendido entre uno y cero, en cambio el sensor analógico puede emitir cualquier valor entre sus rangos de voltaje, su salida de tensión cambia según la lectura del sensor.

1.6.1. Clasificación.

Además, se tiene varias formas para clasificar a los sensores, pero las clasificaciones más comunes son por el principio de transducción y por el tipo de variable medida.

Tabla 2-1. Clasificación de los sensores

Por el principio de transducción	Según la variable física a medir
Químico	De posición, velocidad y aceleración
Magnético	De nivel y proximidad
Fotoeléctrico	De humedad y temperatura
Capacitivo	De fuerza y deformación
Piezoresistivo	De flujo y presión
Piezoeléctrico	De color, luz y visión
Termoeléctrico	De gas y pH
Ultrasónico	Biométricos
-	De corriente

Fuente: (Corona Ramírez, 2014), p.18

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

1.6.2. Características

Todos los sensores poseen características particulares que lo distinguen en comparación con otros tipos, las características de los sensores pueden agruparse en dos bloques: estáticas y dinámicas. Las características estáticas hacen referencia a todos los parámetros que no cambian a través del tiempo o se presentan cambios demasiado lentos en la variable a medir, mientras que las características dinámicas describen las propiedades del sensor en función del tiempo, su comportamiento es transitorio.

1.6.2.1. Características estáticas

Sensitividad. Esta característica se refiere a la magnitud de entrada mínima que se necesita para emitir una señal de salida.

Rango. El rango es el intervalo que existe entre el valor mínimo y el valor máximo que puede leer el sensor.

Precisión. Esta característica se refiere al grado de repetitividad, es decir, la capacidad que tiene el sensor para emitir la misma señal de salida por más de una vez.

Exactitud. Se define como la diferencia máxima que existe entre el valor de salida emitido por el sensor y el valor real medido.

1.6.2.2. Características dinámicas

Tiempo de respuesta. Esta característica se refiere al tiempo que tarda el sensor en entregar el valor de la magnitud medida.

Histéresis. Es la capacidad que tiene el sensor para seguir la curva de salida ideal.

1.7. Unidad de Medición Inercial (IMU)

Los sensores inerciales pueden usarse para reconocer el movimiento de un objeto en tiempo real dado la posición inicial y la orientación de un cuerpo. La técnica o el proceso utilizado se conocen como seguimiento inercial. Una Unidad de Medición Inercial es un dispositivo que generalmente tiene acelerómetros y giroscopios los cuales permiten medir aceleraciones lineales y velocidades angulares respectivamente.

La mayoría de las IMU hoy en día incorporan magnetómetros para ayudar en la orientación. Estas magnitudes físicas pueden ser integradas en el tiempo para obtener una estimación de las posiciones y orientaciones de un cuerpo, pero esto requiere el desarrollo de algoritmos apropiados de fusión de sensores.

Un IMU puede integrar a varios sensores para sus mediciones, giróscopo para la detección de cambios en la rotación, acelerómetro para la aceleración, e incluso un magnetómetro que permita mejorar la precisión de los datos mediante una comparación con los puntos cardinales. En la actualidad el control y procesamiento de estos sensores se lleva a cabo mediante microcontroladores capaces de disminuir el error producido e incluso realizar la denominada “fusión sensorial”, lo cual se consigue mediante la ejecución de variedad de algoritmos. Con esta fusión de mediciones es posible mejorar la precisión del sistema debido al aumento en las fuentes disponibles de información.

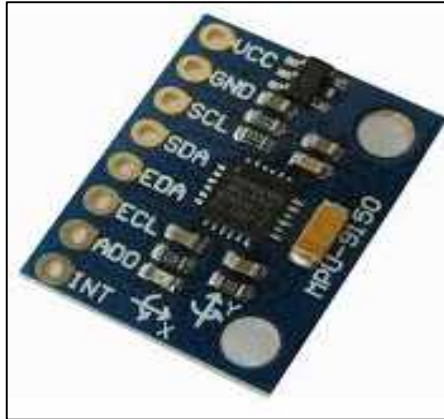


Figura 5-1. Unidad de medición inercial MPU 9150

Fuente: (Juarez, 2015), url: <https://goo.gl/Dy1D6j>

1.8. Grados de libertad (DoF)

Los grados de libertad son uno de los conceptos más importantes de la mecánica. Este concepto es ampliamente utilizado en robótica y cinemática. D.O.F significa cuántas variables son necesarias para determinar la posición de un mecanismo en el espacio.

Seis grados de libertad (6DoF) se refiere a la libertad de movimiento de un cuerpo rígido en el espacio tridimensional, específicamente, el cuerpo es libre de cambiar de posición como avance / retroceso (levantamiento), arriba / abajo (impulso), izquierda / derecha (balanceo) de la traslación en tres ejes perpendiculares, combinado con los cambios de orientación a través de la rotación de tres ejes perpendiculares, Yaw (eje normal), Pitch (eje lateral) y Roll (eje longitudinal).

Una unidad de medición inercial (IMU), es un dispositivo electrónico que mide la velocidad, la orientación y las fuerzas gravitatorias, utilizando una combinación de sensores (acelerómetros, giroscopios y magnetómetros). Hay numerosas referencias a 9 DOF' IMU, pero esto es bastante confuso ya que sólo hay un total de 6 DOF, básicamente muchas empresas que fabrican IMU han introducido el término DOF 9 con el fin de vender sus productos, pero estos no se refieren grados de libertad en la vida real.

El 9 DOF proviene de la adición de DOF que cada tipo de sensor contengan dentro de la IMU. Así que, si el IMU tiene un acelerómetro que es capaz de detectar 3 DOF, un giroscopio que puede detectar 3DOF y un magnetómetro que puede detectar 3DOF se lo llamaría un 9DOF IMU.

1.9. Giroscopio

Es un dispositivo que utiliza la gravedad de la Tierra para ayudar a determinar la orientación. Los giroscopios se utilizan para detectar el giro angular o la velocidad angular alrededor de algún eje. Los giroscopios han evolucionado desde los dispositivos de hilatura mecánicas inercial que consta de rotores, ejes y cardanes a varias encarnaciones de dispositivos electrónicos y ópticos. Cada uno explota alguna propiedad física del sistema que le permite detectar la velocidad de rotación alrededor de algunos ejes.

Los giroscopios convencionales hacen uso de las propiedades inerciales de una rueda que gira a alta velocidad, la rueda giratoria tiende a mantener la dirección de su eje de giro en el espacio y define la dirección de referencia, la inercia giroscópica es fundamental para el funcionamiento de todos los giroscopios de masa elástica define la dirección fija en el espacio que permanece fija en el marco de referencia inercial.



Figura 6-1. Giroscopio

Fuente: (Cienciamatik, 2017), url: <https://goo.gl/8uuLej>

1.10. Magnetómetro

El magnetómetro es un instrumento de medición, se encarga de expresar cuantitativamente el nivel de campo magnético que detecta, no obstante para aplicaciones de navegación inercial, el campo magnético que interesa detectar es el de la magnetósfera (campo magnético del planeta). Debido a que con ello es posible determinar la ubicación de los puntos cardinales, en este tipo de aplicaciones también se conoce al instrumento con el nombre de geo-magnetómetro.

La inserción del magnetómetro de 3 ejes en las IMU ha permitido mejorar la precisión en la detección de las variaciones de ángulos, a pesar de que funcionan a una baja frecuencia, en fusión con los giroscopios, se consiguen resultados más precisos y confiables.

Actualmente los magnetómetros utilizan tecnología anisotrópica magnetorresesiva que ofrecen una precisión en la sensibilidad y linealidad en el eje. Están diseñados para la medición de dirección y magnitud de los campos magnéticos del planeta en mili gauss.

1.11. Acelerómetro

Un acelerómetro es un dispositivo o sensor electromagnético que tiene la función de medir los cambios en las fuerzas de aceleración tanto estáticas como dinámicas. Sabiendo que por fuerza estática se refiere a la gravedad y por dinámica a la producida por vibraciones o cualquier tipo de movimientos.

Dependiendo de los grados de libertad del acelerómetro, es posible detectar la aceleración y medirla en uno, dos o hasta tres ejes.

Generalmente los acelerómetros están constituidos por placas capacitivas tanto fijas como móviles. Las placas móviles se conectan internamente a resortes de un tamaño minúsculo y su movimiento se produce conforme a las fuerzas resultantes del movimiento de todo el dispositivo. Por el cambio de distancia entre placas que se provoca por el movimiento, se genera un cambio en la capacitancia, mismo que es medido para relacionarlo así con una medida de aceleración.

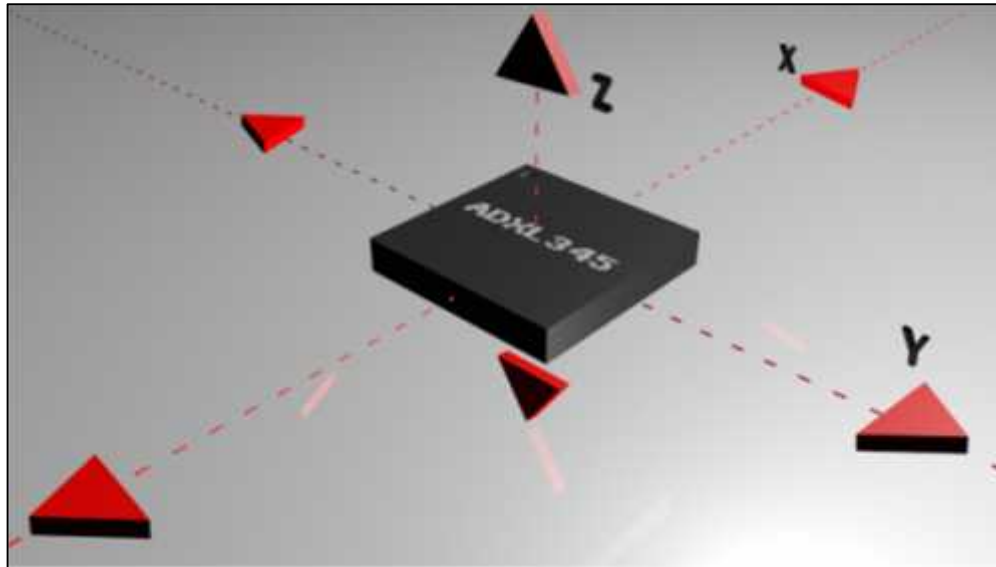


Figura 7-1. Acelerómetro

Fuente: (5Hertz, 2014), url: <https://goo.gl/Lj7x1Y>

Otra variedad de acelerómetros, están compuestos por materiales de tipo piezoeléctricos, mismos que al someterse al efecto de una tensión mecánica, generan una pequeña carga eléctrica, la cual es medida y relacionada con un valor de aceleración.

Por lo general en los acelerómetros es posible seleccionar el rango de las fuerzas que se deberán medir. Dependiendo de la sensibilidad del acelerómetro seleccionado, se podrá medir rangos desde $\pm 1g$ hasta $\pm 250g$

1.12. Arduino

Es una plataforma de dispositivos electrónicos, las placas de Arduino poseen microcontroladores capaces de efectuar un sin número de tareas conforme a la programación que se les imponga, misma que es de carácter abierto (open-source). La programación de las placas de Arduino se efectúa por medio de (Arduino Programming Language) y la licencia de su software es open-source, por lo que se tiene total libertad tanto en el diseño del software como del hardware, debido a que se cuenta con placas pre ensambladas como placas que se pueden ensamblar a mano con respecto a las necesidades del usuario.

Arduino posee la capacidad de comunicarse con una gran cantidad de dispositivos y accesorios compatibles, estos pueden ser sensores de todo tipo, motores, monitores etc.

Abriendo el abanico de posibilidades en cuanto a aplicaciones, a un precio relativamente bajo, las placas Arduino de módulos para ensamblaje a mano como los preensamblados, se encuentran en el mercado a menos de 50 dólares americanos.

Otra ventaja de Arduino es su compatibilidad con los distintos sistemas operativos, ya sea Windows, Macintosh OSX o /Linux, a diferencia de otros microcontroladores que por lo general poseen compatibilidad de ejecución únicamente con la plataforma de Windows.

Una característica de Arduino que le da mucha ventaja en comparación con otros microcontroladores es la disponibilidad existente de una librería en donde se almacena infinidad de programaciones desarrolladas por su comunidad que a su vez son descargables gratuitamente.

Arduino es una plataforma electrónica de código abierto basado en hardware y software fácil de usar. Placas Arduino son capaces de leer las entradas - la luz en un sensor, un dedo sobre un botón, o un mensaje de Twitter - y convertirla en una salida - la activación de un motor, encender un LED, publicar algo en línea. Se puede decir que su tablero qué hacer mediante el envío de un conjunto de instrucciones al microcontrolador en el tablero. Para ello se utiliza el lenguaje de programación de Arduino (basado en el cableado), y el software de Arduino (IDE) , sobre la base de procesamiento . (Arduino, 2017)

Arduino IDE es un entorno de programación que permite al usuario redactar diferentes tipos de programas y cargarlos en el microcontrolador Arduino. Arduino utiliza un lenguaje de programación fácil de usar, que se basa en un lenguaje de programación llamado Processing. Después de que el usuario ha escrito su código, IDE compila y traduce el código al lenguaje ensamblador. Después de traducir el código, el IDE sube el programa al microcontrolador Arduino. Arduino IDE tiene un analizador de código incorporado que comprueba al usuario.

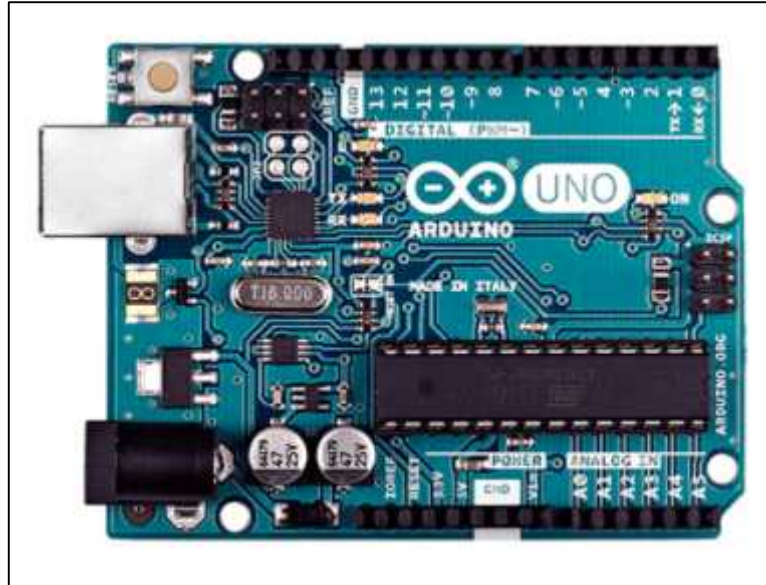


Figura 8-1. Placa Arduino UNO

Fuente: (ARDUINO, 2017), url: <https://goo.gl/kcxJiD>

1.13. El módulo GY-NEO 6MV2

Entrega información como efemérides, almanaque, última posición y tiempo y estado del satélite y una señal opcional de sincronización de tiempo reducirán el tiempo para fijar primero significativamente y mejorar la adquisición sensibilidad. Todos los módulos NEO-6 soportan los servicios on-line y AssistNow Offline A-GPS de u-blox 11 y son compatibles con OMA SUPL.



Figura 9-1. Módulo GY-NEO6MV2

Fuente: (Electronics, 2014), url: <https://goo.gl/krMGh9>

1.14. Interfaz Hombre Máquina (HMI)

Un HMI es una aplicación de software que presenta la información a un operador o usuario sobre el estado de un proceso, y para aceptar y poner en práctica las instrucciones de control de los operadores. Por lo general la información se muestra en un formato gráfico (interfaz gráfica de usuario o GUI). Un operador se encuentra a menudo una parte de un (Control de Supervisión y Adquisición de Datos) sistema SCADA.



Figura 10-1. Interfaz hombre máquina

Fuente: (Schneider Electric, 2017), url: <https://goo.gl/KqHa5x>

1.15. Comunicación inalámbrica con Arduino

Arduino posee una interminable cantidad de aplicaciones, pudiendo así cumplir con distintas necesidades del usuario, sin embargo para el desarrollo de muchas aplicaciones, es necesaria la comunicación y el intercambio de datos de forma inalámbrica. En tal virtud, existen varias alternativas y la selección de una de ellas se hará en función de las necesidades y después de evaluar sus pros y contras.

Una de las opciones de comunicación inalámbrica es mediante wifi, existe muchas formas de llevarlo a cabo, ya sea mediante Shield, integrado en MCU, placas con wifi integrado o Arduino Yun. Facilitando con ello el acceso de cualquier placa Arduino a internet.

También se dispone de la posibilidad de realizar la comunicación ya sea entre placas Arduino o con otros dispositivos como smartphone y computadoras mediante conexión bluetooth, para lo

cual se dispone del dispositivo compatible para la conexión con la placa, sin embargo se debe tener en cuenta que su principal desventaja radica en que la conectividad bluetooth es para distancias relativamente pequeñas.

Otra forma de comunicación disponible es mediante radio frecuencia, por lo cual existen módulos compatibles con Arduino como el nRF24 que permite velocidades de hasta 2Mb/s.

Finalmente, Zigbee/Xbee provee la posibilidad de establecer una comunicación inalámbrica por medio de redes de carácter personal. Xbee es el nombre comercial de un grupo de módulos que se encuentran basados en el estándar zigbee que hace uso de la banda ISM para aplicaciones industriales, científicas y médicas. Para el uso o configuración de un módulo Xbee, hay a disposición las Shield correspondientes para la conexión del puerto serie de Xbee con Arduino que generalmente contienen un chip FTDI que cumple la función de hacer de pasarela.

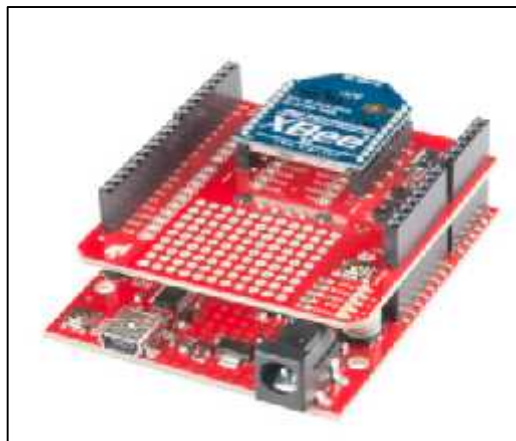


Figura 11-1. Conexión módulo XBee con Arduino mediante Shield

Fuente: (Crespo, 2016), url: <https://goo.gl/6hPw88>

1.16. Filtro Kalman

El filtro de Kalman es un algoritmo que posee gran variedad de aplicaciones, con el uso de este método es posible estimar variables de estado de carácter “no observable”, partiendo de variables de carácter “observable” que a su vez podrían tener un error en su medición.

El algoritmo que compone el filtro de Kalman está formado por dos tipos de ecuaciones, las ecuaciones principales y las ecuaciones de estado. Las principales se ocupan de relacionar las

variables de estado con las observables mientras que las ecuaciones de estado cumplen la función de establecer las variables de estado estructuralmente.

En tal virtud, el filtro de Kalman es utilizado tanto para la corrección de errores de medición como para la predicción de estados. En el sistema de navegación inercial, se encarga de corregir el error acumulativo proporcionado por los sensores de navegación, disminuyendo a su vez el ruido de sus mediciones, además mediante su algoritmo se encarga de la denominada “Fusión sensorial”, debido a que recibe la información de los sensores para poder emitir una sola salida corregida.

El filtro de Kalman hace uso de una dimensión temporal que consiste en el análisis de la dinámica de las variables de estado y también utiliza una dimensión transversal en donde obtiene las mediciones en intervalos de tiempo de las variables observables, ésta dinámica es ejecutada en 2 etapas, la primera es la etapa de predicción y luego sigue la etapa de corrección.

Es decir, en la primera etapa hace la primera estimación de la variable de estado y a continuación se encarga de mejorarla utilizando datos de las observables, tiene la característica de que las futuras predicciones consideran las correcciones anteriores, lo cual se conoce como “extracción de señales”.

1.16.1. Algoritmo del filtro de kalman

Como primer punto, el algoritmo del filtro de Kalman, se compone de la ecuación principal:

Ecuación 1-1. Ecuación principal del algoritmo de Kalman

$$\mathbf{Z}_t = \mathbf{H}_t * \mathbf{x}_t + \mathbf{u}_t$$

Donde:

\mathbf{Z}_t = Vector de variables observables

\mathbf{x}_t = Vector de las variables de estado

\mathbf{H}_t = Matriz de relación, variables observables con variables de estado

\mathbf{u}_t = Error de medición de las variables observables.

El valor de u_t se adopta asumiendo que no se encuentra autocorrelacionado ni tampoco presenta correlacionado con las variables observables y además que posee una distribución normal con una media aritmética de cero.

La ecuación de estado por su lado se encarga de determinar la dinámica de las variables de estado.

Ecuación 2-1. Ecuación de la dinámica de las variables de estado

$$x_t = A_t * x_{t-1} + e_t$$

Donde:

e_t = Vector de errores

x_t = Vector de las variables de estado

e_t es incluido con el fin de hacer una modelación de la incertidumbre de variables de estado, el supuesto para este término es que sus datos poseen una distribución gaussiana con media de cero.

Las ecuaciones de predicción se rigen por:

Ecuación 3-1. Ecuación de predicción

$$x_{t+1,0} = A_t * x_{t,0}$$

Donde:

$x_{t,0}$ = Estimador óptimo del vector

Ecuación 4-1. Ecuación de predicción

$$P_{t+1}^F = A_t * P_t * A_t^T + Q$$

Donde:

P_t = Varianza del vector de estados

Tal como se mencionó anteriormente, los datos de las predicciones son mejorados tomando en cuenta una corrección que se define mediante la siguiente ecuación:

Ecuación 5-1. Ecuación de corrección

$$x_{t+1,0} = x_{t+1,0} + K_{t+1} * \frac{(z_{t+1} - z_{t+1}^p)}{e_{t+1}} = x_{t+1,0} + K_{t+1} * (z_{t+1} - H_{t+1} * x_{t+1,0})$$

Donde:

K_t = Ganancia de Kalman.

1.17. Software y equipos de monitoreo

Todo equipo o proceso automático requiere de una interfaz de monitoreo, la cual debe ser fácil de entender y amigable con el usuario, este elemento forma parte de la HMI (Interfaz hombre máquina).

La creación de la interfaz de monitoreo debe realizarse en función de las necesidades, ya sea que se cree un software desde cero o se realice la interfaz desde un software ya creado y especializado para esta labor, como por ejemplo: LabVIEW, Matlab, etc. Programas que permiten tanto el monitoreo como control en tiempo real de controladores y microcontroladores.

En lo que se refiere a sistemas de navegación, un software de monitoreo deberá cumplir como mínimo con la funcionalidad de proporcionar al usuario de información visual referente a las coordenadas de posicionamiento del IMU. Es decir que la unidad deberá mantener una conexión constante con el equipo de visualización, ya sea el monitor de un computador, de un Smartphone o una pantalla que como se mencionó anteriormente, forma parte de la interfaz hombre máquina.

1.18. Diferencial de GPS

El DGPS o GPS diferencial, este sistema provee a los receptores de GPS correcciones de los datos que son recibidos por los satélites, con el propósito de proveer una mayor precisión en la posición calculada. Este se fundamentó básicamente debido a la introducción a la disponibilidad selectiva. Esto se dio debido a que los errores que se originan en el sistema GPS afectan igualmente a los receptores cuyos sitios están colindantes entre sí. (JIMÉMEZ RODRÍGUEZ, y otros, 2010, p.13)

En tal razonamiento se puede hallar que los errores están potentemente correlacionados en los receptores adyacentes. La posición emitida por el sistema es recibida a través de un receptor cuya ubicación es conocida a través de otras técnicas, y de este modo está en la pre disponibilidad de calcular aquellos errores que se producen por el sistema GPS. El receptor en mención transmite la corrección de errores a aquellos receptores que se encuentra más cercanos a él y de esta manera los errores producidos por el sistema se corregirán a la par, dentro del área de cobertura de transmisión de señales del equipo GPS de referencia. (JIMÉMEZ RODRÍGUEZ, y otros, 2010, p.13)

1.18.1 Arquitectura DGPS

La estructura DGPS se describe de la siguiente manera:

1. Estación monitorizada, que conoce su posición con una precisión muy alta, esta estación está compuesta por: Un receptor GPS, un microprocesador que sirve para calcular los errores del sistema GPS y para generar la estructura del mensaje que se envía a los receptores, un transmisor, mismo que ayuda a establecer un enlace de datos unidireccional hacia los receptores de los usuarios finales. (JIMÉMEZ RODRÍGUEZ, y otros, 2010, p.13)
2. Equipo de usuario, compuesto por un receptor que posee GPS y un receptor del enlace de datos desde la base monitorizada. Existen varias formas de obtener las correcciones DGPS., las más usadas son: las que son recibidas a través de radio, mediante cierto canal preparado para ello, como el RDS que es una emisora de FM; y mediante las descargas de internet, o con una conexión inalámbrica. (JIMÉMEZ RODRÍGUEZ, y otros, 2010, p.13)

En Estados Unidos existe el WAAS, en Europa el EGNOS y en Japón el MSAS, todos compatibles entre sí. En los mensajes que se envían a los receptores próximos se pueden circunscribir dos tipos de correcciones:

1. La primera que está estrechamente ligada a una corrección que está claramente definida y aplicada a la posición, dicha descripción posee un inconveniente en cuanto a lo que tiene que ver con el usuario y a su vez con la estación de monitoreo ya que están expuestos a emplear los mismos satélites, ya que las correcciones a realizar se establecen en los mencionados satélites. (JIMÉMEZ RODRÍGUEZ, y otros, 2010, p.14)
2. En otro caso el usuario está en la posibilidad de realizar las correcciones de cuatro de los satélites que posan mejor relación señal – ruido, determinando una corrección que es aplicada a las pseudo-rangos de cada uno de los satélites visibles, provocando que este tipo de correcciones sean más flexibles. (JIMÉMEZ RODRÍGUEZ, y otros, 2010, p.14)

1.18.1.1 Precisión y limitaciones de DGPS

Las pruebas controladas y el uso operacional ampliamente extendido del DGPS, han demostrado en repetidas ocasiones que el sistema pseudo-rango resulta en una precisión del orden aproximadamente de 1 a 10 metros. Esta cifra es en gran medida independiente del tipo de receptor y a distancias de hasta 500 km de la estación de referencia, con los sistemas de Kinematic GPS (KGPS), los cuales requieren la resolución de las ambigüedades de fase de la portadora mientras el receptor se encuentra en movimiento, se puede lograr precisiones de centímetros. (WILDERMANN, y otros, 1992, p.45)

Las correcciones del DGPS son locales, por lo tanto, sus valores sólo son aplicables en áreas cercanas a la estación de referencia. La distancia máxima depende de la aplicación y la técnica de procesamiento de la señal por parte del receptor. En general, se hace necesario un estudio particular para determinar si los errores debidos a la distancia son aceptables y si se pueden seguir aplicando las correcciones. (WILDERMANN, y otros, 1992,p.45)

CAPÍTULO II

2. MARCO METODOLÓGICO

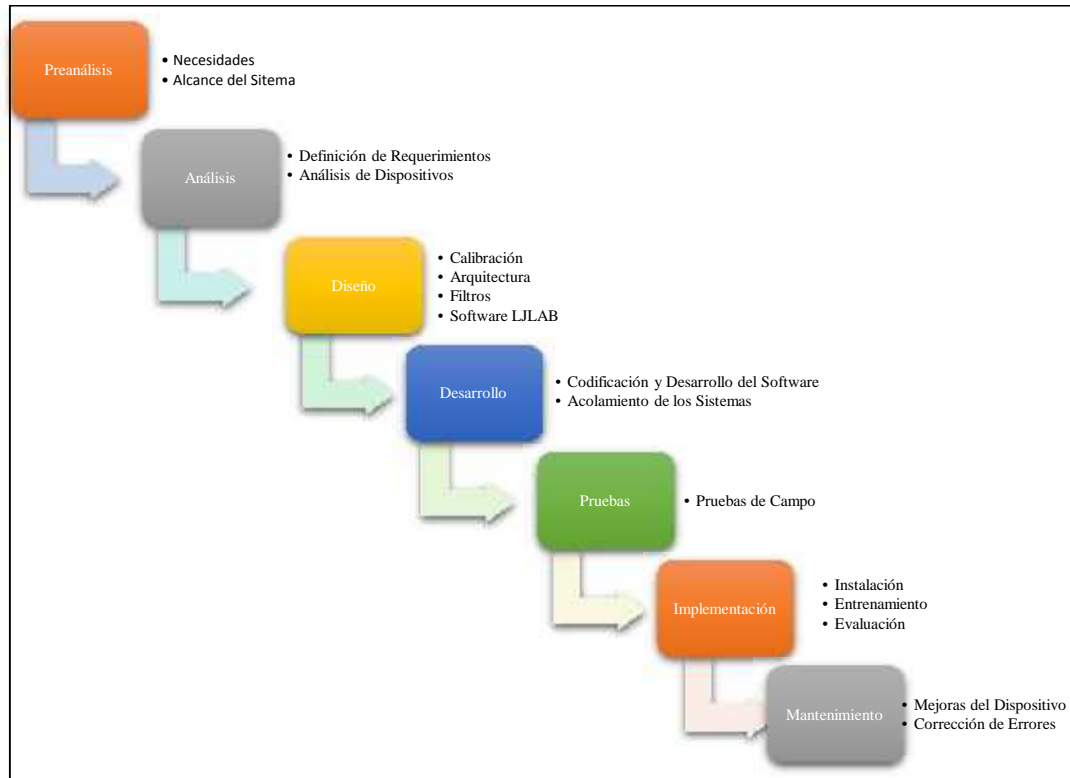


Figura 1-2. Metodología para el desarrollo del prototipo

Fuente: (Lumitex, 2017), url: <https://goo.gl/HzgoSY>

2.1. Selección de componentes

La idea principal de este trabajo es reducir el margen de error que comúnmente se presenta en los dispositivos GPS respecto a la precisión de posicionamiento en los parámetros de longitud y latitud que se obtienen con estos equipos, para cumplir este objetivo se propone incorporar sensores inerciales que mejoren dichas características, en primer lugar se realiza una investigación que permita identificar cuáles serán los componentes electrónicos más apropiados que cumplan los requerimientos del proyecto.

La investigación para seleccionar los dispositivos se divide en tres temas, cada uno de acuerdo al número de dispositivos que se consideran necesarios en la formación del prototipo, el primer

punto se refiere a la elección del micro controlador, para lo cual se ha elegido a la placa electrónica Arduino, con este equipo se realizan varias pruebas con la finalidad de controlar el funcionamiento de todos los componentes del prototipo, de esta manera la programación se desarrolla en base al lenguaje C a través de la plataforma Arduino IDE.



El segundo punto es seleccionar el módulo GPS, un aspecto importante en este tema es considerar que el dispositivo propuesto sea compatible con la plataforma Arduino, el mismo que debe poseer los terminales TX/RX para la comunicación serial entre el módulo y la placa electrónica, es importante primero confirmar que existan librerías disponibles que faciliten su programación, además el equipo debe contar con una antena que permita la recepción de datos del segmento espacial.

El tercer tema es la selección de los sensores que componen el sistema de navegación inercial, en esta fase se toma en cuenta diversas alternativas respecto al equipo a utilizar como aquellos módulos independientes que contienen acelerómetros y giroscopios, así como también dispositivos que combinan ambos tipos en una sola unidad, para decidir que sensor es el más útil se lo elige basándose en ciertas características como por ejemplo los grados de libertad, costos y disponibilidad en el mercado.

2.1.1. Selección del microcontrolador

En esta fase se elige a la placa Arduino como microcontrolador para el presente proyecto, recordando que en el mercado existen varios tipos de placas en donde para su adquisición se debe tener en cuenta las funciones que cumplen, por tal motivo la decisión para seleccionar con cual Arduino tenemos que trabajar va a depender de factores como la tensión de alimentación, el número de entradas y salidas además de la memoria interna que guarda la programación realizada en la plataforma IDE que a nuestra consideración es uno de los aspectos más importantes. Durante el proceso para elegir el mejor dispositivo siempre es recomendable comparar las características básicas de al menos dos equipos que cumplan similares funciones, de esta manera es posible decidir cuál es el más adecuado basándose en determinadas particularidades por tal motivo a continuación se presenta la siguiente tabla en donde se compara dos modelos de placas electrónicas que usualmente son utilizadas para proyectos de electrónica.

Tabla 1-2. Comparación de las placas Arduino

Arduino UNO	Arduino MEGA
	
Características	
<ul style="list-style-type: none"> - 68,6 mm x 53,4 mm - Microcontrolador ATmega328. - Voltaje de entrada 7-12V. - 14 pines digitales de I/O (6 salidas PWM). - 6 entradas análogas. - 32k de memoria Flash. - Reloj de 16MHz de velocidad. 	<ul style="list-style-type: none"> - 10,9 cm x 5,6 cm - Microcontrolador ATmega2560. - Voltaje de entrada de 7-12V. - 54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM). - 16 entradas análogas. - 256k de memoria flash. - Velocidad del reloj de 16Mhz.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Para realizar las pruebas del funcionamiento a los componentes del prototipo como lo son el sensor inercial y el módulo GPS se trabaja con la placa Arduino UNO la cual es apropiada para trabajar individualmente con cada uno de ellos, pero se debe especificar que para el prototipo es preferible trabajar con la placa Arduino MEGA 2560 porque permite conectar todos los módulos en conjunto sin limitación respecto a las entradas y salidas digitales.

2.1.1.1. Arduino MEGA

Como beneficios de este dispositivo se tiene principalmente su bajo costo porque son más accesibles en tiendas electrónicas comparando con otros microcontroladores, también la plataforma IDE no se limita solo en trabajar en windows sino también en otros sistemas operativos como Linux o MAC, su entorno de programación es simple, fácil de utilizar por los estudiantes ya que cuenta con una infinidad de librerías que vuelven posible el uso de sencillos códigos. Es de código abierto pues permite acceder a todo el circuito y modificar las librerías sin necesidad

de una licencia. Existen una gran ventaja que es la disposición de varios módulos externos que son compatibles con la plataforma como es el caso de los sensores. También existe suficiente información en internet al ser un dispositivo con software y hardware libre. Respecto a sus aplicaciones hoy en día la placa Arduino se ha convertido en uno de los microcontroladores más utilizados para realizar varios tipos de proyectos en áreas de ingeniería tales como son la automatización, robótica, CNC, impresión 3D, domótica, educación y por su puesto en electrónica y telecomunicaciones.

La placa Arduino MEGA 2560 posee características mejoradas en comparación con el Arduino UNO debido a que cuenta con un microcontrolador ATmega1280, además incluye cincuenta y cuatro terminales que trabajan como entradas y salidas digitales, también dispone de dieciséis entradas analógicas. En la figura 1-2 se puede observar que los pines de color amarillo son los terminales digitales, los de color celeste son los terminales para comunicación serial, y los pines de color rojo son los terminales analógicos.

La mayoría de los dispositivos requieren de los puertos TX/RX, es importante mencionar que esta placa electrónica cuenta con tres puertos serial, Serial 1 en los pines 19 RX - 18 TX, Serial 2 en los terminales 17 RX - 16 TX, por último, Serial 3 en los terminales 15 RX - 14 TX.

La ventaja de esta placa electrónica es poseer mayor capacidad en la memoria interna, lo cual permite almacenar la programación de todos los dispositivos juntos, esta es la razón principal por lo cual se ha seleccionado el Arduino MEGA como microcontrolador para el prototipo, también es un beneficio que posea el número de terminales exactos para energizar con el voltaje adecuado a los dispositivos que requieren entre tres y cinco voltios. Otro punto a favor es que todas las placas de Arduino tienen en común la plataforma de programación IDE esto ayuda al desarrollo del programa.

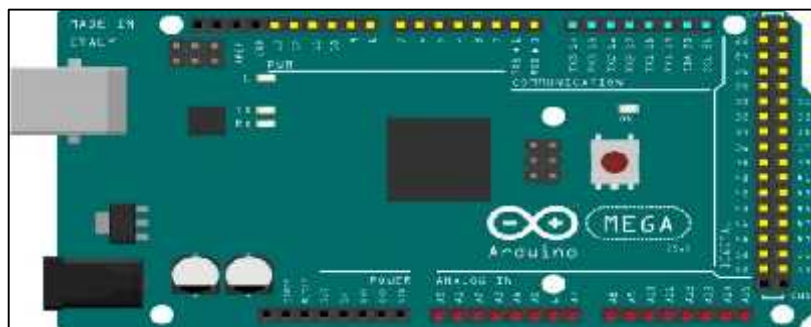


Figura 2-2. Arduino MEGA
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

2.1.2. Selección módulo GPS

En la selección de este equipo tenemos dos modelos disponibles que son compatibles con la plataforma Arduino, dependiendo de ciertas características se escoge el dispositivo que mejor se adapte a las necesidades del proyecto.

Tabla 2-2. Comparación Módulos GPS

Características	GPS SIM808	GY NEO 6M V2
Tipo de receptor	66 canales	50 canales
Tasa máxima de actualización	30 s	27 s
Sensibilidad	-165 dBm	-161 dBm
Precisión de posición horizontal	2.5 m	2.5 m
Antena	Si	Si

Tabla 2-2. (Continuación)

Características	GPS SIM808	GY NEO 6M V2
Voltaje de entrada	3,4 – 4,4 v	3,3 – 5 v
Interface	Serial	Serial UART 5V
EEPROM	-	Si
Soporta	GPRS	SBAS
Velocidad de baudios	19200 Bd	9600 Bd

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

El módulo GPS SIM808 tiene doble función pues permite la recepción de datos GPRS y el envío de información a través de GSM, este dispositivo es capaz de obtener las coordenadas y el horario UTC, además de enviar SMS y recibir llamadas. Para la lectura de datos este equipo requiere trabajar con la plataforma IDE de Arduino donde para obtener los parámetros antes mencionados

se debe utilizar comandos denominados AT. En cambio, el módulo GPS NEO 6M V2 posee un software libre para determinar las coordenadas sin necesidad de conectar el dispositivo a la placa Arduino, para lo cual es necesario tener un adaptador FTDI que permita reconocer el módulo en el ordenador. El programa se llama u-center versión 8.25, basta conectar el dispositivo y seleccionar el puerto COM al cual está conectado para que el módulo reciba los datos de los satélites y se observe la posición actual del dispositivo a través de la aplicación de Google Earth.

El módulo SIM808 requiere precisamente conectarse con Arduino, si no se tiene la placa y el equipo juntos, el usuario no podrá obtener la información que busca, como ventaja se puede mencionar que la programación del dispositivo es mucho más sencilla solo necesita una librería y pocas líneas de código, pero el inconveniente que se presenta es que no todos conocen los comandos AT y para que el dispositivo responda esto es primordial. Además, respecto al tiempo de duración que se demora para conectarse con los satélites es más prolongado en comparación con el NEO 6M. La precisión de posicionamiento de igual manera no es buena al tomar un punto de referencia cerca de la Facultad de Electrónica es evidente que el módulo SIM808 fijo un punto más alejado.

Respecto al módulo NEO 6M de igual manera el tiempo que dura en conectarse con los satélites no es instantáneo, pero se pudo notar que, tarda menos que el SIM808, este dispositivo cuenta con un software propio que le permite modificar su configuración. En cuanto se refiere a la programación con la plataforma Arduino IDE requiere de una librería adicional que es Tiny GPS esta última es una de las versiones más actualizadas para determinar los valores del módulo GPS, Adicionalmente se obtienen los valores en unidades UTM, con el GPS SIM808 se tiene grados, minutos y segundos. También fijándose en las dimensiones es más pequeño, de igual manera la antena no ocupa demasiado espacio esta es una característica primordial para el prototipo ya que se busca que sea lo más reducido posible. Entonces como conclusión se selecciona para la elaboración del prototipo el módulo GPS NEO 6M.

2.1.2.1. Gps NEO 6M V2

La principal función de este dispositivo es determinar las coordenadas de latitud y longitud en un punto para lo cual requiere de la plataforma Arduino IDE y una placa electrónica para su conexión. Este módulo está equipado en el PCB, además posee una EEPROM que sirve como memoria SD la misma que viene con configuración de fábrica. Incluye una pila pequeña para conservar los datos de configuración en la memoria EEPROM. También cuenta con una antena

cerámica. En cuanto a los terminales de conexión posee pines Vcc, Rx, Tx y Gnd por los cuales se conectan al microcontrolador mediante una interfaz serial. Para que el módulo GPS NEO 6M trabaje perfectamente se recomienda hacer las pruebas en un ambiente abierto para una correcta recepción de señal.

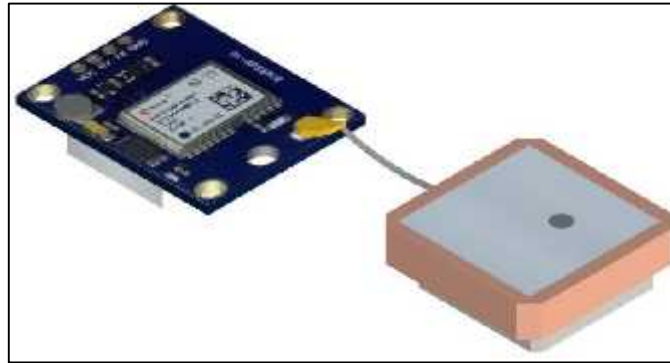


Figura 3-2. Módulo GPS
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Como se requiere obtener los parámetros de longitud y latitud de forma continua, es necesario programar el módulo GPS a través de la plataforma Arduino. Para las pruebas de funcionamiento se conecta directamente el dispositivo al Arduino UNO, percatándose de conectar correctamente como se muestra a continuación.

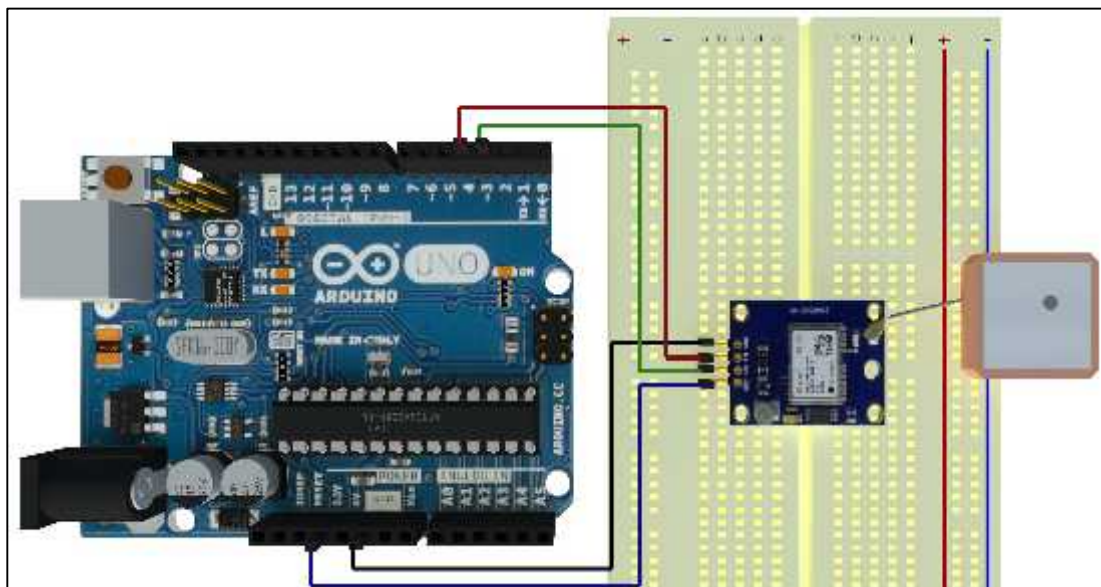


Figura 4-2. Conexión Arduino UNO – Módulo NEO 6M
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

En el anexo A se exponen las líneas de código donde es primordial incluir las librerías Software Serial y Tiny GPS, una vez cargada la programación se compila el programa verificando que los puertos COM coincidan como también la placa y el procesador. Al presionar Ctrl + Mayús + M se visualiza el monitor serial de esta forma se pueden registrar los parámetros obtenidos, principalmente la longitud y latitud.



Figura 5-2. Coordenadas obtenidas con NEO6M
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Listas las coordenadas se comprueba la ubicación ingresando a la web y escribiendo los datos de longitud y latitud.

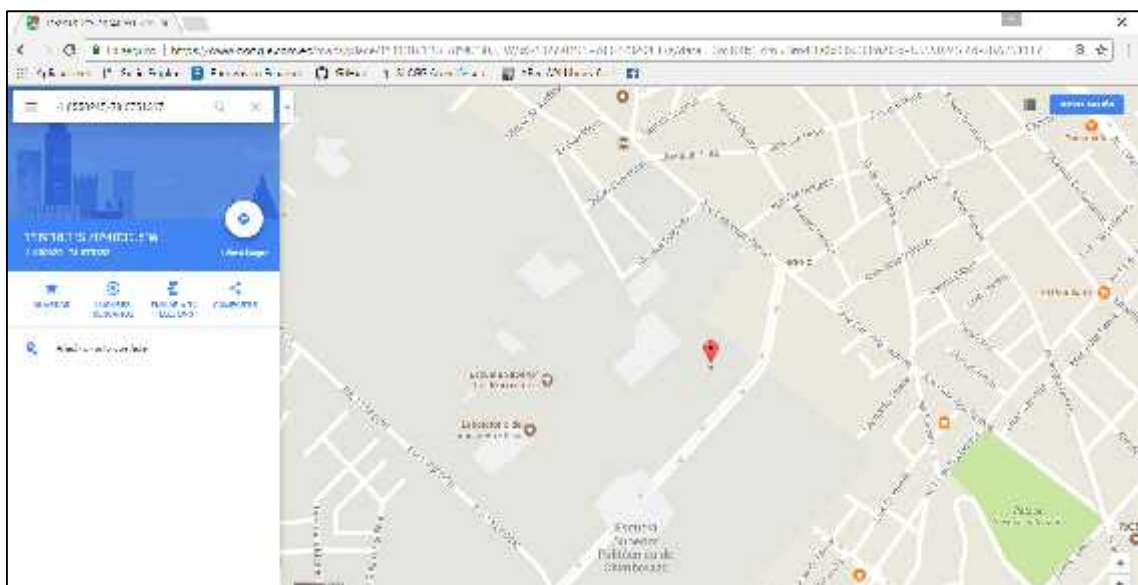


Figura 6-2. Coordenadas obtenidas con NEO6M, Google Maps ESPOCH
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

2.1.3. Selección del sensor inercial

De igual forma se estudian las ventajas que poseen los sensores inerciales con la finalidad de encontrar el dispositivo más conveniente que se ajuste a las necesidades del proyecto, por lo que la siguiente tabla contiene varias especificaciones de módulos electrónicos que han sido seleccionados por su bajo costo y cumplen con el principio de visualizar las señales de un acelerómetro, giroscopio y magnetómetro.

Tabla 3-2. Comparación sensores inerciales

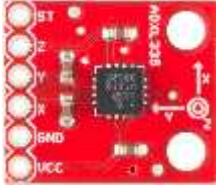



Dispositivo	Características
<p>ADXL345</p> 	<p>El acelerómetro ADXL345 detecta la aceleración en tres ejes X, Y, Z es decir con tres grados de libertad (3 DOF). El voltaje de entrada admite valores entre 2,1 a 3,6 voltios. Su precio económico es de siete dólares, que es relativamente barato en comparación a los sensores sofisticados que tienen un valor aproximado a cien dólares. A su vez tiene un puerto I2C y un bus SPU lo que permite obtener los datos medidos de una forma más sencilla. Para visualizar los tres ejes de aceleración se debe conectar el sensor a la placa Arduino, cargar la programación correspondiente en la plataforma IDE y observar los parámetros medidos en el monitor serial. Sus dimensiones son 3 mm × 5 mm × 1 mm.</p>

Tabla 3-2. (Continuación)

Dispositivo	Características
<p>HMC5883L</p> 	<p>El magnetómetro HMC5883L, con el cual se obtienen parámetros en relacionados a la orientación del dispositivo respecto al polo Norte, posee la capacidad de medir el campo magnético de la tierra permitiendo digitalizar la salida de los ejes X, Y y Z monitoreando de tal forma los componentes en cada uno de los planos cartesianos. En cuanto a la precisión del dispositivo tiene un rango de $\pm 2^\circ$ gauss debido a que integra tres chips de magneto resistencia. Como características se puede mencionar primero que la comunicación es simple y todo se realiza a través de la interfaz I2C, las dimensiones de la placa que contiene el chip son 14 x 15 mm resultando mucho más pequeño que los anteriores sensores. Su valor en el mercado es cercano a diez dólares considerado relativamente económico.</p>

<p>ITG 3200</p> 	<p>El giroscopio ITG-3200 como característica incorpora tres convertidores para digitalizar las salidas proporcionando así la velocidad angular en los tres ejes, soporta un voltaje de entrada en un rango de 2,1 a 3,6 voltios y la comunicación como en el anterior sensor se realiza a través de la interfaz I2C. Su precio es cercano a treinta dólares. Las dimensiones de la placa que contiene el chip son 17.78 x 13.97 mm por lo que es uno de los dispositivos más pequeños del mercado.</p>
<p>MPU 9250</p> 	<p>El sensor MPU-9250 es la unidad de medición inercial que permite obtener los parámetros respecto a la aceleración, velocidad angular y campo magnético. Este dispositivo se compone de dos partes principales, primero por el chip de la IMU 6500 que posee seis grados de libertad 6 DOF al contar con un acelerómetro y giroscopio integrado, segundo del magnetómetro AK863 de tres grados de libertad 3 DOF. La comunicación serial se lo puede realizar a través de la interfaz I2C como también por el bus SPI. A su vez cuenta con varios convertidores que digitalizan las señales ajustando a los dispositivos en rangos como al giroscopio entre ± 250 y ± 2000, al acelerómetro entre $\pm 2g$ y $\pm 16g$, y al magnetómetro hasta $\pm 4800\mu T$. El voltaje de alimentación es de cinco voltios debido a que posee un regulador de tensión. El valor de este sensor es alrededor de trece dólares.</p>

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

La selección del dispositivo depende de varios aspectos, considerando como el más importante la formación del sistema de navegación inercial para lo cual se tiene dos opciones, la primera es utilizar los tres dispositivos ADXL345, ITG-3200 y HMC5883L a la vez y la segunda es hacer uso de la IMU MPU-9250. Los inconvenientes al comparar ambas opciones son que para integrar el funcionamiento de los tres sensores se requiere de una programación distinta o de un filtro además se procura que el prototipo posea dimensiones lo más mínimas posibles es decir que los componentes ocupen el menor espacio físico disponible por lo que obviamente la primera opción no es la recomendable. En cuanto se refiere al valor de los dispositivos la suma de los tres se aproxima a cincuenta dólares, en cambio la IMU tiene un precio de trece dólares resultando la opción más económica.

De acuerdo con las especificaciones del fabricante la IMU disminuye el consumo de energía en un cuarenta y cuatro por ciento, asimismo el rendimiento de magnetómetro es cuatro veces más preciso y el giroscopio tres veces más superior a otras alternativas. Basándose en las características técnicas se tiene que la MPU-9250 proporciona mejores resultados respecto a otros dispositivos ya que al incorporar el magnetómetro AK8963 elimina la deriva que son errores con el giroscopio que se producen con las horas de uso, también reduce las interferencias como el ruido en el acelerómetro. Como defecto el sensor ADXL345 detecta cualquier aceleración que no sea de la gravedad pues al no girar el dispositivo detecta un cambio de rotación. La ventaja del

sensor MPU 9250 es que posee un procesador de movimiento digital DMP (*Data Management Platform, por sus siglas en inglés*) que fusiona los parámetros del giroscopio y del acelerómetro con el propósito de reducir los errores inherentes de cada uno. Es evidente que la mejor opción para trabajar es el sensor MPU-9250 por tal motivo se ha seleccionado este dispositivo como parte del prototipo.

2.1.3.1. IMU MPU-9250

El sistema de navegación inercial básicamente se compone por los ejes medidos con un acelerómetro y un giroscopio, en el mercado existen varios dispositivos que proporcionan estos parámetros al usuario, pero la mayoría son parte de sistemas sofisticados que tienen un elevado costo, para la elaboración del prototipo GPS se ha seleccionado como componente el sensor MPU-9250 debido a su fácil adquisición, su bajo costo y dimensiones apropiadas, como características esta unidad posee dos chips, el primero que integra los tres ejes del acelerómetro, y los tres ejes del giroscopio y el segundo un chip AK8963 que contiene los 3 ejes de un magnetómetro es decir la IMU seleccionada obtiene nueve grados de libertad “9 DOF” siendo capaz de medir la aceleración lineal, la velocidad de rotación y los vectores de campo magnético. Otra característica es su compatibilidad con las plataformas Arduino para las cuales se han desarrollado librerías específicas que permiten leer los parámetros que emite la IMU.

Como se requiere obtener los parámetros 9DOF es necesario programar el sensor a través de la plataforma Arduino, para las pruebas de funcionamiento se conecta directamente el dispositivo al Arduino UNO, conectando como se muestra en el siguiente diagrama.

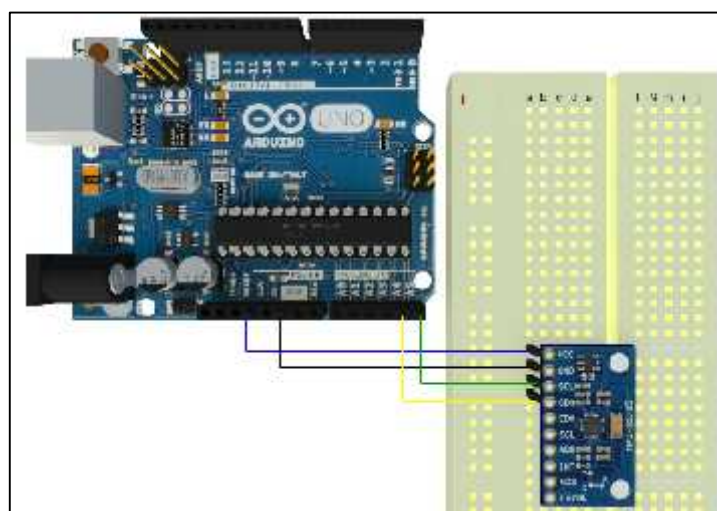


Figura 7-2. Conexión MPU 9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Para entender el funcionamiento de este sensor a continuación se explica cómo se da lectura de los ejes de cada chip que compone la IMU. El acelerómetro expresa la aceleración en las tres dimensiones del espacio, detectando valores al mover el dispositivo en X, Y y Z es decir si se desplaza hacia delante registra mediciones en el eje x, si se desplaza hacia arriba marca un cierto valor en el eje z y si se mueve hacia los lados detecta valores en el eje y. El giroscopio mide los cambios de orientación y velocidad angular, como en el caso del acelerómetro este registra valores al mover el dispositivo alrededor de los tres ejes de rotación.

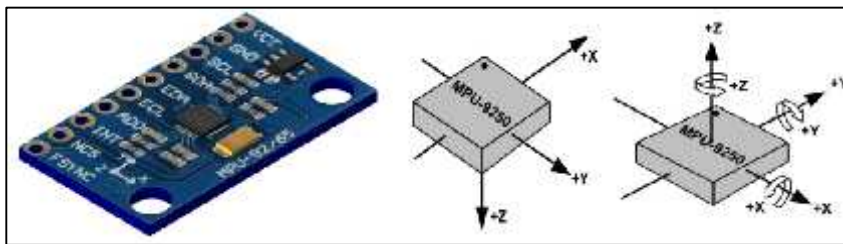


Figura 8-2. Ejes MPU 9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

En el anexo B se expone la programación para obtener dichos parámetros 9 DOF, para empezar con los datos del acelerómetro se requiere de librerías como son Wire, SPI, I2C, además de la librería MPU60X0 que es específica para este tipo de sensor. La variable que se declara para que se lean los datos es “accelgyro”, a continuación, se procede a observar las variables de los tres ejes en el monitor serial.

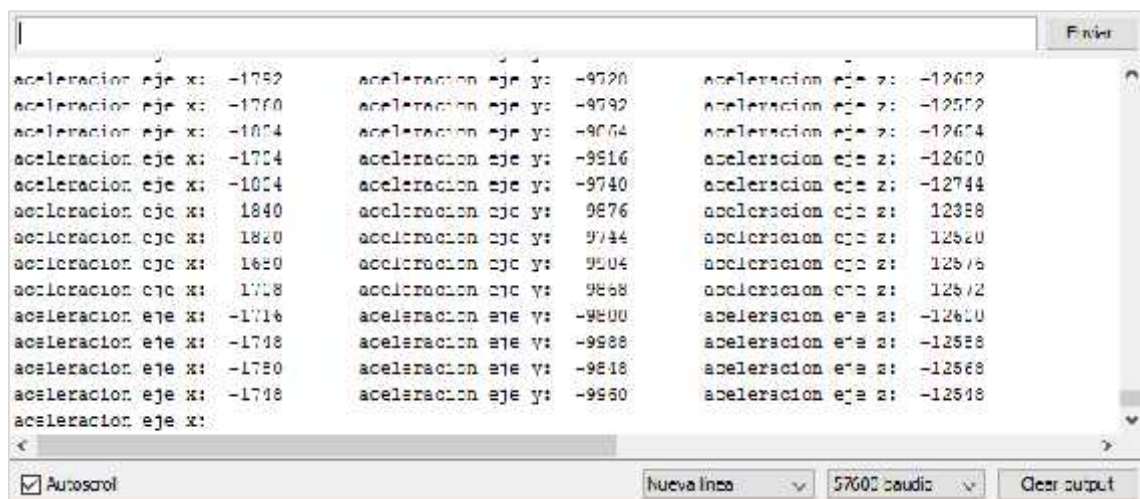


Figura 9-2. Ejes acelerómetro MPU 9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Respecto a los parámetros del giroscopio la variante en la programación es la declaración de variable para cada uno de los ejes, en este caso gx, gy, gz, se debe cambiar en void loop el comando recordando que para el acelerómetro se utiliza “accelgyro.getAcceleration” y para el giroscopio es “accelgyro.getRotation”.

```

giroscopio eje x: -54 giroscopio eje y: 100 giroscopio eje z: -161
giroscopio eje x: 36 giroscopio eje y: 80 giroscopio eje z: -92
giroscopio eje x: 168 giroscopio eje y: 20 giroscopio eje z: 30
giroscopio eje x: 200 giroscopio eje y: 9 giroscopio eje z: 8
giroscopio eje x: 228 giroscopio eje y: 12 giroscopio eje z: 40
giroscopio eje x: 151 giroscopio eje y: 41 giroscopio eje z: -86
giroscopio eje x: 23 giroscopio eje y: 92 giroscopio eje z: -113
giroscopio eje x: -35 giroscopio eje y: 118 giroscopio eje z: -113
giroscopio eje x: -35 giroscopio eje y: 135 giroscopio eje z: -103
giroscopio eje x: 92 giroscopio eje y: 55 giroscopio eje z: -14
giroscopio eje x: 118 giroscopio eje y: 18 giroscopio eje z: -12
giroscopio eje x: 165 giroscopio eje y: -16 giroscopio eje z: -81
giroscopio eje x: 141 giroscopio eje y: 29 giroscopio eje z: -107
giroscopio eje x: -6 giroscopio eje y: 119 giroscopio eje z: -101
giros
 Autodespl
Nueva línea 57600 baudio Clear output

```

Figura 10-2. Ejes giroscopio MPU 9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Entonces en teoría así funciona el chip que integra el acelerómetro y giroscopio, a continuación, demostraremos gráficamente que sucede al comparar dos estados del sensor ya sea en reposo o en movimiento para lo cual se conecta el dispositivo a la placa electrónica, se carga la programación anterior y se visualiza en el serial plotter de la plataforma Arduino IDE.



Figura 11-2. Reposo acelerómetro MPU-9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Como se puede observar en la figura tenemos tres líneas que cada una corresponde a los ejes que lee el sensor respecto a la aceleración, la línea de color azul que es el eje X, la de color rojo es el

eje Y por último la línea de color verde es el eje Z. Se puede observar además que en este estado existen pocas variaciones comprobando que este sensor reduce el error producido por el ruido. En cambio, como se observa en la figura 11-2 al ejercer una fuerza el sensor detecta la aceleración incrementando los valores en sus ejes manteniéndose en un rango de ± 16000

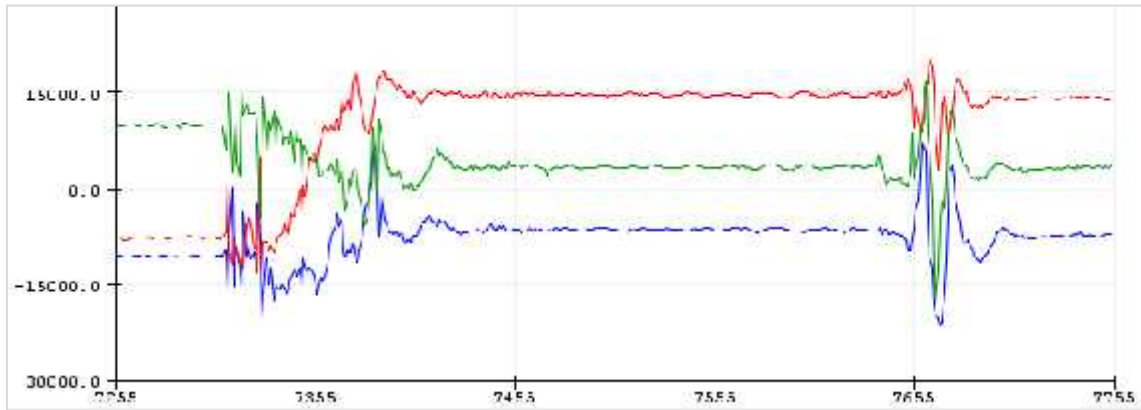


Figura 12-2. Movimiento acelerómetro MPU-9250

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Es el turno del giroscopio de igual manera observaremos en reposo y movimiento. La siguiente figura 12-2 demuestra el primer estado en donde el eje X de color azul es el que menor variación presenta permaneciendo en un rango entre 0 y 200. El eje con mayor variación es el Y de color rojo este se encuentra entre ± 250 y por último el eje Z de color verde tiende a disminuir manteniéndose entre los valores negativos.

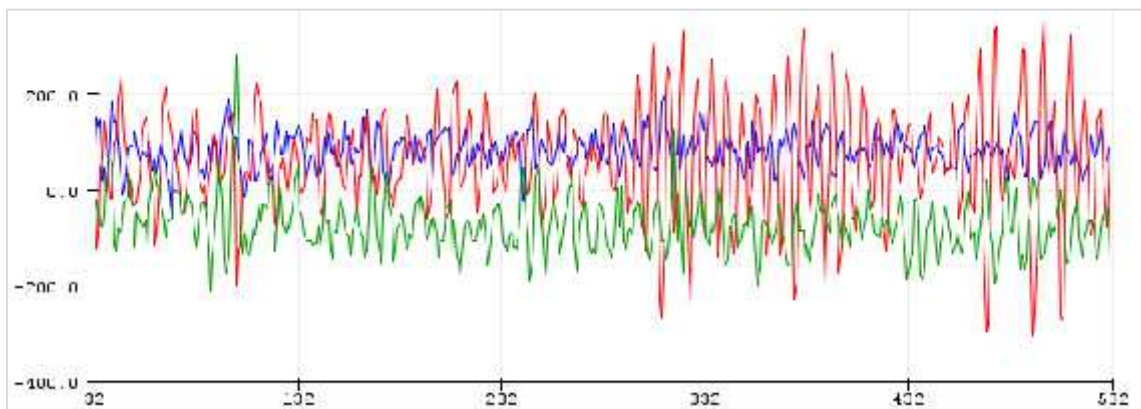


Figura 13-2. Reposo giroscopio MPU-9250

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Al mover el sensor se registra cambios entre ± 20000 en los tres ejes como se observa a continuación.

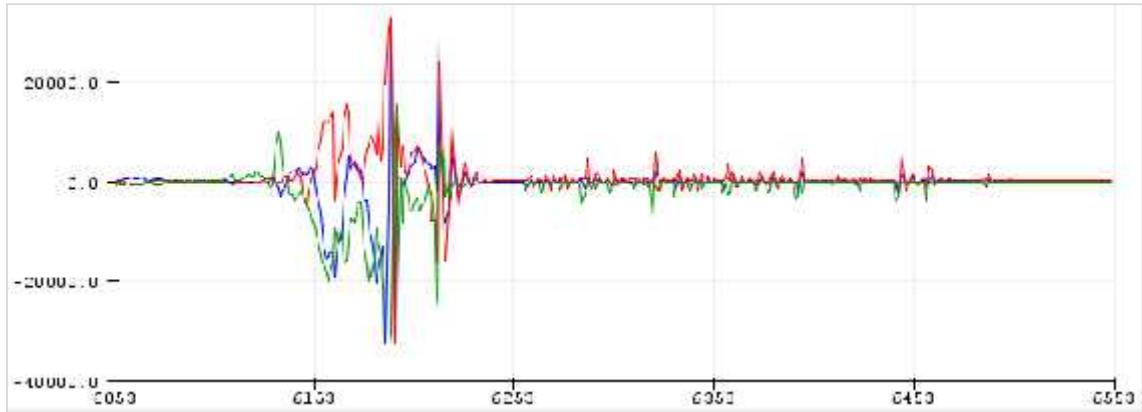


Figura 14-2. Movimiento acelerómetro MPU-9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

El segundo chip es el magnetómetro AK8963 que permite leer las componentes del campo magnético en donde los valores obtenidos se basan en el Norte magnético. Los datos de medición aumentan en proporción a que aumenta la densidad del flujo magnético en las direcciones de las flechas de los ejes X, Y y Z.

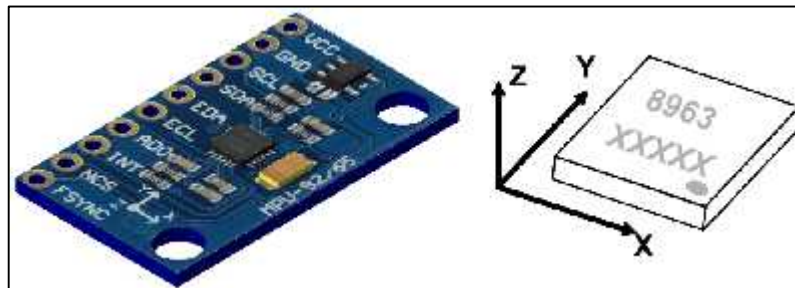


Figura 15-2. Ejes del magnetómetro AK8963 MPU-9250
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

La programación que se necesita para obtener los datos del magnetómetro AK8963, tiene mayor número de líneas de código donde lo principal es incluir la librería propia del dispositivo, las funciones auxiliares para que lea los ajustes de sensibilidad y los registros del magnetómetro, además de calcular los valores de los ejes en mili gauss para lo cual se incluyen los datos de calibración de fábrica de la hoja de datos, por último, los comandos necesarios para visualizar los ejes en el monitor serial.

magnetometro eje x:	magnetometro eje y:	magnetometro eje z:
49.59	-156.98	-65.59
49.59	-156.98	-65.59
49.07	-149.77	-54.97
32.94	-161.35	-61.01
32.94	-161.07	-61.01
46.57	-154.07	-60.57
49.59	-154.07	-56.47
60.22	-164.20	-54.97
60.22	164.25	54.97
48.54	165.71	65.59
48.05	158.44	64.05
60.22	158.53	57.99
60.22	158.53	57.99
48.05	-158.44	-54.97

Figura 16-2. Ejes magnetómetro MPU 9250

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Al igual que con el chip anterior, se compara el estado en reposo y en movimiento del sensor para verificar los rangos que se especifican en la datasheet y además ver el comportamiento. Como se observa en la figura el eje X es la línea de color azul este parámetro se mantiene cercano al valor de 0, el eje Y con el color rojo tiene un valor aproximado de 100 positivo y la última línea que corresponde al eje Z se mantiene cerca de 400 negativo.

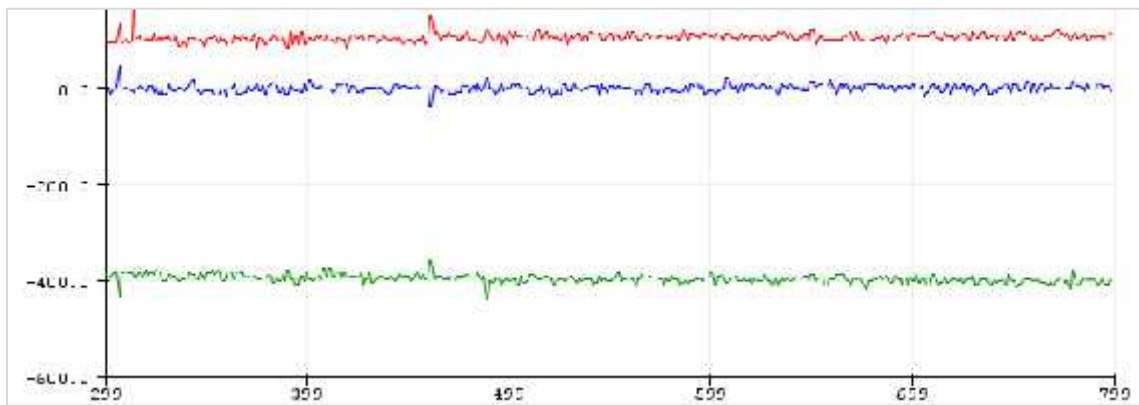


Figura 17-2. Reposo magnetómetro MPU-9250

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Ahora que sucede cuando se presenta movimiento, se puede observar que el eje X varía entre valores positivos y negativos presentando valores picos cercanos a 100 positivo y 250 negativo. Los datos del eje Y de igual manera tienen un pico aproximado de 200 positivo y 100 negativo, por último, el eje Z se mantiene en un rango negativo.

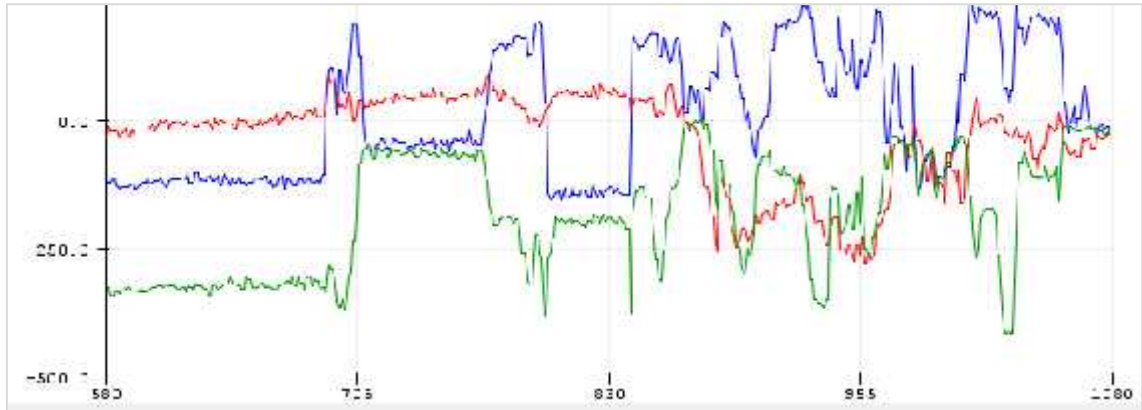


Figura 18-2. Movimiento magnetómetro MPU-9250
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

2.2. Integración GPS – IMU

Después de haber realizado la selección de dispositivos tanto para GPS como para IMU, se prosigue con la programación y la integración de los mismos. Para la programación se hace uso de la placa Arduino MEGA 2560, misma que ha sido seleccionada por su gran capacidad de almacenamiento de datos, debido a que las pruebas realizadas para la placa Arduino UNO demostraron que esta no poseía la capacidad de almacenamiento necesaria.

A continuación se presenta el diagrama de conexión de los dispositivos.

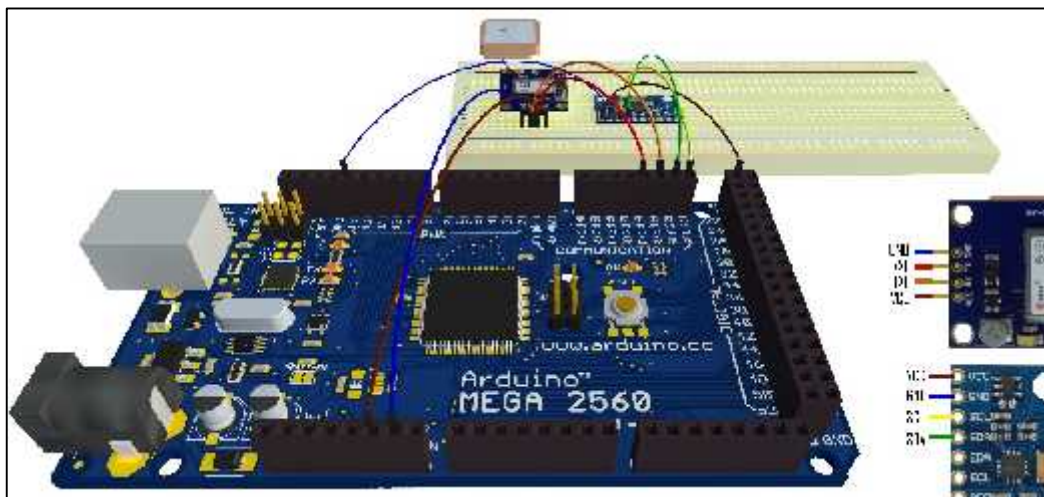


Figura 19-2. Conexión dispositivos
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Trabajar con las placas microcontroladoras de Arduino, posee varias ventajas por el hecho de ser de tipo “código abierto”, además de que constantemente se publican librerías y programaciones para los diferentes dispositivos.

En cuanto al GPS NEO 6M que es el utilizado para el presente proyecto, se cuenta con la librería TinyGPS++ que es una librería Arduino que permite el análisis de flujos de datos NMEA proporcionados por módulos GPS. Mediante este análisis de flujos es posible extraer posición, fecha, hora, altitud, velocidad y curso de los dispositivos GPS, también se cuenta con la versión anterior de esta librería denominada TinyGPS, sin embargo se utilizará TinyGPS++ por ser la más actualizada y poseer una interfaz de programación considerablemente más simple de usar.

La descarga de esta librería como el resto de librerías para Arduino, es completamente libre y no amerita de ningún pago previo, la instalación se la realiza únicamente trasladando a la ubicación configurada en el software de Arduino que por defecto se encuentra en la carpeta “Documentos” pero es posible modificar tal dirección en: Archivo/Preferencias/Localización del proyecto.

Por otro lado para la programación correspondiente al IMU (Unidad de medición inercial) del cual ha sido seleccionado el MPU 9250, se ha hecho uso de la programación publicada por Fabio Veresano en su biblioteca denominada FreeIMU, la cual originalmente hace uso del filtro MARG (Magnético, Angular y Gravedad) basado en la aplicación MARG de Sebastián Madgwick del DCM de Mahoney complementario en forma de cuaternión.

Fabio Veresano en sus publicaciones incluyó soporte para las placas:

- Sparkfun IMU Digital Combo Board - 6 grados de libertad ITG3200 / ADXL345 SEN-10121
- Sparkfun 9 grados de libertad - Razor IMU SEN-10736
- Sparkfun 9 grados de libertad - Sensor Stick SEN-10724
- Sparkfun 9 grados de libertad - Sensor Stick SEN-10183
- DIYDrones ArduIMU + V3
- Placas genéricas MPU6050 Breakout (por ejemplo: GY-521, SEN-11028 y otras MPU6050 que tienen el pin MPU6050 AD0 conectado a GND.)

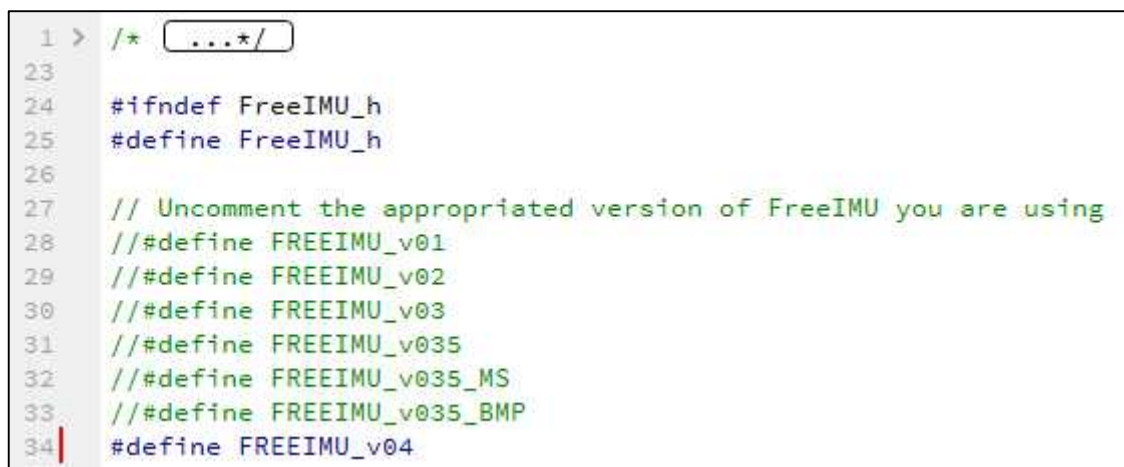
Esto equivale a los sensores:

Acelerómetro ADXL345B, Giroscopio ITG3200, L3GD20H Magnetometro HMC58X3 Combos MPU-6000: Giro y acelerómetro MPU-6050: y sensores MS5611BA, BMA180.

Además, Fabio proporcionó un software de calibración para acelerómetro y magnetómetro visual desarrollado en Python, lamentablemente Fabio falleció en diciembre del 2013.

En la actualidad el gran aporte de Fabio es usado en un gran número de proyectos por parte de la comunidad, por lo que su biblioteca FreeIMU es continuamente modificada, actualizada y mejorada, la versión utilizada para el presente proyecto posee soporte para varios filtros de fusión sensorial y se encuentra disponible en la página de Github, cada filtro ha sido analizado con el fin de seleccionar el más idóneo para el dispositivo seleccionado (MPU 9250).

No obstante, antes de realizar la selección del filtro, se debe realizar la selección del dispositivo, esto se logra mediante el uso de un editor de códigos que permita modificar el contenido del archivo “FreeIMU.h” que se encuentra entre las bibliotecas proporcionadas por Fabio Veresano.



```
1 > /* ... */
23
24 #ifndef FreeIMU_h
25 #define FreeIMU_h
26
27 // Uncomment the appropriated version of FreeIMU you are using
28 // #define FREEIMU_v01
29 // #define FREEIMU_v02
30 // #define FREEIMU_v03
31 // #define FREEIMU_v035
32 // #define FREEIMU_v035_MS
33 // #define FREEIMU_v035_BMP
34 #define FREEIMU_v04
```

Figura 20-2. Selección de IMU

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Después de realizar varias pruebas con las distintas versiones disponibles se concluyó que los mejores resultados se obtuvieron con la versión FREEIMU_v4, y su activación se logra eliminando el comentario de la línea de código 34, tal como se muestra en la Figura 19-2.

Para habilitar el magnetómetro en el algoritmo de fusión sensorial, se consigue mediante la modificación de la línea “//define DISABLE_MAGN”, para lo cual se debe dejar tal línea de código en forma de comentario agregando dos veces el símbolo “/”.

También es importante especificar el ángulo de declinación magnética para iCompass, lo cual se especifica en la línea de código 68, y para el presente proyecto se configuró con un valor de 3, el cual se obtuvo de la página web del Centro Nacional de Datos Geofísicos de NOAA, tal como se muestra en la Figura 20-2. En donde se muestra el ángulo de declinación magnética para la latitud y longitud correspondientes a la ciudad de Riobamba de Ecuador.

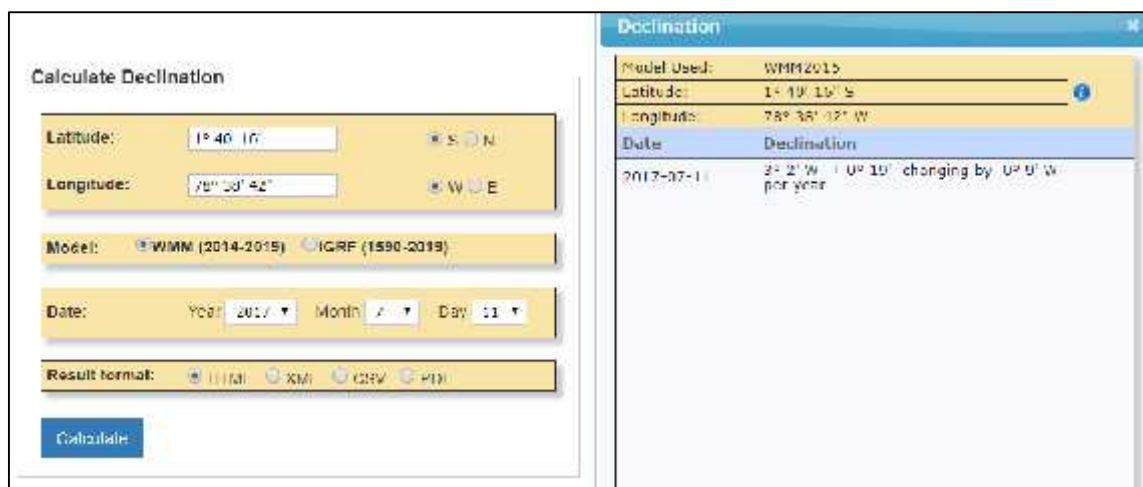


Figura 21-2. Obtención de la declinación magnética.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Para la selección del algoritmo de fusión sensorial se optó por el filtro AHRS de Sebastián Madgwick, el cual es de tipo open source o código abierto, los resultados obtenidos por este filtro en este caso poseían menos deriva y con datos más estables, a continuación se muestran los resultados obtenidos por cada filtro en donde se aprecia claramente que la mejor opción fue el filtro AHRS.

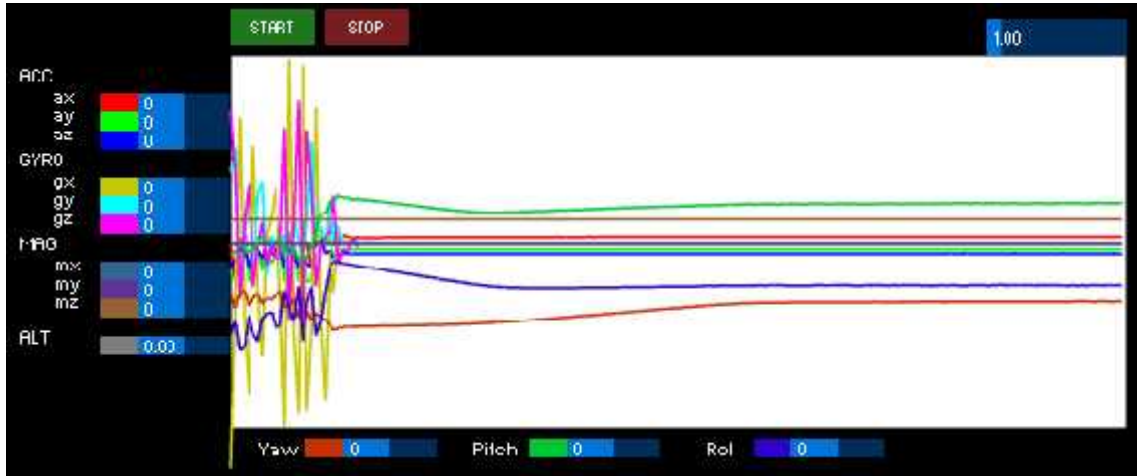


Figura 22-2. Datos de sensores con filtro Madgwick Gradient Descent

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Como se puede apreciar en la figura, con el filtro Madgwick Gradient Descent, después de exponer la IMU a movimiento y dejarlo inmóvil, los datos pasan por un lapso de estabilización considerable, lo cual denota imprecisión de resultados, de igual manera después de estabilizarse se aprecia que los datos poseen cierto nivel de ruido.

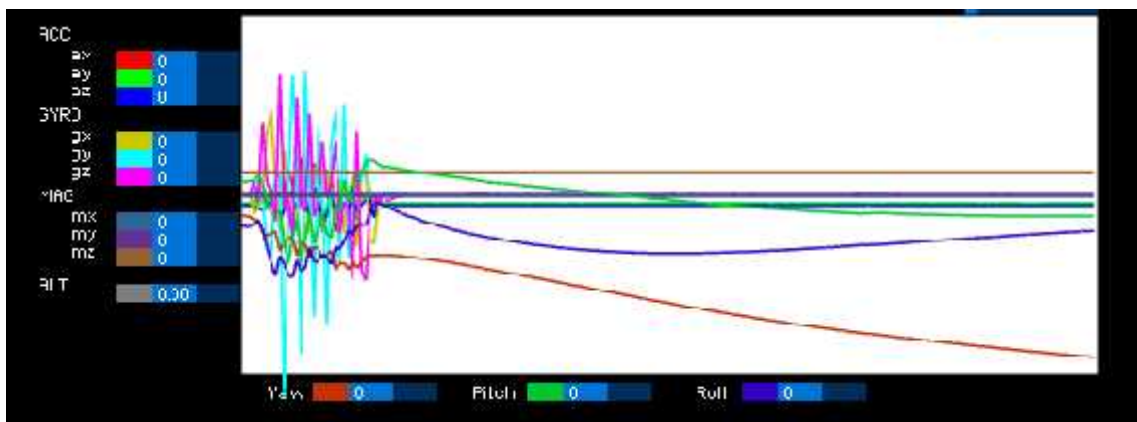


Figura 23-2. Datos de sensores con filtro Madgwick implementation of Mahoney DCM.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Por otra parte, con el filtro Madgwick implementation of Mahoney DCM después de someter la IMU a movimiento y luego estabilizarlo, se puede notar que los valores del acelerómetro pasan por un retardo en cuanto a la estabilización lo cual es razón suficiente para descartar el uso de este filtro en nuestro dispositivo que busca precisión en los datos emitidos.

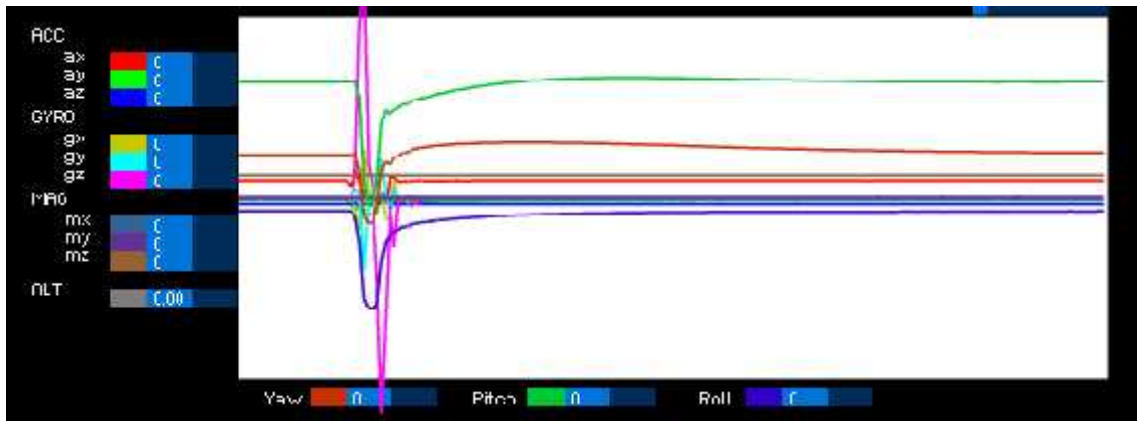


Figura 24-2. Datos de sensores con filtro DCM Implementation.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

El filtro DCM Implementation, por su parte se puede denotar que los datos que emite poseen un ruido aceptable, no obstante después de ser sometida la IMU al movimiento, también pasa por un lapso de estabilización, que a pesar de ser menor al de los filtros anteriormente analizados, sigue sin ser una condición ideal.

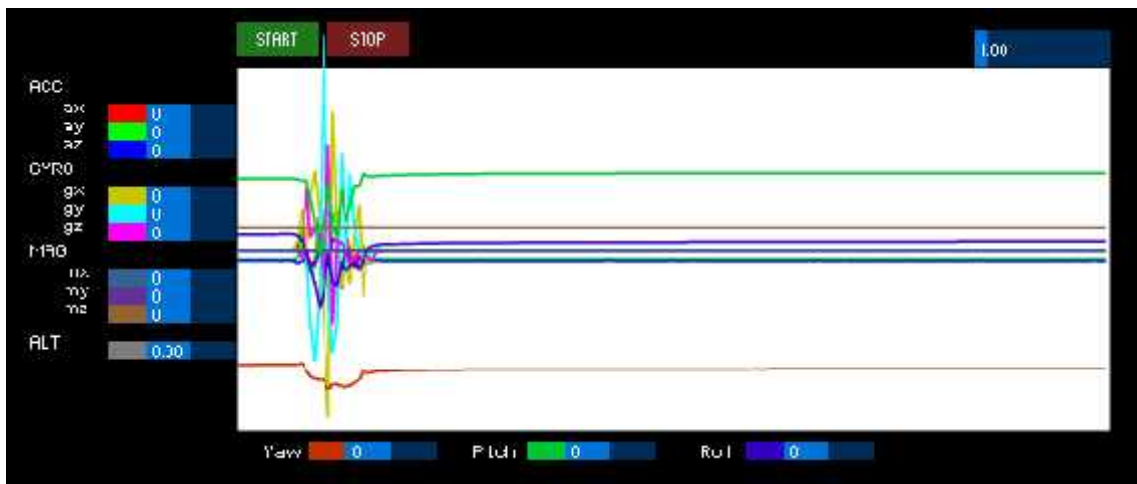


Figura 25-2. Datos de sensores con filtro AHRS de Sebastián Madgwick.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Finalmente al analizar el filtro AHRS de Sebastián Madgwick, se aprecian los resultados esperados, después de someter la IMU a movimiento para luego dejarla en reposo, los datos son estabilizados inmediatamente y sin la presencia de ruido en los mismos. En tal virtud se selecciona este filtro para la fusión sensorial del MPU-9250 utilizado por el presente proyecto.

La selección del filtro también amerita configurar los valores de K_p y K_i , el valor de K_p corresponde a la ganancia proporcional la cual regula la velocidad de convergencia del acelerómetro y magnetómetro, por su parte el valor de K_i corresponde a la ganancia integral, la cual gobierna la tasa de convergencia de los sesgos del giroscopio.

Tomando en cuenta la sugerencia por parte de Sebastián Madgwick autor del filtro AHRS, que en un post de DIY Drones (2011), sugirió dejar el valor de K_i como 0 y comenzar con un valor de K_p de 5, y a partir de ahí reducir el valor de K_p dependiendo de los errores de calibración tanto de polarización como de sensibilidad del giroscopio y dinámica angular esperada de aplicación.

En tal virtud y siguiendo la recomendación proporcionada por Sebastián Madgwick, se han seleccionado los valores de convergencia y ganancia integral más idóneos, mismos que se configuran a partir de la línea 95 del código de la biblioteca FreeIMU.h, tal como se puede apreciar en la Figura 25-2.

```
94 #elif defined(FREEIMU_v04)
95     #define twoKpDef (2.0f * 0.75f)
96     #define twoKiDef (2.0f * 0.1625f)
97     #define betaDef 0.085f
```

Figura 26-2. Datos de sensores con filtro AHRS de Sebastián Madgwick.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Por su parte, el término de β cumple con un rol en específico que para poder entenderlo es necesario conocer que el filtro usa un cuaternión para representar así la orientación, por tanto se utilizan los valores proporcionados por el giroscopio para representarlos como velocidad de cambio de un cuaternión, y al integrarlos se obtiene la orientación estimada de todos los sensores en conjunto. No obstante la estimación realizada se desvía a través del tiempo y por tal motivo surge la necesidad de corregirlo. De tal manera utilizando la orientación estimada de la matriz de sensores es posible realizar el cálculo del error y utilizando métodos de programación no lineal se consigue calcular a su vez un vector en el espacio de cuaternión que será quien defina la corrección requerida al normalizarlo como vector unitario y multiplicarlo por un factor o ganancia β que corresponde a la tasa de convergencia.

En resumen, si se opta por establecer un valor de β muy grande, el resultado será que la salida del filtro será ruidosa, puesto que tratará de realizar un rastreo completo de los datos proporcionados por el acelerómetro y magnetómetro. Y si al contrario la velocidad de

convergencia es muy baja, se producirá deriva integral muy elevada. En tal virtud un valor idóneo de beta se define como aquel que produce que la velocidad de convergencia debida a la etapa correctiva sea igual a la tasa de divergencia correspondiente a la deriva integral. Concluyendo con esto que el valor o las unidades de beta representan la magnitud de la tasa de cambio del cuaternión.

2.2.1. Configuración del acelerómetro, giroscopio y magnetómetro

Es necesario establecer la configuración de los tres dispositivos que conforman la IMU, en función de las especificaciones técnicas de los dispositivos utilizados. Tal como se ha mencionado anteriormente, La IMU utilizada por el presente proyecto es la MPU 9250, misma que integra 3 sensores los cuales se conforman de conversores analógicos digitales (ADC) de 16bits. En cuanto al acelerómetro concierne, este puede ser ajustado en los valores de: ($\pm 2g$, $\pm 4g$, $\pm 8g$, y $\pm 16g$) mientras que el giroscopio puede ajustarse en (± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$), finalmente el magnetómetro hasta $\pm 4800\mu\text{T}$.

Teniendo en cuenta las características de los sensores, se establece la configuración en la programación por medio de la modificación del archivo FreeIMU.cpp. El uso de la nomenclatura tanto para el acelerómetro como para el giroscopio se muestra a continuación.

Tabla 4-2. Comparación dispositivos inalámbricos

Configuración giroscopio	
MPU60X0_GYRO_FS_250	+/- 250 deg/sec, 131 LSB/°/s
MPU60X0_GYRO_FS_500	+/- 500 deg/sec, 65.5 LSB/°/s
MPU60X0_GYRO_FS_1000	+/- 1000 deg/sec, 32.8 LSB/°/s
MPU60X0_GYRO_FS_2000	+/- 2000 deg/sec, 16.4 LSB/°/s
Configuración acelerómetro	
MPU60X0_ACCEL_FS_2	+/- 2g, 16384 LSB/g
MPU60X0_ACCEL_FS_4	+/- 4g, 8192 LSB/g
MPU60X0_ACCEL_FS_8	+/- 8g, 4096 LSB/g
MPU60X0_ACCEL_FS_16	+/- 16g, 2048 LSB/g

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

En cuanto al magnetómetro, no es posible cambiar la configuración de $\pm 4800\mu\text{T}$. para el MPU 9250. A continuación se muestra la configuración establecida mediante el uso de la nomenclatura

detallada anteriormente en la Tabla 4-2. Codificación presente a partir de la línea de código 685 del archivo FreeIMU.cpp

```
accgyro = MPU60X0(false, accgyro_addr);
accgyro.initialize();
accgyro.setDLPFMode(MPU60X0_DLPF_BW_20);
accgyro.setI2CMasterModeEnabled(0);
accgyro.setI2CBypassEnabled(1);
accgyro.setRate(0x13); //Establece la frecuencia de muestreo a 8000/1 + 7 = 1000Hz
accgyro.setFullScaleGyroRange(MPU60X0_GYRO_FS_2000);
accgyro.setFullScaleAccelRange(MPU60X0_ACCEL_FS_2);
gyro_sensitivity = 16.4f;
delay(5);
delay(30);
```

2.2.2. Calibración de la IMU

La IMU o (Unidad de medición Inercial) utilizada es la MPU 9250, dispositivo de alta precisión que integra señales de tres sensores, adquiriendo con ello 9 grados de libertad, no obstante la calibración del dispositivo es una acción que no puede ser pasada por alto para la obtención de datos confiables y con el mínimo error posible.

Omitir la calibración de la IMU conllevaría a la presencia de comportamientos extraños fuera de lo real. En tal virtud es necesario establecer los parámetros que mantengan el funcionamiento óptimo del dispositivo, existe la posibilidad de crear dentro de la programación un algoritmo capaz de realizar la calibración automáticamente, no obstante realizar esta acción posee ciertas desventajas que radican en el tiempo que necesitaría el programa para llevar a cabo su función, es decir que los datos iniciales provenientes del dispositivo carecerían en cierta medida de confiabilidad.

Por tanto la solución óptima a este inconveniente, es establecer manualmente los parámetros de calibración, para con ello omitir el proceso de calibración a cada instante en que se inicie la toma de mediciones por parte de los sensores.

La opción utilizada para la calibración del MPU 9250 del presente proyecto, es por medio de la aplicación en Python desarrollada por Fabio Veresano, no obstante para poder hacer uso de tal aplicación para la calibración del MPU 9250 controlado por el Arduino MEGA 2560, es necesario llevar a cabo una serie de procedimientos, iniciando por la instalación de complementos, y aplicaciones indispensables para poder correr la aplicación, y seguido también se deberá modificar ciertos parámetros dentro de la configuración. Tales procedimientos realizados se detallan a continuación.

2.2.2.1. Requisitos de instalación

Como es de esperarse, el ejecutar una aplicación desarrollada en Python requiere en primer lugar la instalación del software, cabe resaltar que la versión solicitada será la de Python 2.7.13 y en cuanto a los complementos adicionales que se requiere, será necesario instalar las versiones compatibles a la versión 2.7.13 de Python.

Los complementos requeridos son:

- Numpy-1.9.1-win32-superpack-python2.7
- PyOpenGL-3.1.0-win32
- PyQt4.4.11.2-gpl-Py2.7-Qt4.8.6-x32
- Pyqtgraph-0.9.8.win32
- Pyserial-2.7win32
- Scipy-0.14.0-win32-superpack-python2.7

Como se ha mencionado, para poder hacer uso de la aplicación de calibración, se necesitó instalar la totalidad de los complementos y en las versiones detalladas, omitir uno de ellos resultará en error en el momento de compilar la aplicación.

2.2.2.2. Configuraciones de codificación

Como primer punto importante se especifica el tipo de procesador que se utilizará en la calibración ya sea de 16 o de 32 bit. Debido a que el procesador que se utilizó es el del Arduino Mega 2560,

se configuró para un procesador de 16 bit, para lo cual en la línea de código 47 del programa en Python denominado cal_py.py, se estableció la línea “Word = 2”.

En caso de necesitar configurar la programación para un procesador de 32 bit, se deberá cambiar la línea de código 47 por el código: “Word = 4”.

La versión original publicada por Fabio Veresano, poseía un pequeño error de cálculo el cual se corrigió editando el archivo cal_lib.py de la biblioteca de calibración. El cambio se efectuó modificando la línea de código:

```
H = numpy.array([x, y, z, -y**2, -z**2, numpy.ones([len(x)], 1)])
```

Siendo el código correcto:

```
H = numpy.array([x, y, z, -y**2, -z**2, numpy.ones([len(x)])])
```

También fue necesario especificar el baudrate para el cual se necesitaba ejecutar la aplicación, mediante la línea de código 168 con la sentencia “baudrate=57600,” el valor especificado que debe coincidir con el baurate configurado en la programación de la placa controladora, el cual fue establecido en 57600 Baudios.

La versión original posee una configuración de 115200 Baudios, sin embargo produce que la aplicación se bloquee al conectarse.

2.2.2.3. Ejecución de la aplicación de calibración

Finalmente, se ejecuta la aplicación, cabe resaltar que para que el programa reconozca los datos de los sensores fue necesario que la placa controladora incorpore la programación correspondiente a su funcionamiento, es decir la programación denominada FreeIMU_serial misma que como se indicó anteriormente se encuentra disponible para descarga libre en la red. A continuación se muestra la ventana que se abre después de ejecutar la aplicación de calibración:

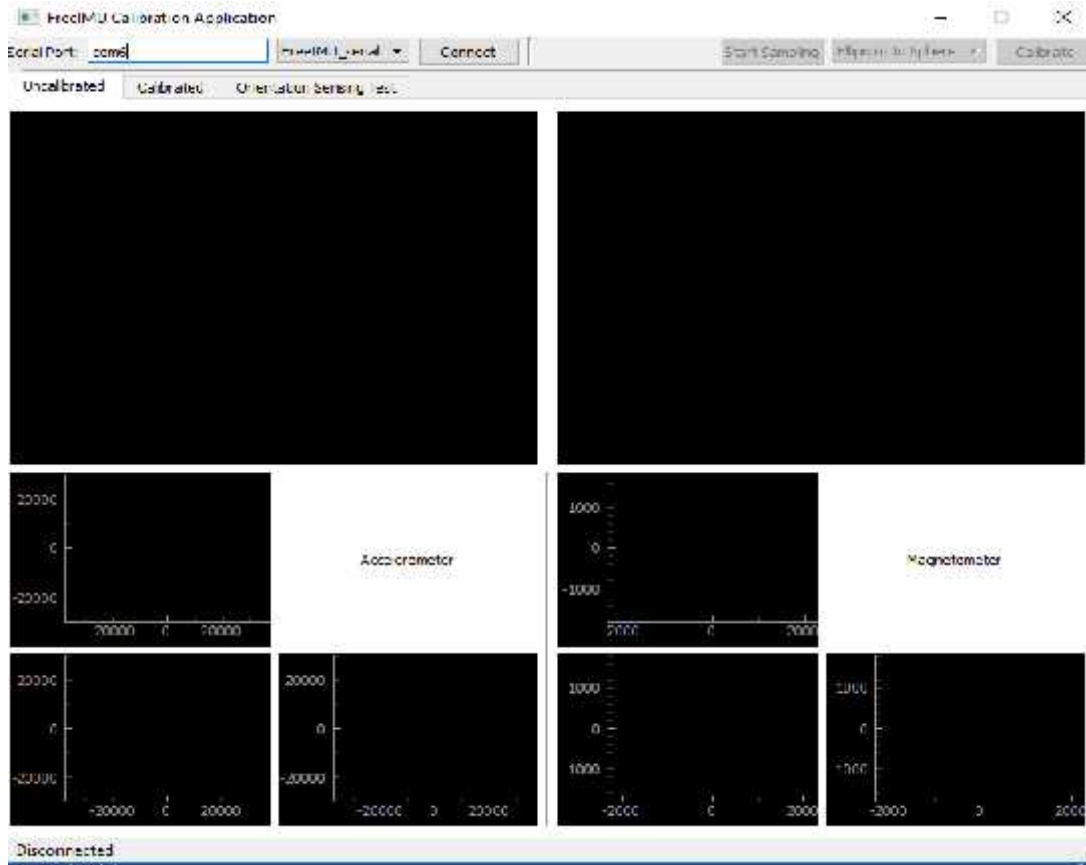


Figura 27-2. Aplicación de calibración en Python.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

En la ventana de la aplicación se especificó el puerto utilizado en la conexión del Arduino Mega, mismo que varía según el ordenador. Seguido se presionó el botón conectar y finalmente el botón Start Sampling. Después de culminado el procedimiento detallado, la aplicación inició la recepción de datos por parte del MPU 9250 utilizado, el cual se hace visible en la interfaz gráfica que la aplicación posee. Llegado a este punto se siguió un procedimiento basado en movimientos de los ejes de la unidad de medición inercial para poder de esta manera realizar la calibración visual.

Los algoritmos utilizados para la calibración se encargan de realizar una adaptación de los elipsoides mediante la disminución de una función de error, logrando con ello la mejora de los parámetros de un modelo matemático para la calibración de los datos en bruto proporcionados por el sensor. Sin embargo se debe tener en cuenta que la sensibilidad de convergencia del algoritmo es muy sensible y por este motivo es importante que los datos de entrada sean distribuidos eficazmente para que el algoritmo pueda ajustarlos de manera correcta con el modelo teórico de elipsoide.

En tal virtud, para que el software pueda obtener datos ordenados, se procedió a realizar movimientos en la unidad de medición inercial, tales movimientos se realizaron tomando en cuenta un eje a la vez a una velocidad angular de 0,3 rpm aproximadamente. En este punto radica la importancia de realizar los movimientos lentamente alrededor de los 3 ejes dimensionales y en todas las posiciones del IMU, con el fin de proporcionar los datos necesarios para la calibración, un ejemplo de un orden de movimientos adecuado se muestra a continuación, mismo que fue utilizado para la calibración.

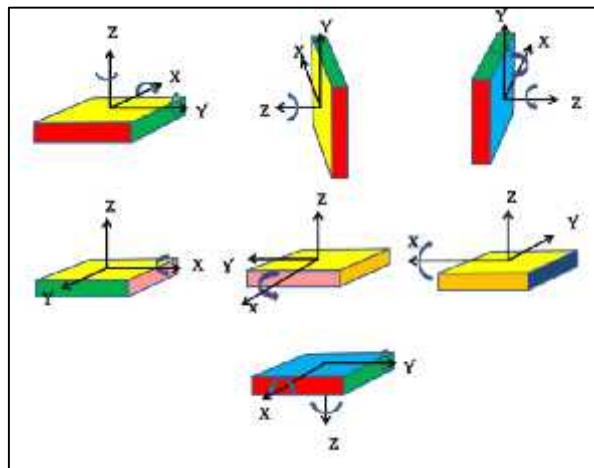


Figura 28-2. Orden de movimientos para calibración

Fuente: (Github, 2014), url: <https://goo.gl/bhzewk>

Una vez terminado el procedimiento, la aplicación muestra gráficas de los datos receptados, siendo lo ideal y lo esperado obtener las gráficas de acelerómetro y magnetómetro que se muestran a continuación.

Finalmente terminado todo el procedimiento detallado se logró obtener los datos de calibración para la corrección de errores de la IMU, en la misma aplicación fue posible visualizar los datos y después de grabarlos se generó un archivo denominado “calibration.h” mismo que se añadió a la biblioteca de Arduino, necesaria para la programación.

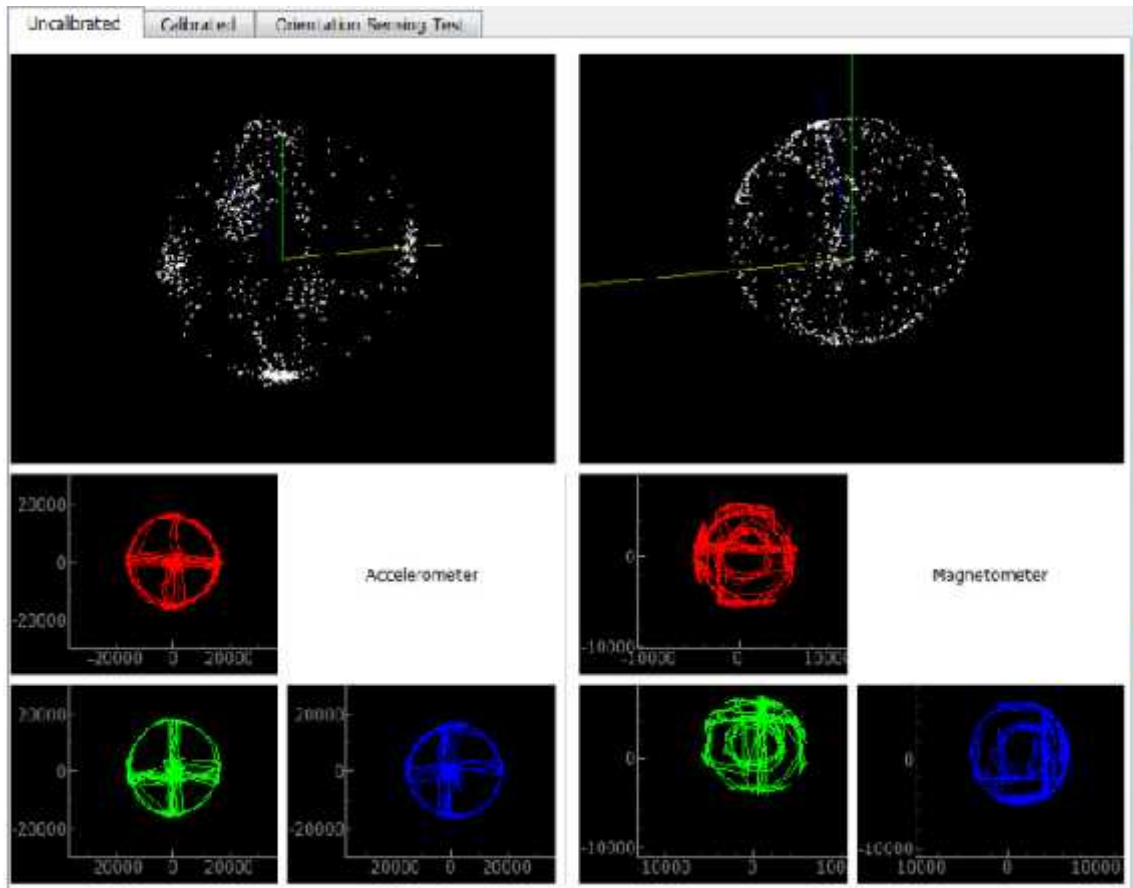


Figura 29-2. Gráficas de calibración

Fuente: (Github, 2014), url: <https://goo.gl/bhzewk>

2.2.3. Programación del Arduino Mega 2560

Después de haber calibrado la IMU se procedió a programar el Arduino Mega 2560, cabe resaltar en este punto que la versión FreeIMU de Fabio Veresano, fueron una fuente importante para la obtención de las bibliotecas que contenían tanto los controladores para la IMU como el filtro correspondiente, no obstante para lograr la obtención de datos por parte del GPS es necesario aumentar en la programación, tanto la biblioteca adecuada para el control del dispositivo GPS como el filtro correspondiente para la fusión sensorial.

El filtro utilizado en el presente proyecto es el filtro de Kalman, que como bien se sabe, es un algoritmo basado en un modelo de espacio de estados del sistema, mediante el cual se logra estimar tanto estados como salidas futuras, así también se consigue mediante la implementación del filtro disminuir los ruidos presentes en las señales de salida puesto que sus ecuaciones poseen probabilidades estadísticas que se enfocan en la aleatoriedad que la señal y el ruido poseen.

Las ecuaciones inmersas en el algoritmo del filtro de Kalman logran entre sí añadir un estimador de tipo predictor y corrector al sistema, característica que es utilizada por el presente proyecto para la estimación demográfica en el sistema de navegación.

En tal virtud la implementación del filtro en la programación para lograr la fusión sensorial mediante un algoritmo basado en predicción y corrección logrará la obtención de datos con mayor precisión con el menor ruido posible y la biblioteca utilizada para este fin es la publicada por Kristian Lauszus denominada como “Kalman”.

A continuación se detallan las líneas de código añadidas, mismas que son ajenas a la versión original publicada por Fabio Veresano que únicamente se encarga de la obtención de datos de la IMU.

```
#define HAS_GPS 1
#define BaudRate 57600
KalmanFilter kFilters[4];
int k_index = 3;
#if HAS_GPS
#define ss Serial1
#include <TinyGPS++.h>
static const unsigned long GPSBaud = 9600;
TinyGPSPlus gps;
#endif
#if HAS_GPS
#define ss Serial1
#include <TinyGPS++.h>
static const unsigned long GPSBaud = 9600;
TinyGPSPlus gps;
#endif
char cmd, tempCorr;
void setup() {
  Serial.begin(57600);
  Wire.begin();
  float qVal = 0.125; //Set Q Kalman Filter(process noise) value between 0 and 1
  float rVal = 32.; //Set K Kalman Filter (sensor noise)
```

```

for(int i = 0; i <= k_index; i++) { //Initialize Kalman Filters for 10 neighbors
kFilters[i].KalmanInit(qVal,rVal,5.0,0.5);
}
my3IMU.init(true);
delay(500);
#if HAS_GPS
ss.begin(GPSBaud);
#endif
pinMode(13, OUTPUT);
}
static void smartDelay(unsigned long ms)
{
#if HAS_GPS
unsigned long start = millis();
do
{
while (ss.available())
gps.encode(ss.read());
} while (millis() - start < ms);
#endif
}

```

Tal como se aprecia, se incluye en la programación la facultad de poder identificar si se posee o no conectado el dispositivo GPS, para lo cual se define la variable “HAS_GPS 1” en el caso de no contar con el GPS y únicamente se requiera obtener los datos de la IMU se deberá cambiar la mencionada definición por “HAS_GPS 0”. La biblioteca añadida es la de TinyGPS++, que como se analizó anteriormente en las programaciones de prueba, se estableció como la opción más idónea. En cuanto a los puertos de conexión para el GPS, se puede apreciar que se ha definido la variable “ss” en el Serial1, la cual puede ser modificada según se convenga. Finalmente un detalle importante a destacar es el Baudrate definido para el dispositivo GPS, a pesar de que el monitor serial se podrá comunicar con la placa controladora en un Baudrate de 57600, no se ha utilizado el mismo valor para el GPS puesto que la obtención de datos en el mencionado valor no es posible, por tanto se define la nueva variable “GPSBaud” en un valor de 9600, en el cual se logró con éxito la lectura de datos por parte del GPS.

2.2.4. *Aplicación de interpretación de datos*

Después de haber realizado la programación del Arduino Mega 2560, conectado a la IMU MPU 9250 y GPS NEO 6M, ya se logró la obtención de datos por parte de los sensores, no obstante la interpretación únicamente a nivel numérico es compleja y poco amigable con el usuario, en tal virtud surge la necesidad de implementar una aplicación capaz de interpretar los resultados y presentarlos al usuario de una forma más amigable.

Con el fin de poner a disposición una aplicación capaz de desarrollar una interfaz gráfica a partir de los resultados, se trabajó en el software Processing para dar cumplimiento a esta necesidad.

Nuevamente en esta ocasión se ha hecho uso de la aplicación de tipo “open source” desarrollada por Fabio Veresano, no obstante su versión original solo ofrece soporte para la lectura de datos por parte de la IMU, y en el caso de querer obtener los datos provenientes de la fusión sensorial con el módulo GPS, se requiere hacer ciertas modificaciones que se exponen a continuación.

En primer lugar es necesario la instalación de las librerías requeridas por la aplicación para el desarrollo de su interfaz gráfica. Las librerías utilizadas se enumeran a continuación:

- ControlP5
- G4P
- Gwoptics
- OBJLoader
- Peasycam
- Udp

Y en cuanto a la programación, a continuación se muestran las líneas de código añadidas para el reconocimiento e impresión de datos proporcionados por el GPS.

En primer lugar es necesario añadir las variables correspondientes al GPS.

```
float hdop, lat, longt, cog, sog, gpsalt, gpschars;  
float hdop_val, loc_val, gpsalt_val, sog_val, cog_val;  
float S;
```

```

float A;
String seapresscmd = "99";
float STATIONALTFT = 36.0;
float sea_press = 1013.25;          // Entrada de la presión local del nivel del mar
float declinationAngle = -13.1603; // Flushing, NY declinación magnética en grados
float SEA_PRESS = 1013.25;         // nivel de presión del nivel del mar por defecto en mbar
float KNOWNALT = 65.0
float INHG = 0.02952998751;       // Convertir mbar a inHg constante
float MB = 33.8638815;            // Convertir en inHg a mbar constante
float FTMETERS = 0.3048;
float METERS2FT = 3.2808399;
float PI = 3.14159;
float [] hq = null;
float [] Euler = new float [3];
int lf = 10; // 10 is '\n' in ASCII
byte[] inBuffer = new byte[22];

```

También se añade a la aplicación la facultad de poder detectar si se cuenta con el módulo GPS, mediante la variable “int HAS_GPS = 1”. Y en el caso de activar esta función, continúa con el siguiente procedimiento para la lectura de datos proporcionados por el módulo.

```

if(HAS_GPS == 1){
hdop_val = decodeFloat(inputStringArr[1]);
lat = decodeFloat(inputStringArr[20]);
longt = decodeFloat(inputStringArr[21]);
gpsalt = decodeFloat(inputStringArr[22]);
cog = decodeFloat(inputStringArr[23]);
sog = decodeFloat(inputStringArr[24]);

```

Para la impresión de información dentro de la interfaz gráfica:

```

fill(#ffff00);
text("Latitud:\n", 700, 130 );
text("Long:\n", 700, 175);
text("CoG:\n", 700, 220);

```

```

text("SoG:\n", 700, 265);
text("GPS Alt:\n", 700, 310);
if(motionDetect == 0) {
sog = 0;
cog = -9999; }
fill(#00CF00);
text(nfp(lat,3,5), 720, 155);
text(nfp(longt,3,5), 720, 200);
text(nfp(cog,3,2), 720, 245);
text(nfp(sog,3,2), 720, 290);
text(nfp(gpsalt,3,2), 720, 335);

```

2.2.4.1. Filtro de Kalman

El filtro de Kalman hace uso de una dimensión temporal que consiste en el análisis de la dinámica de las variables de estado y también utiliza una dimensión transversal en donde obtiene las mediciones en intervalos de tiempo de las variables observables, ésta dinámica es ejecutada en 2 etapas, la primera es la etapa de predicción y luego sigue la etapa de corrección.

Es decir, en la primera etapa hace la primera estimación de la variable de estado y a continuación se encarga de mejorarla utilizando datos de las observables, tiene la característica de que las futuras predicciones consideran las correcciones anteriores, lo cual se conoce como “extracción de señales”.

2.2.4.2. Algoritmo Filtro de Kalman

Como primer punto, el algoritmo filtro de Kalman, se compone de la ecuación principal:

Ecuación 1-2. Ecuación principal del algoritmo de Kalman

$$Z_t = H_t * x_t + u_t$$

Donde:

Z_t = Vector de variables observables

x_t = Vector de las variables de estado

H_t = Matriz de relación, variables observables con variables de estado

u_t = Error de medición de las variables observables.

El valor de u_t se adopta asumiendo que no se encuentra autocorrelacionado ni tampoco presenta correlacionado con las variables observables y además que posee una distribución normal con una media aritmética de cero.

La ecuación de estado por su lado se encarga de determinar la dinámica de las variables de estado.

Ecuación 2-2. Ecuación de la dinámica de las variables de estado

$$x_t = A_t * x_{t-1} + e_t$$

Donde:

e_t = Vector de errores

x_t = Vector de las variables de estado

e_t es incluido con el fin de hacer una modelación de la incertidumbre de variables de estado, el supuesto para este término es que sus datos poseen una distribución gaussiana con media de cero.

Las ecuaciones de predicción se rigen por:

Ecuación 3-2. Ecuación de predicción

$$x_{t+1,0}^p = A_t * x_{t,0}$$

Donde:

$x_{t,0}$ = Estimador óptimo del vector

Ecuación 4-2. Ecuación de predicción

$$P_{t+1}^p = A_t * P_t * A_t^t + Q$$

Donde:

P_t = Varianza del vector de estados

Tal como se mencionó anteriormente, los datos de las predicciones son mejorados tomando en cuenta una corrección que se define mediante la siguiente ecuación:

Ecuación 5-2. Ecuación de corrección

$$x_{t+1,0} = x_{t+1,0}^p + K_{t+1} * \frac{(z_{t+1} - z_{t+1}^p)}{e_{t+1}} = x_{t+1,0}^p + K_{t+1} * (z_{t+1} - H_{t+1} * x_{t+1,0}^p)$$

Donde:

K_t = Ganancia de Kalman.

```
public class MyKalman {
private float Q = 0.0000005;
private float R = 0.01;
private float P = 1, X = 0, K;
private void measurementUpdate(){
K = (P + Q) / (P + Q + R);
P = R * (P + Q) / (R + P + Q);
public float update(float measurement){
measurementUpdate();
float result = X + (measurement - X) * K;
X = result;
return result;
}
```

Finalmente los resultados obtenidos se muestran a continuación.

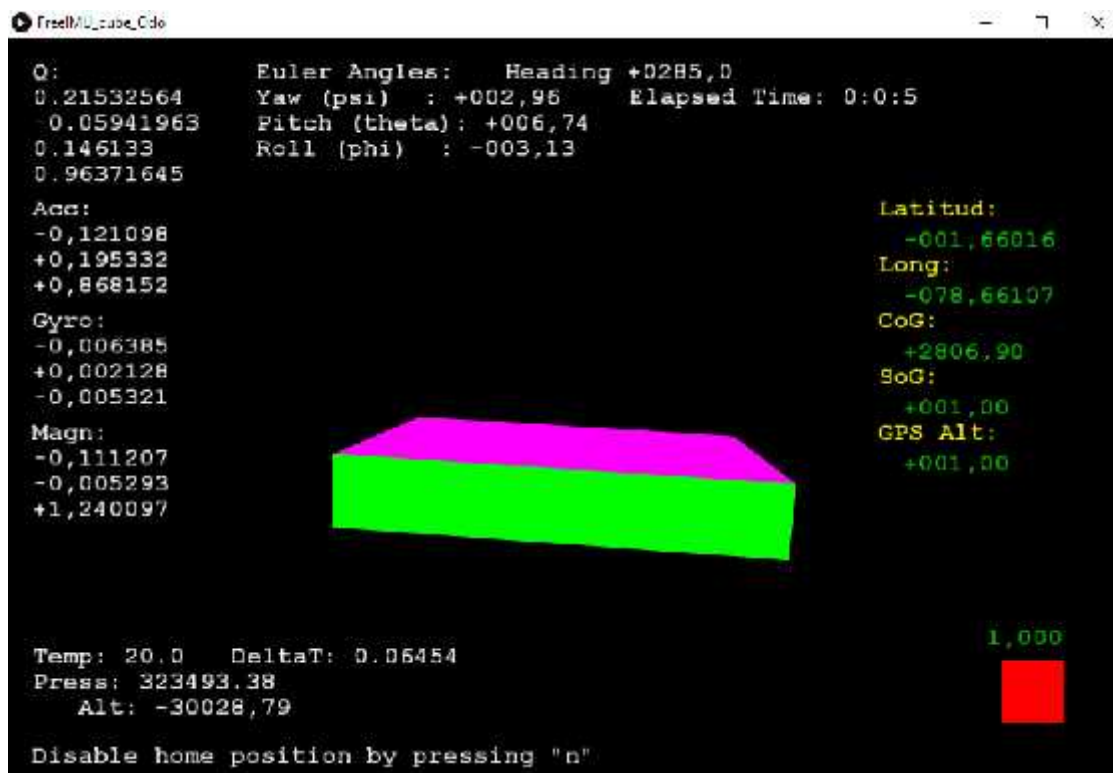


Figura 30-2. Gráfica de interfaz

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Tal como se puede apreciar en la Figura 29-2. Se hacen visibles los datos provenientes del acelerómetro, giroscopio y magnetómetro, a su vez, gráficamente se muestra un cubo en tres dimensiones que representa la orientación en tiempo real del MPU 9250, conseguir esta representación gráfica requiere de la obtención de los tres ángulos de navegación denominados como “yaw, pitch y roll” también conocidos como la actitud del sistema. En sí son coordenadas angulares mediante las cuales es posible describir un triedro en comparación de otro triedro referencial, al ser coordenadas angulares, son representadas con la unidad de medición “grados”. Cabe resaltar que la obtención de estas coordenadas angulares se da únicamente a través de la unidad de medición inercial y el GPS presente en el proyecto no interviene en esta parte de la obtención de datos. Sin embargo la IMU no entrega los valores angulares directamente, sino valores de voltaje que varían en función de la velocidad angular a la que se somete el dispositivo.

Es importante conocer que los ángulos de posición se pueden obtener mediante la integración de la velocidad angular que no es más que la derivada del ángulo en el tiempo.

Ecuación 6-2 Fórmula de obtención de ángulos de posición

$$\hat{\alpha}_n = \int v_{\alpha} \cdot dt + C$$

Siendo C la constante de integración y responsable de la presencia de errores en la medición que son de carácter acumulativo, por tal motivo se ha realizado la implementación del filtro AHRS de Sebastián Madgwick.

Los valores de actitud yaw, pitch y roll, aparte de ser representados gráficamente por medio de la aplicación en Processing, también se muestran numéricamente, sin embargo es en la representación gráfica en donde se puede encontrar infinidad de aplicaciones, pues permiten el monitoreo de la posición de los cuerpos, ya sean aviones, drones, vehículos terrestres o cualquier objeto que requiera un monitoreo de posición continuo.

Finalmente, en la aplicación se logra observar los valores fruto de la fusión sensorial del módulo GPS NEO 6M y la IMU MPU 9250, los valores de latitud y longitud son los que indican la ubicación del dispositivo en el planeta con una precisión aumentada en comparación con la latitud y longitud que proporcionaría únicamente el módulo GPS, tal afirmación es comprobada por medio de experimentación en el capítulo tres del presente proyecto, cumpliendo así con los objetivos planteados.

2.3. Interfaz gráfica LJLAB

Con el propósito de visualizar los datos obtenidos por parte de los dispositivos GPS y la IMU 9250 se procede a desarrollar una interfaz gráfica que contenga la información sobre los parámetros del acelerómetro, giroscopio y magnetómetro que conforman el sistema de navegación inercial además de los datos de latitud y longitud por parte del GPS NEO 6M. En el caso de los datos de latitud y longitud se presenta los valores entregados por el GPS y la integración del sistema GPS / INS en coordenadas geométricas. Además se busca una interfaz que permita ver los cambios de movimiento del prototipo por lo cual se mantiene el cubo en representación al dispositivo propuesto.

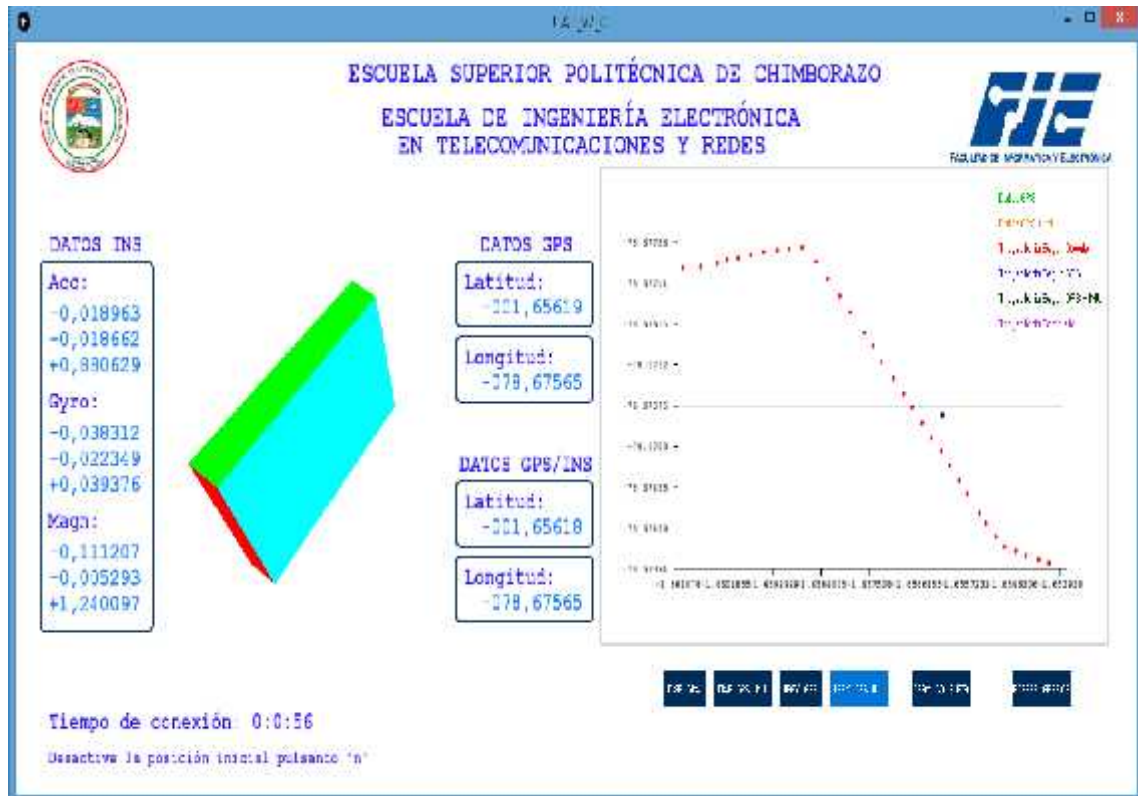


Figura 31-2. Interfaz LLAB

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

2.4. Propuesta de mejora del dispositivo - soporte de conexión inalámbrica

La conexión inalámbrica es un aspecto de gran ayuda en sistemas de posicionamiento y navegación inercial, esto se debe a las aplicaciones que estos sistemas poseen, siendo la mayoría para el monitoreo de posicionamiento de vehículos tanto aéreos como terrestres. En tal virtud en el presente proyecto se especifica los requerimientos y procedimientos necesarios para llevar a cabo este cometido.

El dispositivo seleccionado fue el XBee Serie1, por las ventajas que se detallaron frente a los distintos módulos de conexión inalámbrica existentes.

Se analizaron las programaciones posibles para la puesta en marcha de los Xbee, no obstante es necesario desarrollar la codificación que se aplique a la aplicación ya desarrollada y aplicada por el proyecto para el cumplimiento de los objetivos propuestos. Por tal motivo es necesario nuevamente realizar algunas modificaciones a la programación establecida en base la versión FreeIMU de Fabio Veresano.

2.3.1 Codificación Arduino IDE

La primera codificación que se modificó es la de la tarjeta controladora, es decir la programación desarrollada por medio del software Arduino IDE. A continuación se muestran las líneas de código añadidas seguido de la respectiva explicación de las mismas.

```
#define HAS_telem 1
#define telemSerial Serial2
#define telemBaud 57600
uint8_t count = 64;
#if HAS_telem
#include <XBee.h>
XBee xbee = XBee();
XBeeResponse response = XBeeResponse();
ZBRxResponse rx = ZBRxResponse();
XBeeAddress64 Broadcast = XBeeAddress64(0x0013A200, 0x40C04290);
char Message[230], charBuf[2];
#endif
char tempCorr;
char cmd1[9];
String cmd, Payload;
void setup() {
Serial.begin(BaudRate);
Wire.begin();
#if HAS_telem
telemSerial.begin(telemBaud);
xbee.setSerial(telemSerial);
xbee.setAPImode(1);
#endif
#if HAS_telem
xbee.readPacket();
if (xbee.getResponse().isAvailable()) {
cmd1[0] = '\0';
getMessage();
} else if (xbee.getResponse().isError()) {
}
```

```

#else
if(Serial.available()) {
cmd1[0] = '\0';
cmd1[0] = Serial.read(); }
#endif

#if HAS_telem
Message[0] = '\0';
XBeeSerialPrintFloatArr(val_array,20);
#else
serialPrintFloatArr(val_array,19);
#endif

#if HAS_telem
XBeeSerialPrintFloatArr(val_array,6);
#else
serialPrintFloatArr(val_array,6);
Serial.print('\n');
#endif
smartDelay(20);
#else
#if HAS_telem
strcat(Message, "\n");
#else
Serial.print('\n');
#endif
#endif

#if HAS_telem
sendPayload();
Serial.print(Message);
#endif
cmd1[0] = '\0';
char serial_busy_wait() {
#if !HAS_telem
while(!Serial.available()) {
return Serial.read();
#else
delay(1);

```

```

#endif
#if HAS_telem
void XBeeSerialPrintFloatArr(float * arr, int len) {
for(uint8_t i=0; i<len; i++) {
XBeeSerialFloatPrint(arr[i]);
strcat(Message, ",");
void XBeeSerialFloatPrint(float f) {
byte * b = (byte *) &f;
for(uint8_t i=0; i<4; i++) {
byte b1 = (b[i] >> 4) & 0x0f;
byte b2 = (b[i] & 0x0f);
char c1 = (b1 < 10) ? ('0' + b1) : 'A' + b1 - 10;
char c2 = (b2 < 10) ? ('0' + b2) : 'A' + b2 - 10;
String(c1).toCharArray(charBuf, 2); strcat(Message, charBuf);
String(c2).toCharArray(charBuf, 2); strcat(Message, charBuf);
void getMessage(){
//Serial.println(xbee.getResponse().getApiId(), HEX);
if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
xbee.getResponse().getZBRxResponse(rx);
if (rx.getOption() & ZB_PACKET_ACKNOWLEDGED)
//the sender got an ACK
//Serial.println("packet acknowledged");
cmd = "";
cmd1[0] = '\0';
for (int i = 0; i < rx.getDataLength(); i++) {
//print8Bits(rx.getData()[i]);
cmd1[i] = (char) rx.getData()[i];
cmd += cmd1[i];
void sendPayload(){
ZBTxRequest zbtx = ZBTxRequest(Broadcast, (uint8_t *) Message, strlen(Message));
xbee.send(zbtx);
#endif

```

La codificación expuesta hace referencia únicamente a las líneas de código añadidas a la programación original en cuanto al funcionamiento del Xbee, la codificación completa de la

programación de Arduino como las programaciones de las bibliotecas se muestran posteriormente en los anexos del proyecto.

Tal como es posible apreciar en las líneas de código, de inicio se define la variable que permitirá al programa identificar si el dispositivo Xbee se encuentra conectado, la línea utilizada para este fin es “#define HAS_telem 1” de esta manera es posible especificar manualmente si se cuenta o no con el dispositivo.

Los puertos de conexión se definieron en el Serial 2, que corresponden a los pines 16 y 17 del Arduino Mega 2560. Importante tener en cuenta que en la conexión del Xbee con el Arduino, se conectan los puertos de transmisión con los de recepción, es decir que el RX del Xbee irá en el puerto 16 y el TX en el puerto 17.

El Baudrate para el Xbee ha sido configurado en 56700 Baudios que es la misma cantidad definida para la placa en general.

La biblioteca añadida para el control del módulo Xbee es la denominada Xbee.h, la programación de esta librería se detalla en los anexos del proyecto.

La dirección de comunicación se establece en la línea “XBeeAddress64 Broadcast = XBeeAddress64(0x0013A200, 0x40C04290);” cabe destacar que esta dirección es exclusiva para cada módulo Xbee y se encuentra grabada en la parte posterior del mismo, la dirección detallada en esta línea de código corresponde a la dirección del módulo que estará conectado al computador, es decir al que hará la función de receptor.

Finalmente otro detalle que se especifica en la codificación por medio de la línea de código “xbee.setAPImode(1);” es el modo API, mismo que se seleccionó para la comunicación entre el módulo transmisor y receptor por las ventajas que posee en cuanto a sus funcionalidades frente al modo transparente que no permitiría la transmisión de datos analógicos tan complejos de una forma ordenada tal como se requiere.

2.3.2. Configuración de la aplicación en Processing

Como primer paso es necesario añadir las bibliotecas requeridas por la aplicación para su funcionamiento, aparte de las que se necesitó instalar antes de implementar la conexión inalámbrica, adicionalmente se instaló la biblioteca “XbeeJavaLibrary-1.1.0” misma que se encuentra en la red para descarga libre.

En la carpeta “code” de la aplicación en Processing, también se requiere añadir las bibliotecas detalladas a continuación:

- log4j.jar
- RXTXcomm.jar
- slf4j-api-1.7.7.jar
- slf4j-jdk14-1.7.12.jar
- xbee-api-0.9.jar
- xbjlib-1.0.1.jar

Cada archivo de la lista es posible descargarlo de manera libre en la red por lo que no presenta dificultad alguna su obtención.

El siguiente paso fue la modificación de la codificación en la aplicación en Processing, a continuación se muestran únicamente las líneas de código añadidas para el reconocimiento del módulo Xbee mas la programación completa se detalla en los anexos del presente proyecto.

```
int HAS_Xbee = 1;
String XbeePort = "COM6";
String DATA_TO_SEND = "z";
String REMOTE_NODE_IDENTIFIER = "END_ROUTER_1";
int BAUD_RATE = 57600;
XBeeDevice myDevice = new XBeeDevice(XbeePort, BAUD_RATE);
byte[] dataToSend = DATA_TO_SEND.getBytes();
XBeeNetwork xbeeNetwork;
RemoteXBeeDevice remoteDevice;
```

```

int packetCnt = 0;
String [] inputStringArr;
try {
myDevice.open();
xbeeNetwork = myDevice.getNetwork();
remoteDevice = xbeeNetwork.discoverDevice(REMOTE_NODE_IDENTIFIER);
if (remoteDevice == null) {
System.out.println("Couldn't find the remote XBee device with " +
REMOTE_NODE_IDENTIFIER + " Node Identifier.");
System.exit(1);
myDevice.addDataListener(new MyDataReceiveListener());
void writePacket() {
try {
myDevice.sendData(remoteDevice, dataToSend);
} catch (XBeeException e) {
System.out.println("Error");
e.printStackTrace();
System.exit(1);
void decodePacket() {
if (inputStringArr != null) {
if(inputStringArr.length >= 18) { // q1,q2,q3,q4,\r\n so we have 5 elements
q[0] = decodeFloat(inputStringArr[0]);
q[1] = decodeFloat(inputStringArr[1]);
q[2] = decodeFloat(inputStringArr[2]);
q[3] = decodeFloat(inputStringArr[3]);
acc[0] = decodeFloat(inputStringArr[4]);
acc[1] = decodeFloat(inputStringArr[5]);
acc[2] = decodeFloat(inputStringArr[6]);
gyro[0] = decodeFloat(inputStringArr[7]);
gyro[1] = decodeFloat(inputStringArr[8]);
gyro[2] = decodeFloat(inputStringArr[9]);
magn[0] = decodeFloat(inputStringArr[10]);
magn[1] = decodeFloat(inputStringArr[11]);
magn[2] = decodeFloat(inputStringArr[12]);
temp = decodeFloat(inputStringArr[13]);
press = decodeFloat(inputStringArr[14]);

```

```

//dt = (1./decodeFloat(inputStringArr[15]))/4;
dt = (1./decodeFloat(inputStringArr[15]));
heading = decodeFloat(inputStringArr[16]);
//dt = tnew - told;
//told = tnew;
if(heading < -9990) {
heading = 0;
altitude = decodeFloat(inputStringArr[17]);
//motionDetect = decodeFloat(inputStringArr[18]);
if(HAS_GPS == 1){
hdop = decodeFloat(inputStringArr[19]);
lat = decodeFloat(inputStringArr[20]);
longt = decodeFloat(inputStringArr[21]);
gpsalt = decodeFloat(inputStringArr[23]);
cog = decodeFloat(inputStringArr[24]);
sog = decodeFloat(inputStringArr[25]);

```

Es importante tener en cuenta que el puerto especificado en esta codificación varía conforme al computador en el que se conecte, por lo que deberá configurarse para poder establecer la comunicación.

En esta ocasión, se muestran las líneas de código añadidas, no obstante, no fue suficiente simplemente con añadir estas líneas a la programación, sino también se requirió anular algunas líneas de código de la programación sin soporte para conexión inalámbrica, por este motivo se sugiere revisar el detalle de la codificación completa presente en los anexos del proyecto.

Tal como se muestra en la programación, específicamente en la línea de código (`String REMOTE_NODE_IDENTIFIER = "END_ROUTER_1";`). Se definió al identificador de nodo con la denominación: "END_ROUTER_1" por lo cual es necesario configurar al Xbee transmisor con la misma denominación para su identificador de nodo. Para lograr este cometido fue necesario seguir el procedimiento detallado a continuación

2.3.3. Configuración de los módulos Xbee Serie 1

La configuración de los módulos se realizó por medio del software DIGI XCTU, en el mismo se estableció tanto el canal como la ID por medio de la cual se comunicarán los módulos entre sí, tal como se muestra en la Figura 31-2.

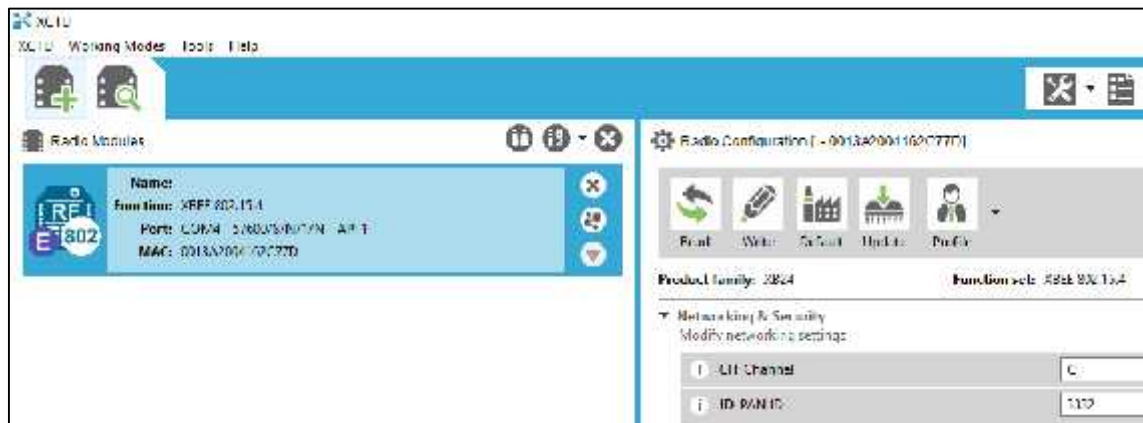


Figura 32-2. XCTU Configuración de canal e ID de comunicación

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

De igual manera en cuanto al protocolo de comunicación, el cual como se mencionó anteriormente el seleccionado fue el modo API 1, tal como se muestra en la Figura 32-2.

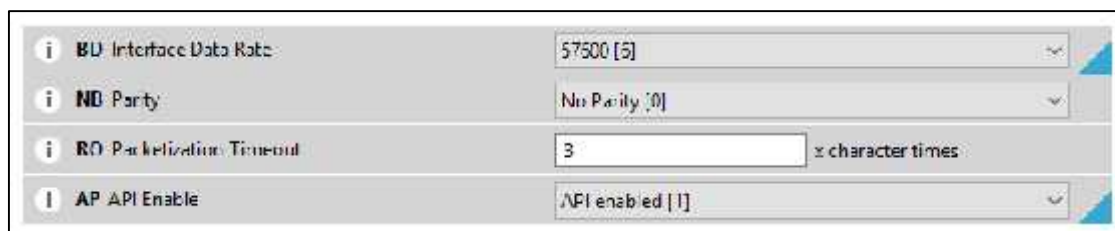


Figura 33-2. XCTU Configuración del protocolo de comunicación

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Y Finalmente la configuración del identificador de nodo se realiza en la opción con la abreviación “NI”, como se aprecia en la Figura 33-2.

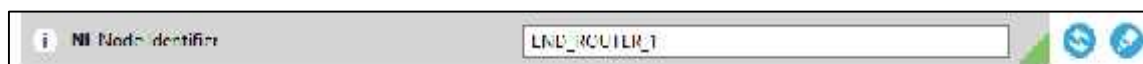


Figura 34-2. Gráficas de calibración.

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Con este último procedimiento realizado, se hace posible la ejecución de la aplicación en Processing y la obtención de datos de los sensores de manera inalámbrica.

Para el prototipo propuesto también se ha considerado una shield que permita mantener todos los pines de placa electrónica disponibles y a su vez suministre energía a los dispositivos que lo requieran como es el caso del energy shield 2 que contiene en su interior una batería de lipo de 1800 mAh. Es capaz de proporcionar salidas de 3,3 y 5 voltios, además cuenta con una entrada USB para su recarga con esto se evita el uso de dispositivos externos. En lo particular se adapta con normalidad a las placas Arduino UNO y MEGA con las cuales hemos trabajado como se puede observar en la siguiente figura.



Figura 35-2. EnergyShield2
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

A continuación se presenta el prototipo propuesto:

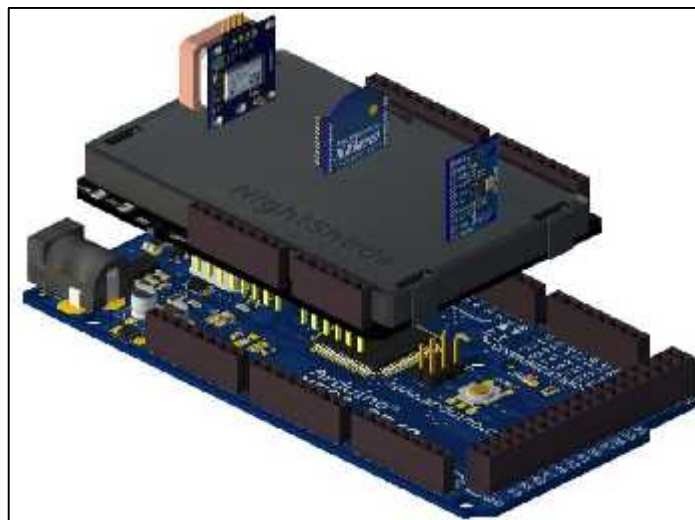


Figura 36-2. Propuesta de dispositivo
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

CAPITULO III

3. RESULTADOS

Como principal resultado obtenido del desarrollo del proyecto se tiene la fusión sensorial entre los dispositivos GPS NEO 6M e IMU 9250, cuyo principal objetivo es mejorar la precisión de posicionamiento.

3.1. Precisión y fusión sensorial

El método elegido por el proyecto para mejorar la precisión de posicionamiento fue la fusión sensorial, esto se debe a que la principal causa de error en las mediciones es la falta de variables de entrada, es decir que se toman datos tanto ineficientes como incompletos. En tal virtud un método para mejorar la precisión de las mediciones es por medio del aumento de las fuentes de información. Aplicando este concepto obvio en el caso de los sistemas globales de posicionamiento se pudo analizar las fuentes actuales de información para con ello seleccionar nuevas fuentes que complementen con las primeras.

Como es conocido, la fuente actual de información de un dispositivo GPS es obtenido a través de comunicación satelital, y es en este punto en donde la información adquirida presenta sus deficiencias por varios motivos relacionados a la interrupción de la comunicación entre dispositivos, como se sabe los principales factores que pueden afectar la transmisión de cualquier señal son:

- Atenuación. - Producida en señales digitales o analógicas, siendo un efecto natural en la transmisión de las señales a través de distancias elevadas.
- Ruido. - Cualquier fuente de energía, magnética, electromagnética, eléctrica, que pueda alterar las señales de comunicación inalámbrica.
- Absorción atmosférica. - Obstáculos que disminuye la intensidad de la señal.
- Refracción. - Inclinación de las microondas por causa de la atmósfera.
- Multitrayecto. - Reflexión de la señal por causa de obstáculos.

Al existir tantos factores que pueden afectar la calidad de la comunicación entre el GPS y el satélite, es evidente que la comunicación entre ambos no es confiable, conllevando con ello a la disminución de la precisión obtenida.

Con el aumento de las fuentes de información es posible reducir esta deficiencia identificada y la IMU MPU 9250 es una fuente variada de información, como se explicó anteriormente, la IMU es capaz de tomar mediciones con 9 grados de libertad provenientes del acelerómetro, giroscopio y magnetómetro. La información recopilada de estos tres sensores que integra la IMU permite monitoriar en todo momento los movimientos realizados por el dispositivo ya sean movimientos lineales o angulares. Lo cual es el complemento perfecto para el sistema de posicionamiento global puesto a que con ello se puede suplir las deficiencias producidas como fruto de la mala comunicación satelital.

Con la fusión del GPS y la IMU se consiguió un dispositivo capaz de seguir evaluando movimientos lineales y angulares con o sin la presencia de comunicación satelital, comparando en todo momento los datos obtenidos tanto del GPS como de la IMU para proporcionar resultados con el menor error posible.

3.2. Filtros (Madgwick y Kalman)

El proyecto dio como resultado un dispositivo que en su fusión sensorial hace uso de dos filtros, en primer lugar se hizo uso del filtro AHRS de Sebastián Madgwick y después del filtro de Kalman. Sin embargo, ¿qué se obtuvo del uso de los filtros?

Es necesario entender el funcionamiento de la IMU. Tal como se ha explicado anteriormente, la unidad de medición inercial está compuesta por tres sensores:

- Acelerómetro
- Giroscopio
- Magnetómetro

No obstante la información que cada sensor provee no es precisamente la velocidad a la que el dispositivo se mueve, ni la dirección o sentido del movimiento. La señal de entrada percibida por la tarjeta ARDUINO MEGA es eléctrica, misma que varía en función del movimiento al que se

somete la IMU. Por tanto en la programación de la tarjeta que se puede apreciar en el Anexo E. Se indica a la tarjeta la forma de interpretar estos resultados para convertir la señal eléctrica en datos numéricos, no obstante estos datos muestran movimientos de cambio y no de posicionamiento, es decir que indican la cantidad que se movió la IMU, ya sea en velocidad lineal o angular mas no la posición en la que se encuentra en dirección y sentido, que corresponde a los ángulos de yaw, pitch y roll que son los valores que se necesitan para saber la dirección y sentido en el que el dispositivo se mueve.

Para la obtención de yaw pitch y roll se hizo uso de los tres sensores, puesto que con mayores fuentes de obtención de datos es posible obtener datos mayormente precisos, sin embargo es necesario analizar la ecuación que conlleva a transformar las mediciones de cambio angular, en mediciones de yaw pitch y roll por parte del giroscopio. Como ya se expuso en el capítulo anterior, la ecuación es:

$$\text{Áng} = \int \dot{a} \cdot dt + C$$

Siendo C la constante de integración y responsable de los errores en la medición que son de carácter acumulativo, por tal motivo se implementó el filtro AHRS de Sebastián Madgwick. Los resultados obtenidos se mostraron de igual manera en la Figura 24-2. Misma que muestra que el uso del filtro logró eliminar el cúmulo de errores, al lograr una estabilización de los datos.

En resumen con el filtro de Madgwick se logró reducir significativamente el error producido por la constante C de la ecuación de obtención de ángulos.

Finalmente al obtener las mediciones de yaw pitch y roll necesarias para la fusión de datos por parte de GPS e IMU, se hizo uso del filtro de Kalman que consiguió la anhelada fusión sensorial. Las ecuaciones base del filtro ya establecidas se basan en la predicción en función de la información recibida, y mediante la programación desarrollada por el proyecto, se consigue la implementación del filtro al procesamiento de datos efectuado por Arduino. Programación que se explicó en el capítulo anterior y se muestra completa en el Anexo E. Con ello se consiguió obtener los valores de las coordenadas de posicionamiento en función de ambos dispositivos.

3.3.1. Arquitectura de acoplamiento

Como resultado final del uso de los dos filtros, se obtiene la siguiente arquitectura de acoplamiento:

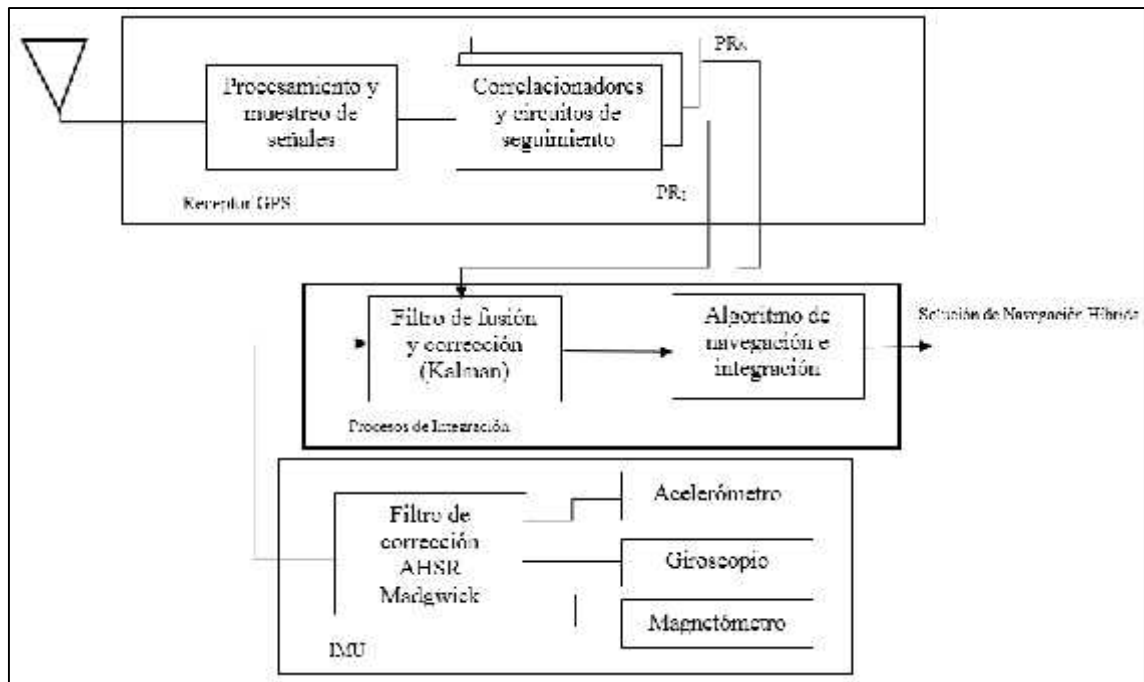


Figura 1-3. Tight coupling modificado
Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Como se puede apreciar en la Figura 1-3. La arquitectura no hace uso del procesador interno de la IMU MPU 9250. En lugar de ello emplea el filtro AHRS de Madgwick.

3.3. Validación de resultados mediante métodos estadísticos

En el presente capítulo se evidencian los resultados obtenidos por medio de la integración del GPS NEO 6M y la IMU MPU 9250, la metodología utilizada para hacer una comparación entre los métodos se efectuará por medio de análisis estadísticos y diseño experimental, tomando en cuenta que de los dos métodos evaluados, el método uno o inicial refiere a la precisión que posee el dispositivo GPS de manera individual y el método dos o propuesto refiere a la precisión obtenida mediante la fusión sensorial entre el dispositivo GPS y la unidad de medición inercial.

Por medio del presente análisis estadístico se pretende obtener una medida cuantificable que demuestre que mediante el método propuesto es posible adquirir datos más precisos.

Para la obtención de la información necesaria para llevar a cabo la comparación entre los métodos inicial y propuesto, se planteó establecer un recorrido a través de las instalaciones de la Escuela Superior Politécnica de Chimborazo, recorrido por el cual se registró las coordenadas mostradas por el dispositivo GPS para el método inicial y las coordenadas producto de la integración del NEO 6M y la IMU MPU 9250 para el método propuesto.

3.4. Recorrido a través de la ESPOCH

Se seleccionó la ruta por la cual se efectuó la toma de mediciones, la ruta seleccionada contiene puntos específicos que representan el lugar en el cual se registraron las coordenadas proporcionadas por el dispositivo. A continuación en la Figura 1-3. Se muestra por medio de la aplicación Google Maps, la ruta y puntos especificados.



Figura 2-3. Trayectoria y coordenadas de puntos evaluados dentro de la ESPOCH

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Tal como puede apreciarse en la Figura 1-3. Se seleccionó la trayectoria que une las dos puertas de acceso de la ESPOCH y se la dividió en 35 puntos en los cuales se procedió a registrar las coordenadas que el dispositivo proporcionó, tanto para el método inicial como para el propuesto.

3.5. Comparación de resultados mediante diseño experimental

El diseño experimental es una herramienta científica muy poderosa de la estadística, la cual consigue analizar los resultados de metodologías diferentes, en función de factores de variabilidad

para con ello determinar la influencia de dichos factores en las variables respuesta. Por tanto se hizo uso de esta técnica para evaluar nuestros resultados y comprobar la influencia de los métodos de medición en la precisión de posicionamiento.

El primer paso en la evaluación de resultados es determinar si existe o no diferencia significativa entre las dos metodologías, refiriéndose como método inicial al dispositivo que hace uso solamente del GPS NEO 6M y como método propuesto al dispositivo conformado por la integración sensorial del GPS NEO 6M con la IMU MPU 9250. En tal virtud, se ha considerado hacer uso de las herramientas de diseño experimental para llevar a cabo este cometido.

El experimento diseñado considera obtener el error existente por parte de las mediciones proporcionadas por los dos métodos para posteriormente llevar a cabo el análisis mediante el uso del ANOVA (Análisis de varianza), con ello se podrá determinar si existe o no diferencia significativa entre los resultados de ambas metodologías.

Por tanto, es necesario adquirir el error absoluto existente entre las mediciones, que como se sabe resulta de la diferencia entre el valor real y el valor medido. El valor real se lo obtuvo mediante la aplicación Google Maps, en la cual es posible adquirir los valores reales de longitud y latitud para cada uno de los 35 puntos especificados en la Figura 1-3.

A continuación en la Tabla 1-3. Se puede apreciar las coordenadas proporcionadas por Google Maps para cada uno de los 35 puntos seleccionados anteriormente.

Tabla 1-3. Coordenadas de referencia de la ruta

Nº	Coordenadas
1	-1.661028, -78.677287
2	-1.660694, -78.677292
3	-1.660387, -78.677333
4	-1.660205, -78.677366
5	-1.659992, -78.677392
6	-1.659763, -78.677419
7	-1.659523, -78.677452
8	-1.659268, -78.677469
9	-1.659045, -78.677485
10	-1.658799, -78.677505
11	-1.658537, -78.677347
12	-1.658321, -78.677165

Tabla1-3. (Continuación)

13	-1.658098, -78.676962
14	-1.657898, -78.676792
15	-1.657634, -78.676562
16	-1.657481, -78.676407
17	-1.657320, -78.676229
18	-1.657100, -78.676058
19	-1.656907, -78.675884
20	-1.656745, -78.675735
21	-1.656560, -78.675569
22	-1.656388, -78.675401
23	-1.656212, -78.675247
24	-1.656040, -78.675090
25	-1.655872, -78.674920
26	-1.655711, -78.674774
27	-1.655490, -78.674570
28	-1.655372, -78.674455
29	-1.655187, -78.674300
30	-1.655021, -78.674190
31	-1.654813, -78.674142
32	-1.654626, -78.674088
33	-1.654386, -78.674051
34	-1.654187, -78.674013
35	-1.653988, -78.673993

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Partiendo de las coordenadas de los puntos evaluados, se procede a diseñar el tipo de experimento a llevar a cabo, el cual como ya se mencionó, tiene el objetivo de evidenciar de manera cuantificable la diferencia existente entre la metodología inicial que hace uso únicamente del GPS y la metodología propuesta que hace uso de la fusión sensorial fruto del presente proyecto.

Para la planeación y posterior realización de cualquier diseño experimental, es necesario cumplir con seis pasos básicos que permitirán obtener resultados confiables en cuanto a la investigación, en tal virtud, a continuación se muestra el planteamiento del diseño experimental en torno a estos seis indispensables pasos.

1.- Entender y delimitar el problema u objeto de estudio. - Como se pudo explicar, el objetivo será determinar si existe diferencia significativa entre los resultados de los dos métodos

evaluados, lo cual conducirá a afirmar con exactitud cuál de los dos métodos posee mejor precisión en cuanto al posicionamiento.

2.- Elegir variables respuesta. - La variable elegida para el experimento parte del error absoluto entre las mediciones, es decir la distancia existente entre el valor real y el valor medido con cada uno de los dos métodos evaluados.

3.- Determinar los factores a investigar con respecto a la influencia con la variable respuesta.- No cabe duda que el principal factor a evaluar en este caso es el tipo de dispositivo utilizado para adquirir las coordenadas, no obstante hay que tomar en cuenta otro factor que sin duda alguna repercute en la variable respuesta, y es la velocidad en el que el dispositivo se mueve hasta el momento de hacer la lectura de las coordenadas, esta velocidad de movimiento afectará sin duda alguna los resultados del experimento.

4.- Selección del diseño experimental adecuado.- Como se explicó anteriormente, el principal factor es el tipo de dispositivo, no obstante se conoce que la velocidad del dispositivo al momento de tomar la medida también afectará los resultados, según los principios del diseño experimental, una opción para nulificar este “sub factor” sería tomar las mediciones en ambos métodos utilizando la misma velocidad, no obstante no es aconsejable debido a que pueda ser que la influencia de este factor nulificado sea tan grande que cause la adquisición de resultados erróneos del experimento, en tal virtud se optó por realizar un DBCA (Diseño de bloques completamente al azar), en el cual se evaluará como factor principal la influencia del uso del dispositivo, y como factor de bloque la velocidad en la toma de mediciones. En otras palabras, mientras los métodos de toma de datos (inicial y propuesto) son evaluados para determinar la mejor opción en términos de precisión, en cuanto a la velocidad, no se trata de elegir una, en todo caso puede darse una sugerencia de la velocidad que deberá utilizarse para la toma de datos con el fin de adquirir mayor eficiencia.

5.- Escenario: Plantear y organizar el trabajo experimental.- En cuanto a las dos metodologías evaluadas, tal como se indicó, el factor principal a evaluar es la influencia del dispositivo utilizado para la medición sobre la variable respuesta que será la distancia existente entre la coordenada real y la coordenada medida, el primer método utiliza el GPS NEO 6M y el segundo utiliza la fusión sensorial entre el GPS e IMU, El factor de bloque a evaluar es la velocidad del dispositivo en el momento de tomar la medición, para lo cual se experimentará con tres velocidades diferentes por cada método (40;160;280) (m/min).

3.6. Ejecución del experimento

En primer lugar se toman las mediciones con los dos métodos y a las 3 respectivas velocidades seleccionadas con cada método. A continuación en la Tabla 2-3 se muestran las coordenadas obtenidas.

Tabla 2-3. Coordenadas medidas

	(GPS)						(GPS+IMU)					
	40 m/min		160 m/min		280 m/min		40 m/min		160 m/min		280 m/min	
	Lat.	Long.	Lat.	Long.	Lat.	Long.	Lat.	Long.	Lat.	Long.	Lat.	Long.
	-	-	-	-	-	-	-	-	-	-	-	-
1	1,661 024	78,677 282	1,661 016	78,677 284	1,661 021	78,677 279	1,661 027	78,677 284	1,661 027	78,677 286	1,661 029	78,677 286
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,660 700	78,677 301	1,660 690	78,677 287	1,660 691	78,677 286	1,660 694	78,677 294	1,660 694	78,677 290	1,660 695	78,677 290
	-	-	-	-	-	-	-	-	-	-	-	-
3	1,660 381	78,677 329	1,660 386	78,677 328	1,660 382	78,677 328	1,660 386	78,677 332	1,660 388	78,677 333	1,660 384	78,677 330
	-	-	-	-	-	-	-	-	-	-	-	-
4	1,660 202	78,677 362	1,660 208	78,677 358	1,660 202	78,677 358	1,660 205	78,677 364	1,660 204	78,677 364	1,660 202	78,677 366
	-	-	-	-	-	-	-	-	-	-	-	-
5	1,659 982	78,677 384	1,659 990	78,677 388	1,659 986	78,677 387	1,659 992	78,677 394	1,659 990	78,677 390	1,659 992	78,677 389
	-	-	-	-	-	-	-	-	-	-	-	-
6	1,659 751	78,677 4147	1,659 758	78,677 418	1,659 756	78,677 414	1,659 760	78,677 419	1,659 760	78,677 419	1,659 761	78,677 419
	-	-	-	-	-	-	-	-	-	-	-	-
7	1,659 516	78,677 451	1,659 519	78,677 445	1,659 521	78,677 447	1,659 522	78,677 449	1,659 521	78,677 446	1,659 522	78,677 452
	-	-	-	-	-	-	-	-	-	-	-	-
8	1,659 261	78,677 468	1,659 265	78,677 461	1,659 263	78,677 468	1,659 266	78,677 468	1,659 266	78,677 468	1,659 269	78,677 468
	-	-	-	-	-	-	-	-	-	-	-	-
9	1,659 038	78,677 484	1,659 042	78,677 482	1,659 041	78,677 479	1,659 046	78,677 481	1,659 042	78,677 485	1,659 044	78,677 486
	-	-	-	-	-	-	-	-	-	-	-	-
10	1,658 803	78,677 507	1,658 792	78,677 503	1,658 798	78,677 496	1,658 798	78,677 503	1,658 798	78,677 505	1,658 798	78,677 502
	-	-	-	-	-	-	-	-	-	-	-	-
11	1,658 536	78,677 345	1,658 534	78,677 351	1,658 533	78,677 343	1,658 535	78,677 345	1,658 534	78,677 345	1,658 534	78,677 347
	-	-	-	-	-	-	-	-	-	-	-	-
12	1,658 320	78,677 1568	1,658 315	78,677 158	1,658 312	78,677 163	1,658 320	78,677 164	1,658 323	78,677 165	1,658 318	78,677 164
	-	-	-	-	-	-	-	-	-	-	-	-
13	1,658 092	78,676 961	1,658 095	78,676 954	1,658 095	78,676 961	1,658 099	78,676 964	1,658 100	78,676 963	1,658 099	78,676 963
	-	-	-	-	-	-	-	-	-	-	-	-
14	1,657 897	78,676 7861	1,657 895	78,676 784	1,657 889	78,676 790	1,657 897	78,676 790	1,657 897	78,676 793	1,657 895	78,676 789

Tabla 2-3. (Continuación)

	-	-	-	-	-	-	-	-	-	-	-	-
1	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676
5	633	5506	625	559	632	557	635	562	633	560	630	562
	-	-	-	-	-	-	-	-	-	-	-	-
1	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676
6	480	3978	476	402	477	406	478	405	480	404	479	410
	-	-	-	-	-	-	-	-	-	-	-	-
1	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676
7	319	2204	316	223	319	221	318	231	321	225	320	224
	-	-	-	-	-	-	-	-	-	-	-	-
1	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676	1,657	78,676
8	099	0502	096	056	097	059	100	057	099	062	101	055
	-	-	-	-	-	-	-	-	-	-	-	-
1	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675
9	906	8789	907	885	902	875	903	881	903	880	906	884
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675
0	744	7236	741	736	736	735	743	732	743	735	743	732
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675
1	559	5574	558	564	557	573	561	568	561	567	559	567
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675
2	387	3929	383	399	384	397	387	402	388	399	388	399
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675
3	211	2377	209	238	214	245	212	244	211	245	212	249
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675	1,656	78,675
4	039	0799	041	085	035	084	039	088	037	091	041	092
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674
5	871	9167	871	914	867	917	869	917	871	919	873	921
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674
6	710	7633	708	773	707	772	709	770	710	770	712	772
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674
7	489	5609	485	569	487	565	490	567	491	567	487	569
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674
8	371	4442	371	451	365	453	370	453	371	453	370	458
	-	-	-	-	-	-	-	-	-	-	-	-
2	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674
9	186	2926	178	294	188	295	186	299	188	297	189	299
	-	-	-	-	-	-	-	-	-	-	-	-
3	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674	1,655	78,674
0	020	1838	018	183	014	185	018	190	018	187	021	187
	-	-	-	-	-	-	-	-	-	-	-	-
3	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674
1	812	1296	814	141	815	138	812	140	811	141	809	141
	-	-	-	-	-	-	-	-	-	-	-	-
3	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674
2	625	0779	624	084	621	080	626	086	625	087	626	088
	-	-	-	-	-	-	-	-	-	-	-	-
3	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674
3	385	0401	384	047	384	049	386	054	386	049	387	049

Tabla 2-3. (Continuación)

3	-	-	-	-	-	-	-	-	-	-	-	-
4	1,654	78,673	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674	1,654	78,674
	186	998	177	011	182	013	186	015	185	012	186	011
3	-	-	-	-	-	-	-	-	-	-	-	-
5	1,653	78,673	1,653	78,673	1,653	78,673	1,653	78,673	1,653	78,673	1,653	78,673
	987	9881	979	990	984	987	988	992	987	993	987	993

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

En la Tabla 3-3. Se muestran los datos del ejercicio experimental con los cuales se parte para su análisis mediante la tabla ANOVA. En mencionada tabla, se muestra el error calculado en centímetros a partir de la distancia existente entre las coordenadas proporcionadas por los dos métodos y las coordenadas reales de cada uno de los 35 puntos mostrados en la Tabla 1-3.

Tabla 3-3. Error absoluto de posicionamiento en centímetros

N° Obs	(GPS)			(GPS+IMU)		
	Velocidades			Velocidades		
	40 m/min	160 m/min	280 m/min	40 m/min	160 m/min	280 m/min
1	76,3 (cm)	159,2(cm)	167,7(cm)	7,6(cm)	12,6(cm)	4,1(cm)
2	119,9(cm)	105,1(cm)	175,8(cm)	4,2(cm)	10,6(cm)	8,8(cm)
3	82,7(cm)	104,2(cm)	80,0(cm)	3,9(cm)	7,9(cm)	16,3(cm)
4	58,9(cm)	88,3(cm)	105,2(cm)	1,4(cm)	11,3(cm)	6,3(cm)
5	141,7(cm)	109,5(cm)	340,4(cm)	1,3(cm)	11,1(cm)	13,3(cm)
6	87,4(cm)	50,0(cm)	168,9(cm)	6,3(cm)	7,9(cm)	10,8(cm)
7	79,3(cm)	161,5(cm)	146,6(cm)	9,3(cm)	5,7(cm)	6,5(cm)
8	80,8(cm)	138,0(cm)	131,5(cm)	11,8(cm)	9,5(cm)	5,7(cm)
9	77,3(cm)	104,5(cm)	186,1(cm)	7,1(cm)	6,0(cm)	7,6(cm)
10	48,5(cm)	147,2(cm)	141,7(cm)	8,0(cm)	11,2(cm)	7,8(cm)
11	25,0(cm)	128,1(cm)	86,5(cm)	8,3(cm)	11,1(cm)	8,5(cm)
12	92,1(cm)	77,9(cm)	142,6(cm)	1,9(cm)	8,4(cm)	8,3(cm)
13	67,6(cm)	102,9(cm)	72,8(cm)	4,0(cm)	5,0(cm)	11,2(cm)
14	36,2(cm)	149,0(cm)	130,3(cm)	7,4(cm)	3,1(cm)	13,2(cm)
15	127,8(cm)	170,7(cm)	149,2(cm)	8,3(cm)	8,6(cm)	13,9(cm)
16	102,2(cm)	78,7(cm)	231,4(cm)	5,7(cm)	8,7(cm)	11,7(cm)
17	97,8(cm)	129,4(cm)	34,7(cm)	2,6(cm)	12,2(cm)	8,1(cm)
18	85,6(cm)	91,5(cm)	179,9(cm)	2,5(cm)	4,5(cm)	12,6(cm)
19	59,7(cm)	80,7(cm)	119,1(cm)	10,0(cm)	3,8(cm)	12,6(cm)
20	126,3(cm)	125,7(cm)	175,4(cm)	3,0(cm)	8,4(cm)	14,5(cm)
21	128,2(cm)	58,4(cm)	55,9(cm)	1,7(cm)	9,8(cm)	12,4(cm)
22	92,6(cm)	168,1(cm)	4,0(cm)	3,8(cm)	7,6(cm)	14,4(cm)
23	102,6(cm)	133,2(cm)	192,8(cm)	7,1(cm)	7,3(cm)	11,1(cm)
24	115,6(cm)	195,9(cm)	138,2(cm)	5,5(cm)	3,6(cm)	10,3(cm)
25	36,8(cm)	161,2(cm)	193,2(cm)	3,6(cm)	8,9(cm)	8,2(cm)
26	118,8(cm)	222,5(cm)	145,5(cm)	3,9(cm)	8,5(cm)	9,0(cm)

Tabla 3-3. (Continuación)

27	101,9(cm)	176,5(cm)	134,2(cm)	8,7(cm)	6,3(cm)	10,7(cm)
28	120,9(cm)	187,0(cm)	171,5(cm)	3,4(cm)	4,6(cm)	10,6(cm)
29	81,6(cm)	151,1(cm)	151,6(cm)	7,6(cm)	9,8(cm)	5,0(cm)
30	70,1(cm)	120,6(cm)	183,6(cm)	7,5(cm)	7,6(cm)	12,3(cm)
31	137,6(cm)	128,8(cm)	63,7(cm)	6,8(cm)	6,0(cm)	13,6(cm)
32	113,3(cm)	146,9(cm)	263,0(cm)	6,2(cm)	9,7(cm)	12,3(cm)
33	124,3(cm)	54,9(cm)	176,3(cm)	3,7(cm)	12,1(cm)	12,5(cm)
34	162,1(cm)	85,6(cm)	151,7(cm)	5,5(cm)	9,4(cm)	19,7(cm)
35	57,5(cm)	170,4(cm)	157,4(cm)	3,5(cm)	10,4(cm)	9,0(cm)

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Los valores de las distancias se calcularon mediante la ecuación:

Ecuación 1-3. Distancia entre dos puntos geográficos

$$d = (6371 * \text{acos}(\cos(\lambda_1 - \lambda_2) * \cos(90 - \phi_1) * \cos(90 - \phi_2) + \sin(90 - \phi_1) * \sin(90 - \phi_2) * \cos(\lambda_1 - \lambda_2))) + 100000$$

Después de obtener el error absoluto de cada una de las 35 observaciones, para los dos métodos y para los 3 factores de bloque, se procedió a calcular sus medias, mismas que se muestran a continuación en la Tabla 4-3.

Tabla 4-3. Error medio en centímetros del factor principal y factor de bloque

MÉTODO	VELOCIDAD		
	40 m/min	160 m/min	280 m/min
1 (GPS)	92,5	109,5	340,4
2 (GPS+IMU)	5,5	8,3	10,7

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Para realizar la comparación entre los dos métodos, es necesario plantear las siguientes hipótesis:

Hipótesis nula:

$$H_0: \mu_1 = \mu_2 = \mu$$

Hipótesis alternativa:

$$H_A: \mu_i \neq \mu_j \quad \text{a} \quad \text{ún } i \neq j$$

Es decir que la hipótesis nula plantea que no existe diferencia significativa entre los dos tratamientos, mientras que por otro lado la hipótesis alternativa plantea la existencia de una diferencia significativa. Afirmación establecida mediante la comparación de media y varianza.

Las hipótesis planteadas se ponen a prueba mediante el análisis de varianza, teniendo en cuenta que se manejan dos criterios de clasificación, siendo estos, el factor del tratamiento que es el principal a evaluar y el factor de bloque que permite eliminar una fuente de variación considerable y a su vez analizarlo para establecer el más eficaz.

A continuación en la Tabla 5-3. Se muestra la estructura que posee el ANOVA en un diseño DBCA (Diseño de Bloque Completamente al Azar)

Tabla 5-3. Estructura ANOVA para DBCA

Fuente de variabilidad	Suma de cuadrados	Grado de libertad	Cuadrado medio	F_0	Valor-p
Tratamientos	SC_{TRAT}	$k - 1$	CM_{TRAT}	$F_0 = \frac{CM_{TRAT}}{CM_E}$	$P(F > F_0)$
Bloques	SC_B	$b - 1$	CM_B	$F_0 = \frac{CM_B}{CM_E}$	$P(F > F_0)$
Error	SC_E	$(k - 1)(b - 1)$	CM_E		
Total	SC_T	$N - 1$			

Fuente: (Gutiérrez Pulido, 2008), p.104

Con el fin de poder obtener los datos mostrados en la Tabla 5-3, es necesario primero calcular la media global, y los totales tanto por tratamiento como por bloque, tales cálculos son ilustrados en la Tabla 6-3.

Tabla 6-3. Totales por tratamiento y por bloque

MÉTODO	VELOCIDAD			Total por tratamiento
	40 m/min	160 m/min	280 m/min	
1 (GPS)	92,5	127,5	146,9	$Y_{1\bullet} = 366,9$
2 (GPS+IMU)	5,5	8,3	10,7	$Y_{2\bullet} = 24,4$
Total	$Y_{\bullet 1} = 98,0$	$Y_{\bullet 2} = 135,8$	$Y_{\bullet 3} = 157,5$	Total global $Y_{\bullet\bullet} = 391,3$

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Con estos resultados se procede a obtener la suma de cuadrados.

Ecuación 2-3. Suma de cuadrados totales

$$S_T = \sum_{j=1}^b \sum_{i=1}^k Y_{ij}^2 - \frac{Y_{..}^2}{N}$$

$$S_T = (92,5^2 + 5,5^2 + 127,5^2 + 8,3^2 + 146,9^2 + 10,7^2) - \frac{391,3^2}{6}$$

$$S_T = 46605,74 - 25532,32$$

$$S_T = 21073,4$$

Ecuación 3-3. Suma de cuadrados del tratamiento

$$S_T = \sum_{i=1}^k \frac{Y_{i.}^2}{b} - \frac{Y_{..}^2}{N}$$

$$S_T = \frac{366,9^2 + 24,4^2}{2} - \frac{391,3^2}{6}$$

$$S_T = 45071,95 - 25532,32$$

$$S_T = 19539,6$$

Ecuación 4-3. Suma de cuadrados del bloque

$$S_B = \sum_{j=1}^b \frac{Y_{.j}^2}{k} - \frac{Y_{..}^2}{N}$$

$$S_B = \frac{98^2 + 135,8^2 + 157,5^2}{3} - \frac{391,3^2}{6}$$

$$S_B = 26441,7 - 25532,32$$

$$S_B = 909,4$$

Ecuación 5-3. Suma de cuadrados del error

$$S_E = S_T - S_T - S_B$$

$$S_E = 21073,4 - 19539,6 - 909,4$$

$$S_E = 624,4$$

Los grados de libertad de la suma de cuadrados totales, corresponde a N-1 (número de observaciones menos uno), de igual manera con los grados de libertad del tratamiento, corresponde al número de tratamientos menos uno, y los grados de libertad del bloque corresponde al número de velocidades evaluadas menos uno, es decir 1 y 2 respectivamente.

Tabla 7-3. ANOVA

Fuente de variabilidad	Suma de cuadrados	Grados de libertad	Cuadrado medio	F (Estadístico de prueba)	F crítico
Métodos	19539,6	1	19539,6267	62,5855523	18,51
Velocidades	909,4	2	454,686667	1,45636437	19
Error	624,4	2	312,206667		
Total	21073,4	5			

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

3.7. Interpretación de resultados

Finalmente, por medio de la Tabla 7-3. Es posible rechazar la hipótesis nula y aceptar la hipótesis alternativa que afirma que existe diferencia significativa entre los dos métodos, dado que el estadístico de prueba F (calculado al dividir el cuadrado medio tanto del método como del bloque para el cuadrado medio del error) es mayor que el F crítico obtenido de tablas (tabla F de Fisher).

En cuanto al factor de bloque evaluado, que es la velocidad, la tabla ANOVA sugiere que este factor no influye significativamente en la variable respuesta, o al menos no tanto como si lo hace el método de medición.

Esto quiere decir que el método utilizado para obtener el posicionamiento influye fuertemente en la variable respuesta, que en el caso de nuestro experimento, es la distancia entre el punto real y el punto registrado por el dispositivo, mismo que se ha especificado como error absoluto. No obstante, se sabe que hay gran diferencia entre el método que utiliza solo el GPS con el que hace uso del GPS más IMU, pero ¿Cuál de los dos es el mejor? La respuesta a esta interrogante puede ser fácilmente encontrada por medio de una comparación entre medias del experimento. Mientras que el método inicial que hace uso solamente del GPS registró una media de 122,3 cm que representa el error en cuanto al posicionamiento, el método propuesto como fruto del presente proyecto registró una media de 8,17cm. Con lo cual se puede afirmar con certeza es el mejor método.

Debido a que existe variabilidad en cuanto al error que los dispositivos pueden presentar en cada medición para ambos métodos, no es posible fijar un porcentaje exacto de reducción del error, sin embargo mediante una comparación entre medias, se hace posible conseguir una estimación del mencionado porcentaje. Con lo cual se llega a la conclusión que por medio de la fusión sensorial propuesta es posible reducir el error de posicionamiento aproximadamente en un 93,3%, lo cual representa una mejora altamente significativa.

A continuación mediante las siguientes gráficas se hacen evidentes visualmente los resultados obtenidos, en ellas se grafica la dispersión de puntos para los dos métodos y tres velocidades por cada método. Las distancias de cada eje X, Y se obtuvieron mediante la Ecuación 1-3, adoptando un valor en la longitud dos de cero para calcular la latitud y adoptando un valor de cero en la latitud dos para obtener la longitud.

Respecto a las gráficas de comparación de dispersión de puntos en la trayectoria se visualizó los diez primeros puntos de la trayectoria para observar la ruta de referencia y las rutas elaboradas a partir de los datos de GPS y GPS + IMU con el propósito de verificar que existe menor error con la arquitectura de Z estrecho.

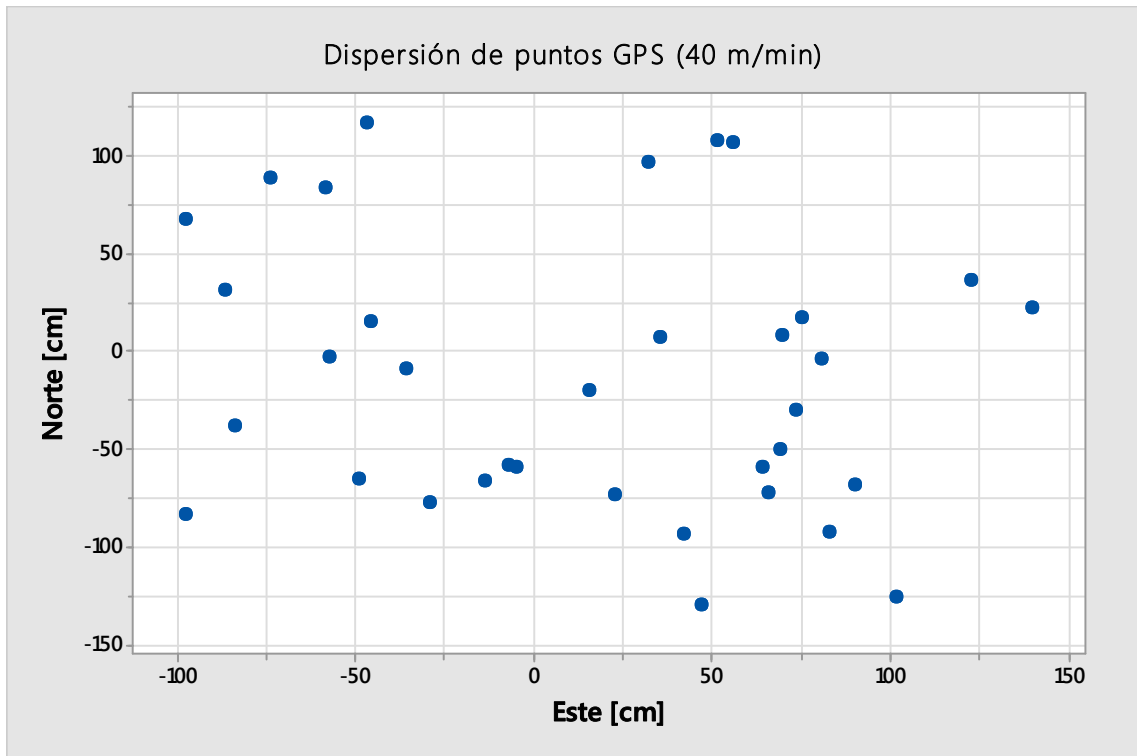


Gráfico 1-3. Dispersión de puntos GPS (40 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

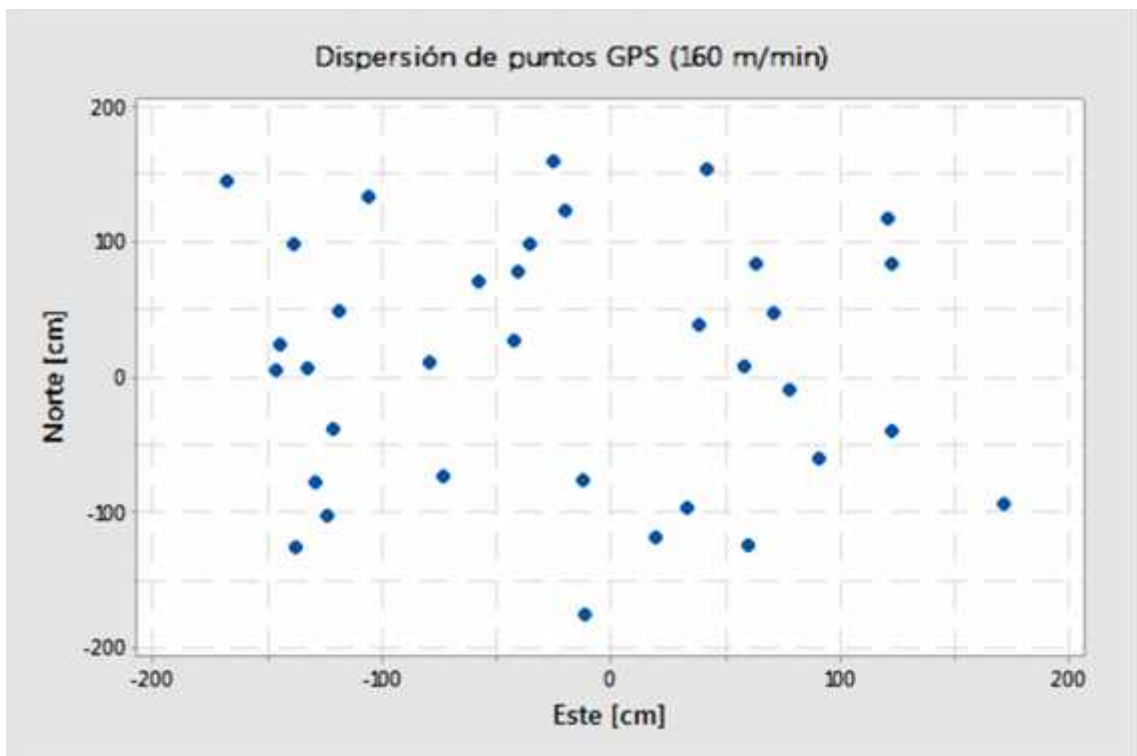


Gráfico 2-3. Dispersión de puntos GPS (160 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

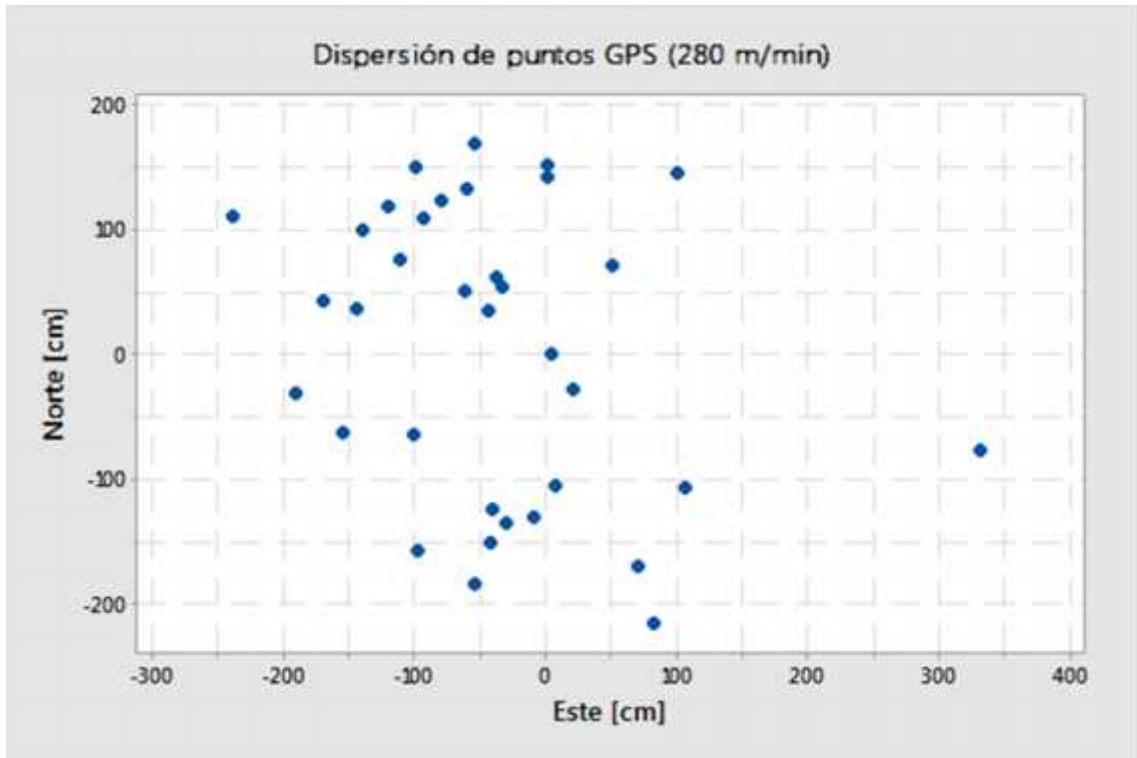


Gráfico 3-3. Dispersión de puntos GPS (280 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

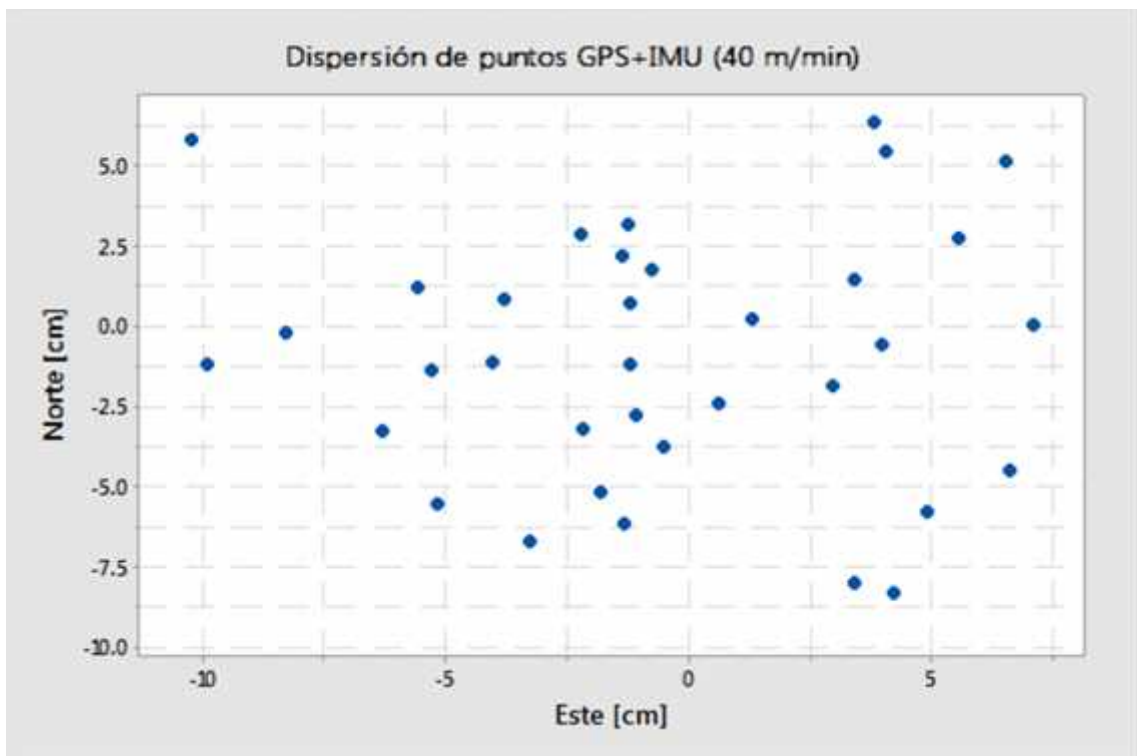


Gráfico 4-3. Dispersión de puntos GPS+IMU (40 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

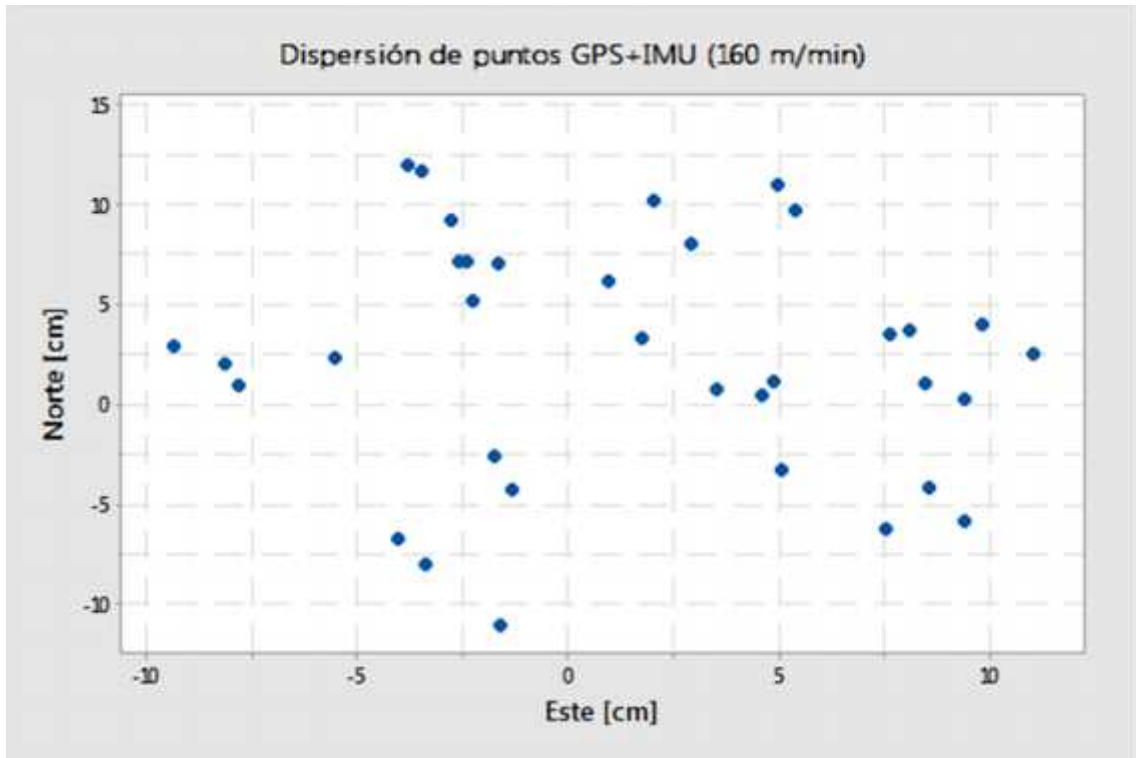


Gráfico 5-3. Dispersión de puntos GPS+IMU (160 m/min)
 Realizado por: CUENCA, Lorena & León, Jorge, 2017

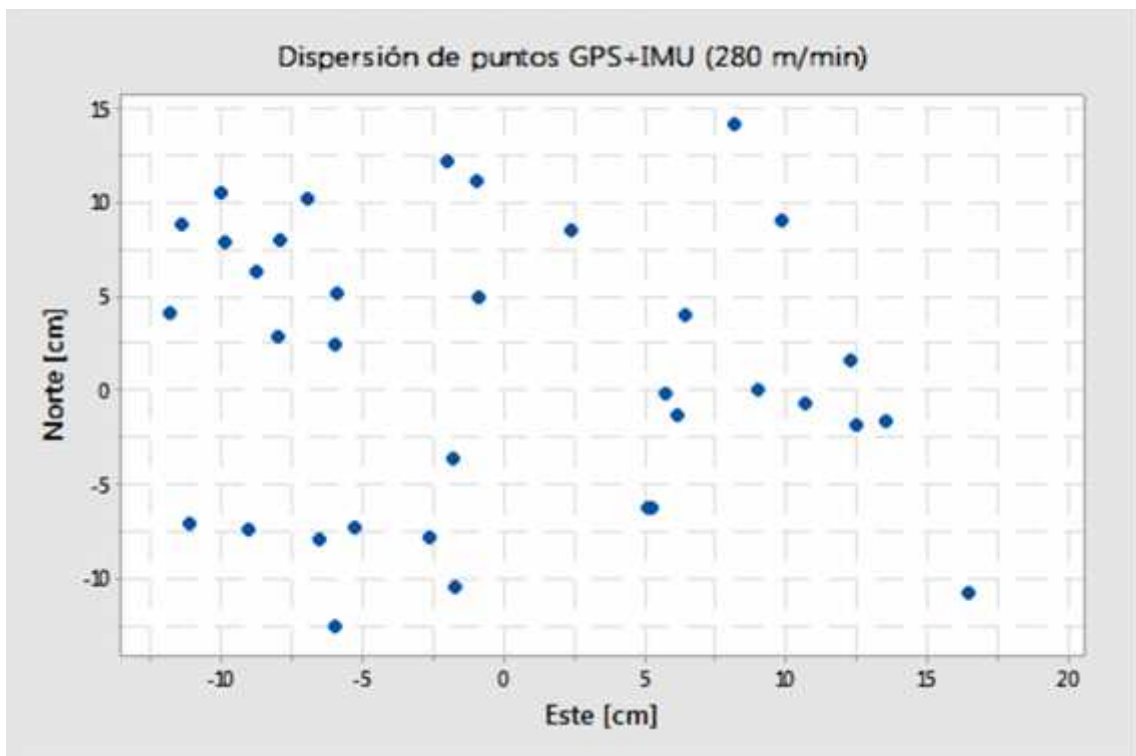


Gráfico 6-3. Dispersión de puntos GPS+IMU (280 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

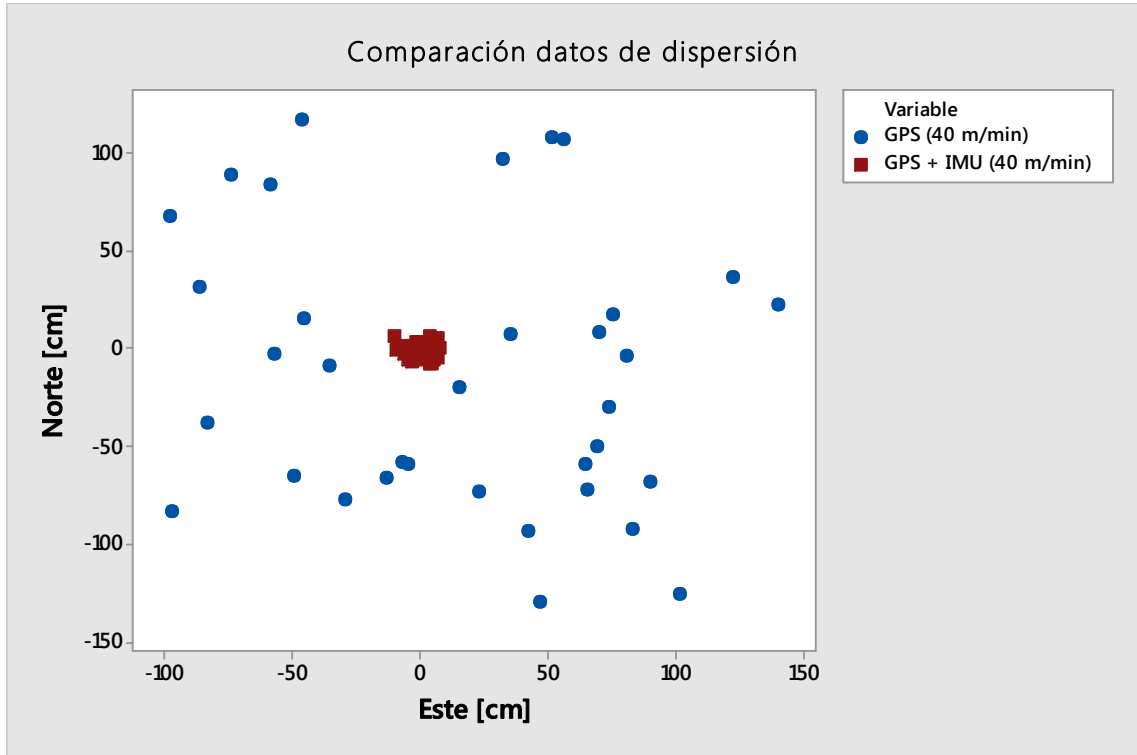


Gráfico 7-3. Comparación de dispersión de puntos (GPS, GPS+IMU 40 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

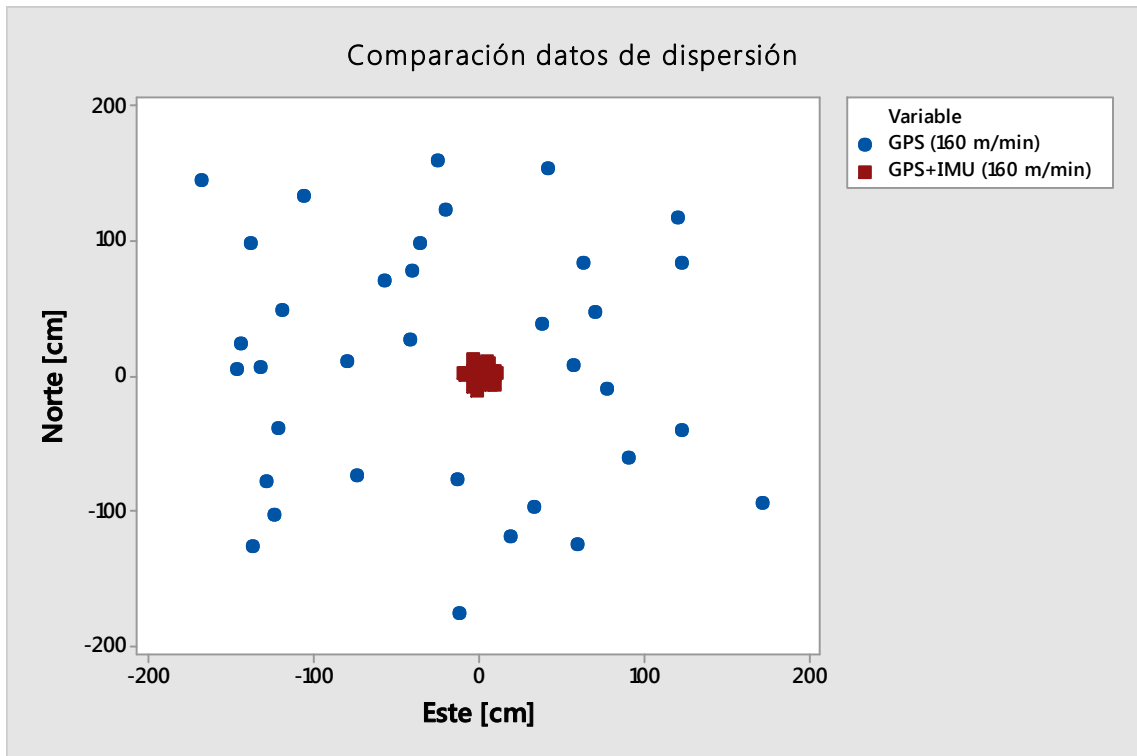


Gráfico 8-3. Comparación de dispersión de puntos (GPS, GPS+IMU 160 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

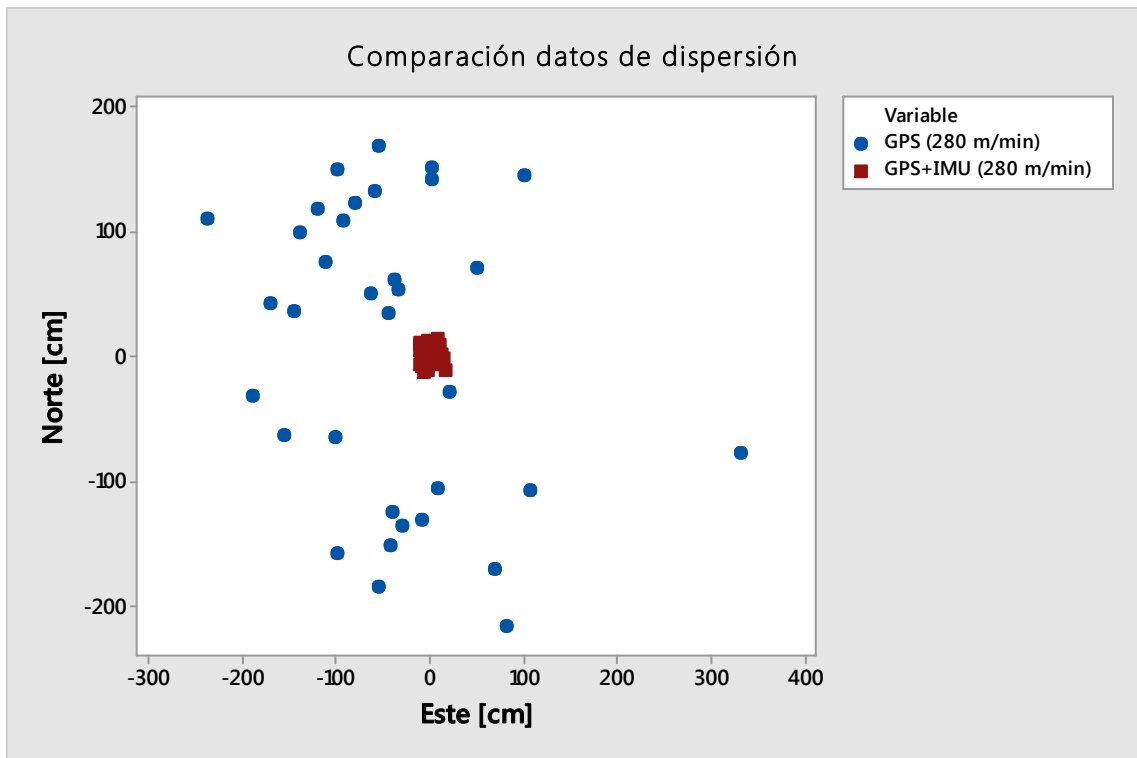


Gráfico 9-3. Comparación de dispersión de puntos (GPS, GPS+IMU 280 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

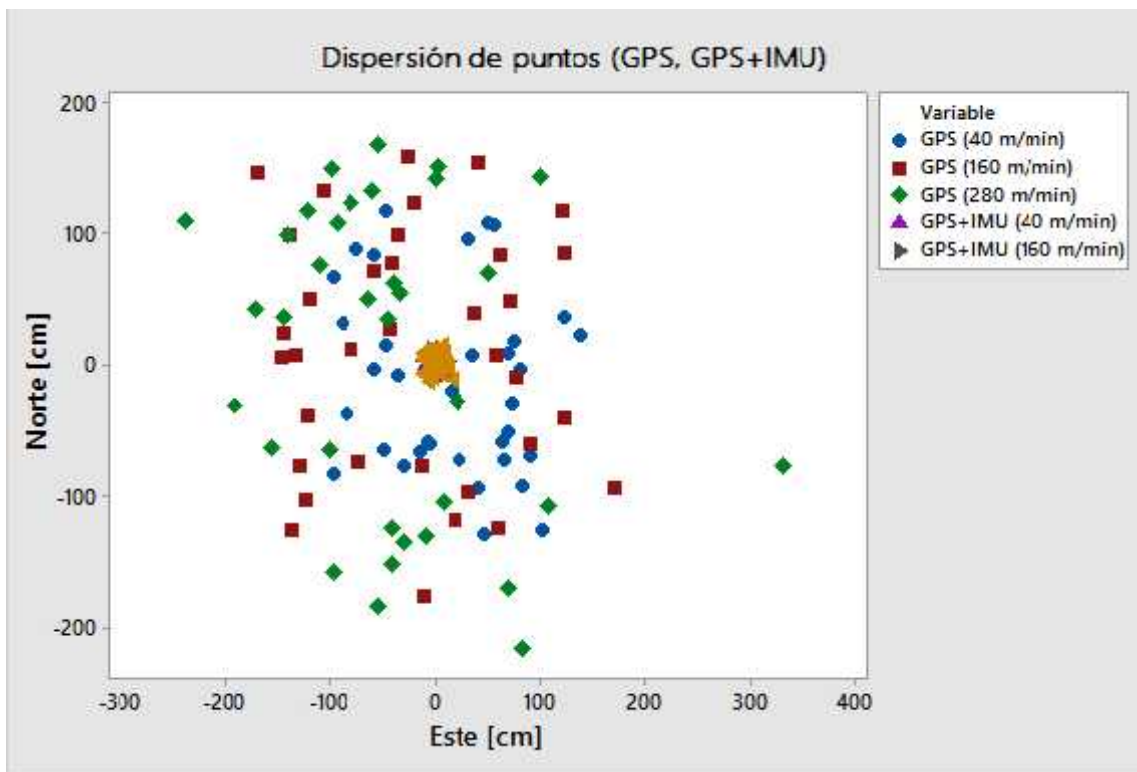


Gráfico 10-3. Dispersión de puntos (GPS, GPS+IMU)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

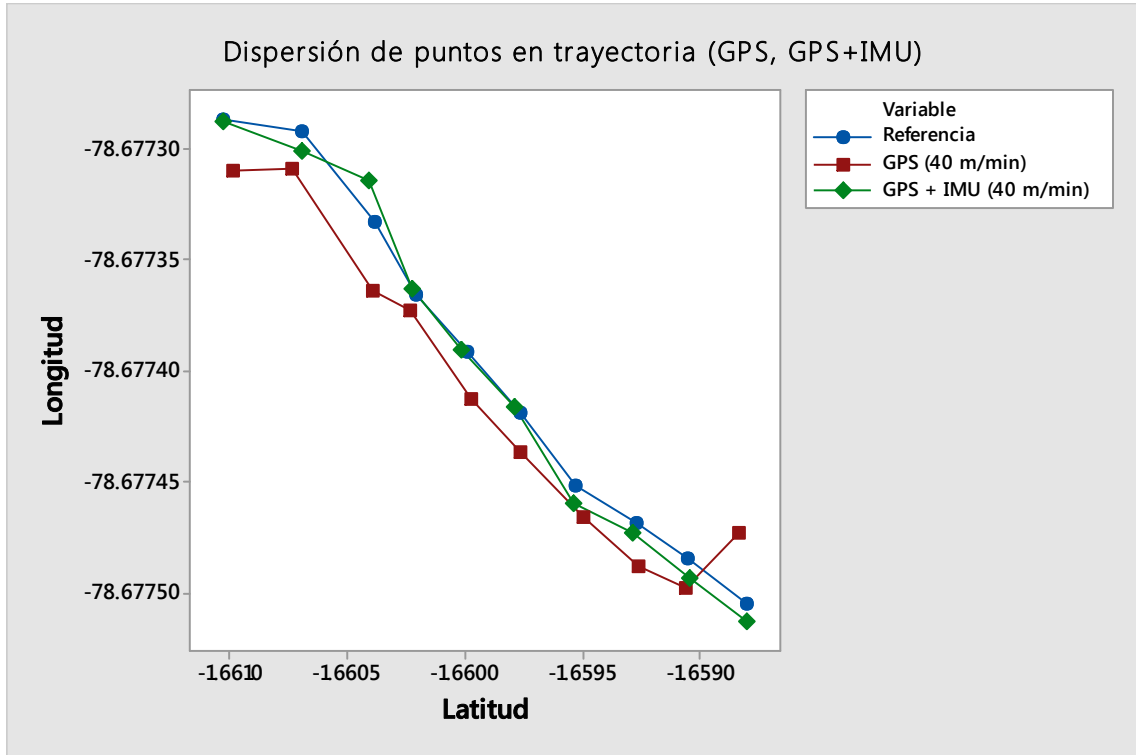


Gráfico 11-3. Dispersión de puntos en trayectoria (GPS, GPS + IMU 40 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

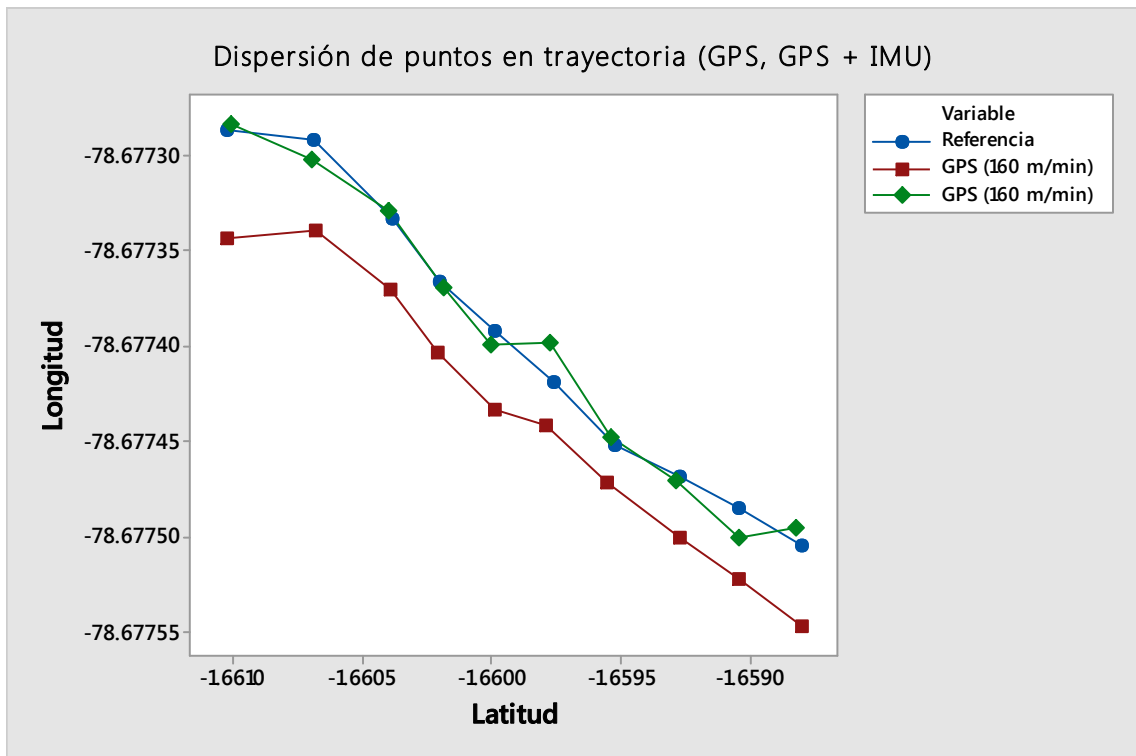


Gráfico 12-3. Dispersión de puntos en trayectoria (GPS, GPS + IMU 160 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

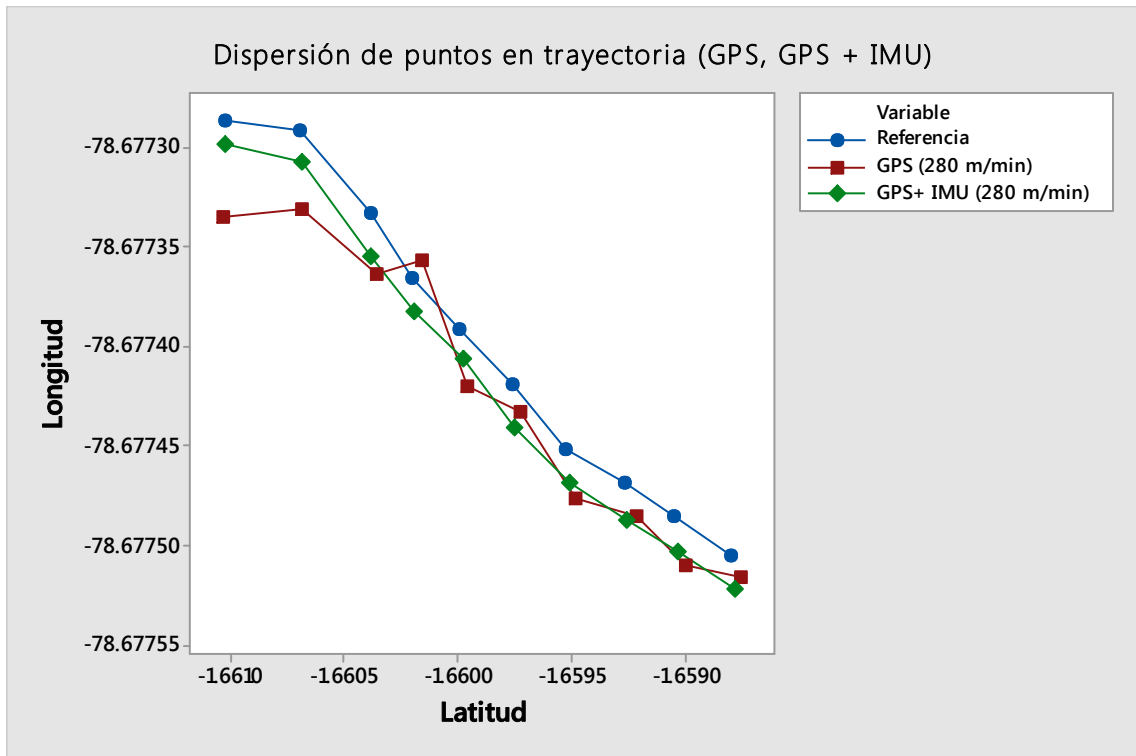


Gráfico 13-3. Dispersión de puntos en trayectoria (GPS, GPS + IMU 280 m/min)
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Con el propósito de visualizar las gráficas de puntos y la gráfica de trayectoria para identificar la diferencia existente entre el uso del GPS y la integración GPS / IMU se procede a incluirlas en la interfaz LJLAB

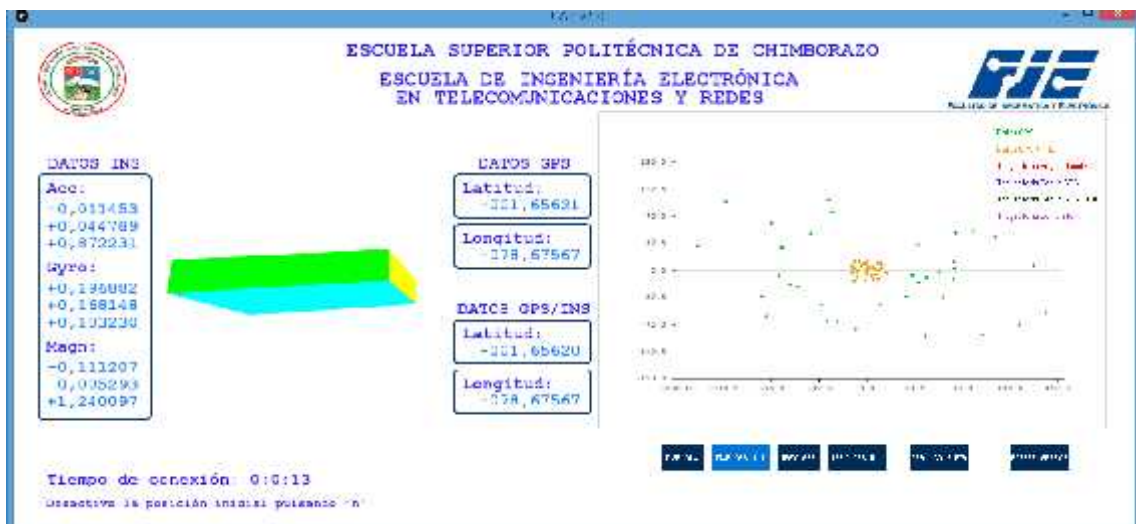


Figura 3-3. Dispersión de puntos LJLAB
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

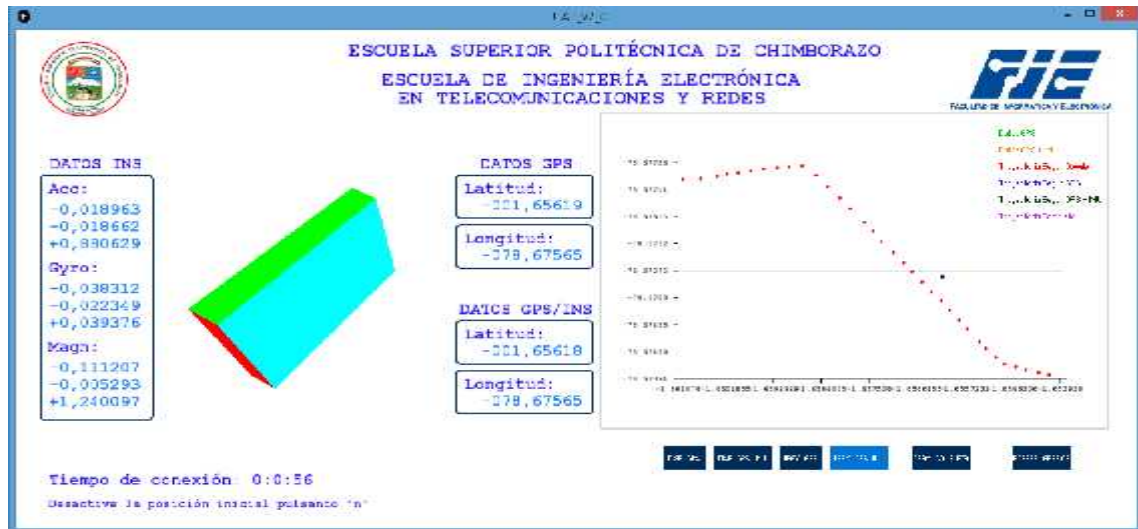



Figura 4-3. Dispersión de puntos en trayectoria LJLAB
 Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

3.8. Ventajas de la metodología propuesta

Como ya se analizó y se demostró por métodos estadísticos científicos, la primera y gran ventaja del dispositivo creado por el presente proyecto, es la disminución del error en un valor del 93,3% lo cual es un cambio bastante considerable tomando en cuenta que la inversión para alcanzar esta meta es relativamente baja.

Con ello se consigue realizar una propuesta económica que permita obtener un dispositivo GPS con alta precisión. A continuación, se muestra una tabla de comparación que permite evaluar las diferencias en el costo de adquisición entre un dispositivo de posicionamiento sofisticado y el dispositivo fruto del proyecto obtenido mediante fusión sensorial.

Tabla 8-3. Comparación de precios

Dispositivo propuesto de posicionamiento		Dispositivos presentes en el mercado		
		Costo:		Costo:
GPS NEO 6M			OPCIÓN 1	Spatial Navigation
	\$20			\$3.500
IMU MPU 9250			OPCIÓN 2	Vectornav
	\$15			\$2.900
PLACA ARDUINO MEGA			OPCIÓN 3	GEO Hnav
	\$30			\$3.000
ACCESORIOS				
Cables, estaño, pasta de soldar, cautín		\$30		
COSTO TOTAL DE LA PROPUESTA		\$95		

Realizado por: CUENCA, Lorena & LEÓN, Jorge, 2017

Los precios ilustrados en la Tabla 8-3. Hacen referencia a los precios ofertados en su país de fabricación, por lo que adquirir estos componentes en diferentes países, involucra añadir el precio concerniente al envío.

En la Tabla 8-3. Se hace evidente la diferencia abismal existente en los costos de adquisición entre el dispositivo propuesto por el presente proyecto, y los dispositivos existentes en el mercado que cumplen la función de integrar GPS e IMU.

Como puede apreciarse, el costo de reproducir el dispositivo propuesto es de noventa y cinco dólares americanos más el costo de envío, mientras que los otros dispositivos que cumplen la misma función, poseen un costo de adquisición entre dos mil novecientos y tres mil quinientos

dólares más el costo de envío del producto. Lo que indica que la propuesta representa un ahorro de aproximadamente el 94% frente a las opciones del mercado.

Adicionalmente es importante tener en cuenta que el dispositivo desarrollado por el proyecto incluye el software de monitoreo, que no solamente proporciona las coordenadas de posicionamiento, sino que a su vez pone a disposición del usuario una interfaz gráfica que permite conocer la dirección y sentido del dispositivo, mediante las coordenadas angulares yaw, pitch y roll.

3.9. Aplicaciones

Existe gran cantidad de aplicaciones posibles para el dispositivo GPS-IMU, ya sea para sistemas de posicionamiento como de navegación, en especial para el desarrollo de los modernos sistemas de pilotaje automático incrustados en vehículos tripulados y no tripulados como autos, aeronaves, submarinos, drones, etc.

Con ello se pone en claro que la investigación desarrollada aporta positivamente al desarrollo de las nuevas tecnologías. El software de monitoreo permite a su vez obtener información de las coordenadas angulares del dispositivo, función sumamente útil en el caso de aplicar el dispositivo en aeronaves en donde es importante conocer ángulos de inclinación y sentido para un correcto monitoreo de vuelo.

Como se mencionó anteriormente, las posibles aplicaciones son ilimitadas, no solamente en cuanto al posicionamiento, también es posible aplicar esta tecnología en el mundo de la realidad virtual, en donde se necesita de dispositivos especializados que registren movimiento, dirección y sentido, necesidad que es perfectamente cubierta por el dispositivo GPS/IMU.

CONCLUSIONES

- Se logró analizar el sistema de posicionamiento global determinando los principales errores que se presentan en la solución final de los equipos receptores GPS, además seleccionamos la arquitectura de acoplamiento estrecho (Tight Coupling) con el propósito de acoplar el sistema de navegación GPS con navegación inercial y el filtro de fusión.
- Se utilizó los componentes adecuados para la construcción del prototipo que consta del módulo GPS NEO 6M, el sensor IMU MPU 9250 y la placa Arduino MEGA, estos dispositivos son elegidos por sus características técnicas útiles para la fusión sensorial al igual que sus características económicas que representan un ahorro aproximado del 94% frente a otras opciones del mercado.
- Se elaboró el prototipo GPS mediante la conexión de los dispositivos antes mencionados, la fusión sensorial permite adquirir señales más detalladas de la magnitud en cuanto a velocidad, giros, campo magnético, latitud y longitud, no obstante la desventaja de los INS es la acumulación de errores en periodos largos de tiempo que son eliminados mediante filtros de estimación que permitan eliminar el bias creado.
- El acoplamiento estrecho (Tight Coupling) permite mediante fusión de las medidas INS (MPU 9250) y del GPS (NEO 6M V2), logrando reducir el error de GPS de 122.3 cm a 8.17 cm que se obtuvo mediante la fusión de GPS/INS, que representa un 93% aproximadamente del error de posicionamiento comparado con el error del dispositivo GPS en posicionamiento estándar.
- Se realizó pruebas individuales determinando la longitud y latitud del dispositivo GPS y se comparó con los mismos parámetros determinados con la integración del GPS/INS siendo efectiva esta aplicación al verificar que existe mejor precisión de posicionamiento.
- Se realizaron las pruebas de campo pertinentes dentro de la Escuela Superior Politécnica de Chimborazo cuya trayectoria seleccionada fue el ingreso de la puerta principal hacia la puerta secundaria tramo de 1 kilómetro de distancia en la cual se tomó 35 puntos para realizar las comparaciones entre dispositivos.
- Se realizó las pruebas a diferentes velocidades (40, 160 y 280 m/min) para poder obtener mediciones y así observar el comportamiento de los sistemas, dando como resultado que a mayor velocidad se obtiene mayor cantidad de datos, debido a que el GPS trabaja a 10 Hz y el INS a 50 Hz, es decir se pudo obtener mayor cantidad de datos para la respectiva corrección de errores a mayor velocidad.
- El software de monitoreo denominado LJLAB permite visualizar en tiempo real la dirección y sentido del dispositivo mediante sus coordenadas angulares e interfaz gráfica además de obtener los parámetros de longitud y latitud de la integración GPS/INS.

RECOMENDACIONES

- Se recomienda en el caso de necesitar medidas extremadamente precisas de posicionamiento, reducir la velocidad del dispositivo de ser posible en el momento de la medición, las pruebas demostraron que este factor no produce cambios altamente significativos, no obstante se puede apreciar en los resultados de las pruebas que existe una pequeña influencia de la velocidad en la variable respuesta.
- El dispositivo necesita de un pequeño tiempo de estabilidad, en donde los filtros se encargan de fijar las señales provenientes de la unidad de medición inercial, por tal motivo se recomienda esperar un tiempo de tres minutos una vez se haya energizado el dispositivo con el propósito de evitar errores en la lectura de los datos.
- Se recomienda utilizar nuevas arquitecturas de acoplamiento de fusión sensorial con los dispositivos modernos que cada vez poseen mayor tecnología, lo cual reducirá más el error existente en las mediciones.
- La mayoría de aplicaciones para sistemas de navegación inercial, requieren de un dispositivo inalámbrico, por tal motivo se recomienda hacer uso de la programación propuesta por el presente proyecto para la conversión del sistema alámbrico en inalámbrico por medio de XBee (radio frecuencia).
- Esta investigación permite abrir campo a otros trabajos de titulación donde pueden aplicar esta tecnología en sistemas de navegación autónomos como lo son los drones o vehículos no tripulados, se recomienda desarrollar y mejorar aplicaciones para este tipo de equipos.
- Se recomienda que en caso de necesitar ayuda o el código de programación de los dispositivos contactarse con los autores a los siguientes correos: Lorena Cuenca (cuencasaenz_lorena@hotmail.com) y Jorge León (jal_c17@hotmail.com).

BIBLIOGRAFÍA

Abc del acelerómetro [En línea] 17 de Febrero de 2014. [Consulta: 06 de Mayo de 2017.]. Disponible en: <https://www.5hertz.com/image/cache/catalog/tutoriales/acelerometro/ac1-500x500.png>

Arduino uno. [En línea] 15 de Enero de 2017. [Consulta: 06 de Mayo de 2017.]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardUno>

¿Qué es arduino?. [En línea] 2017. [Consulta: 06 de Mayo de 2017.]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>

Brandão, Samuel. *Sobre gps.* [En línea] 2016. [Consulta: 03 de Mayo de 2017.]. Disponible en: <http://vaztolentino.com.br/secao/46-Sobre-GPS>

Practicaciencia. [En línea] 2017. [Consulta: 05 de Mayo de 2017.]. Disponible en: <https://practicaciencia.com/giroscopios-peonzas-y-trompos/925-giroscopio-original-chandler.html>

Corona Ramírez, Leonel Germán y otros. *Sensores y actuadores. Aplicaciones con Arduino.* México D.F- México. : Grupo editorial PATRIA, 2014,p.18

Crespo, Enrique. *Aprendiendo Arduino. Wordpress.* [En línea] 30 de Junio de 2016. [Consulta: 07 de Mayo de 2017.]. Disponible en: <https://aprendiendoarduino.wordpress.com/tag/xbee/>

Modulo GPS GY-GPS6MV2 / NEO-6M-V2 R3. [En línea] 2014. [Consulta: 07 de Mayo de 2017.]. Disponible en: <https://www.ardobot.com/gps-neo-6m-v2.html>

España, Martín. *Sistema de navegación integrada con aplicaciones.* [en línea] Madrid - España: AADECA, 2012, p.22.[Consulta: 04 de Mayo de 2017]. Diponible en: <https://goo.gl/CWyCXZ>

FreeIMU. [En línea] 11 de Noviembre de 2014. [Consulta: 05 de Junio de 2017.]. Disponible en :https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.com&sp=n

mt4&tl=es&u=https://github.com/mjs513/FreeIMU-Updates/wiki/04.-FreeIMU-
Calibration&usg=ALkJrhBkqEFReGZbjnGIjLn2k0wa9jIA

Constellation, [En línea] 13 de Febrero de 2017. [Consulta: 02 de Mayo de 2017.]. Disponible en:
<http://www.gps.gov/systems/gps/space/>

Gutiérrez Pulido, Humberto. *Análisis y diseño de experimentos*. Segunda Edición, México D.F
- México. : McGraw Hill, 2008, p. 104

JIMÉMEZ RODRÍGUEZ, Tamara y ROS BERNABEU, María Elena. Sistema de
Posicionamiento Global (GPS). *Gravitación y Astrofísica*. [En línea] 2010. [Consulta: 10 de
Mayo de 2017.] Disponible en: http://webs.um.es/bussos/GPSresumen_TamaraElena.pdf,
pp.13-14

Juarez, Cristian. controlaplicado.blogspot.com. [Blog] 8 de Octubre de 2015. [Consulta: 04 de
Mayo de 2017.]. Disponible en: [http://controlaplicado.blogspot.com/2015/10/unidad-de-
medicion-inercial-imu-mpu9150.html](http://controlaplicado.blogspot.com/2015/10/unidad-de-medicion-inercial-imu-mpu9150.html)

Grewal.M, y otros. Fusión Sensórica INS/ GPS para Navegación en Plataformas. [En línea]
[Tesis Postgrado]. Universidad Militar Nueva Granada, Facultad Ingeniería, Maestría en
Ingeniería Mecatrónica. Bogotá-Colombia. (2007).p. 2. [Consulta: 02 de Mayo de 2017.]
Disponible en: [http://repository.unimilitar.edu.co/bitstream/10654/11374/1/CastroPescadorAndresMauricio201
3.pdf](http://repository.unimilitar.edu.co/bitstream/10654/11374/1/CastroPescadorAndresMauricio2013.pdf). 1

Aggarwal.P, Syed.Z, y otros. *MEMS-Based Integrated Navigation, ser. GNSS technology and
applications series*. s.l., Calgary - Canadá : Artech House, 2010, pp.63-80

WILDERMANN, Eugen y HOYER, Melvin. *Aspectos del procesamiento de mediciones GPS
para labores de prospección*. Zulia : Escuela de Ingeniería Geodésica. Departamento de Geodesia
Superior. Universidad de Zulia, 1992. Vol. 15. ISSN: 0254-0770, p.45

Metodología de prototipos: pasos sobre cómo usarla correctamente. [En línea] 2017. [Consulta: 13 de Noviembre de 2017.], Disponible en: <https://www.lumitex.com/blog/prototyping-methodology>.

ANEXOS

ANEXO A. PROGRAMACIÓN MÓDULO NEO 6M

```
GPS_HCO6M                                     GPS_V106V
#include <SoftwareSerial.h>
#include <TinyGPS.h>
TinyGPS gps;
SoftwareSerial serialgps(4,3);
int year;
byte month, day, hour, minute, second, hundredths;
unsigned char received, failed_checksum;
void setup()
{
  Serial.begin(19200);
  serialgps.begin(9600);
  Serial.println("");
  Serial.println("GPS NEO-6M V2 Instalado");
  Serial.println("---Buscando señal---");
  Serial.println("");
}
void loop()
{
  while(serialgps.available())
  {
    int c = serialgps.read();
    if(gps.encode(c))
    {
      //Time, Location, Longitudes
      unsigned long time = gps.time();
      Serial.print("Fecha: "); Serial.print(time);
      Serial.print(" Hora: "); Serial.print(hour);
      Serial.print(" Minuto: "); Serial.print(minute);
      Serial.print(" Segundo: "); Serial.print(second);
      Serial.print(" "); Serial.println(hundredths);
      Serial.print("Latitud (Norte): "); Serial.print(gps.location().lat());
      Serial.print("Longitud (Oeste): "); Serial.print(gps.location().lon());
      Serial.print("Velocidad (km/h): "); Serial.println(gps.speed_kmh());
      Serial.print("Altitud (m): "); Serial.println(gps.altitude_m());
      Serial.println("");
      gps.reset();
    }
  }
  Serial.println("Recepción de datos completada");
}
```

ANEXO B. PROGRAMACIÓN ACELERÓMETRO MPU 9250

```
MPU9250-ACELEROMETRO
#include "Wire.h"
#include "SPI.h"
#include "I2Cdev.h"
#include "MPU60X0.h"
MPU60X0 accelgyro;
int16_t ax, ay, az;
void setup() {
  Wire.begin();
  Serial.begin(57600);
  accelgyro.initialize();
}
void loop() {
  accelgyro.getAcceleration(&ax, &ay, &az);
  Serial.println(ax);
  Serial.print("aceleracion eje x: ");
  Serial.print(ax); Serial.print("\t");
  Serial.print("aceleracion eje y: ");
  Serial.print(ay); Serial.print("\t");
  Serial.print("aceleracion eje z: ");
  Serial.print(az); Serial.print("\t");
}
```

ANEXO C. PROGRAMACIÓN GIROSCOPIO MPU 9250

```

MPU9250-GIROSCOPIO

#include "Wire.h"
#include "SPI.h"
#include "I2Cdev.h"
#include "MPU60X0.h"
MPU60X0 accelgyro;
int16_t gx, gy, gz;
void setup() {
  Wire.begin();
  Serial.begin(57600);
  accelgyro.initialize();
  void loop() {
    accelgyro.getRotation(&gx, &gy, &gz);
    Serial.println(gx);
    Serial.print ("giroscopio eje x: ");
    Serial.print(gy); Serial.print("\t");   Serial.print ("giroscopio eje y: ");
    Serial.print(gz); Serial.print("\t");   Serial.print ("giroscopio eje z: ");
  }
}

```

ANEXO D. PROGRAMACIÓN MAGNETÓMETRO MPU 9250

MPU9250-MAGNETÓMETRO	MPU9250-MAGNETÓMETRO	MPU9250-MAGNETÓMETRO
<pre> #include "Wire.h" #include <AK8963.h> #include <MPU60X0.h> #include <I2Cdev.h> #include <SPI.h> AK8963 mag(magI2C, I2C); MPU60X0 accelgyro = MPU60X0(false, 0x68); int16_t mx, my, mz; enum Mode_t { MFS_14BITS = 0, MFS_16BITS = 1 }; #define AK8963_MODE_SINGLE 0x0 #define AK8963_MODE_CONT1 0x02 #define AK8963_MODE_CONT2 0x06 #define AK8963_MODE_ZRX 0x04 uint8_t ModeLe = MFS_16BITS; uint8_t Mode = AK8963_MODE_CONT1; float mRes; uint8_t axsk, asay, asaz; float ASRX, ASRY, ASRZ; float heading, MX, MY, MZ, magbias[3]; void setup() { Wire.begin(); Serial.begin(57600); getI2C(); magbias[0] = +470.; magbias[1] = +120.; magbias[2] = +125.; mag.initialize(); Serial.println(mag.getMode(), BIN); </pre>	<pre> Serial.println(); mag.getSelfTest(axsk, asay, asaz); Serial.println("mag:\n"); Serial.print(ax): Serial.print("\t"); Serial.print(ay): Serial.print("\t"); Serial.print(mz): Serial.println("\t"); mag.getAdjustment(ASRX, ASRY, ASRZ); ASRX = ((float) axsk - 128)/256. + 1.; ASRY = ((float) asay - 128)/256. + 1.; ASRZ = ((float) asaz - 128)/256. + 1.; mag.setModeRes(Mmode, Mscale); Serial.println(); Serial.print(ASRX); Serial.print(" "); Serial.print(ASRY); Serial.print(" "); Serial.print(ASRZ); Serial.println(); delay(1000); void loop() { mag.getHeading(awx, awy, awz); MX = (float)awx*MRes*ASRX - magbias[0]; MY = (float)awy*MRes*ASRY - magbias[1]; MZ = (float)awz*MRes*ASRZ - magbias[2]; Serial.print(MY); Serial.print("\n"); Serial.print("magnetometro eje y: "); Serial.print(MZ); Serial.print("\n"); Serial.print("magnetometro eje z: "); Serial.print(MX); Serial.print("\n"); Serial.print("magnetometro eje x: "); void getMRes() { switch (ModeLe) case MFS_14BITS: mRes = 10.*4219./8190.; break; case MFS_16BITS: </pre>	<pre> mRes = 10.*4219./8190.; break; } </pre>