



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA
ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA**

**TRABAJO DE TITULACIÓN PREVIO A LA
OBTENCIÓN DEL TÍTULO DE
INGENIERO MECATRÓNICO**

TEMA:

**“DESARROLLO DE UN SISTEMA DE
ESTABILIZACIÓN PARA CÁMARAS A BORDO
DE MINI RUAVS QUE MINIMICE EL EFECTO
DE LOS MOVIMIENTOS FANTASMA EN
VIDEOS CAPTURADOS – MKCHASER”**

AUTORES:

**ABAROA DUQUE, THOMAS EDUARDO
FUERTES TROYA, RONNIE ALEXANDER**

DIRECTOR:

ING. AGUILAR, WILBERT

SANGOLQUÍ - ECUADOR

2017



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA**

Certificación de la Elaboración del Trabajo de Titulación

El suscrito, Dr. Wilbert G. Aguilar, director del trabajo de titulación, respectivamente, certifica que el trabajo de titulación titulado “DESARROLLO DE UN SISTEMA DE ESTABILIZACIÓN PARA CÁMARAS A BORDO DE MINI RUAVS QUE MINIMICE EL EFECTO DE LOS MOVIMIENTOS FANTASMA EN VIDEOS CAPTURADOS -MKCHASER”, fue realizado en su totalidad por Abaroa Duque Thomas Eduardo y Fuertes Troya Ronnie Alexander, como requerimiento parcial para la obtención del título de Ingeniero en Mecatrónica.

Wilbert G. Aguilar, Ph.D.

DIRECTOR



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA**

AUTORÍA DE RESPONSABILIDAD

El trabajo de titulación “DESARROLLO DE UN SISTEMA DE ESTABILIZACIÓN PARA CÁMARAS A BORDO DE MINI RUAVS QUE MINIMICE EL EFECTO DE LOS MOVIMIENTOS FANTASMA EN VIDEOS CAPTURADOS -MKCHASER” ha sido desarrollado citando las fuentes correspondientes y respetando las disposiciones legales que protegen los derechos de autor vigente.

En virtud de esta declaración, las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo son de exclusiva responsabilidad de los autores.

ELABORADO POR:

Abaroa Duque Thomas Eduardo

Fuertes Troya Ronnie Alexander



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRÓNICA**

AUTORIZACIÓN

Nosotros: Abaroa Duque Thomas Eduardo y Fuertes Troya Ronnie Alexander

Autorizamos a la universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la institución el trabajo de titulación titulado: “DESARROLLO DE UN SISTEMA DE ESTABILIZACIÓN PARA CÁMARAS A BORDO DE MINI RUAVS QUE MINIMICE EL EFECTO DE LOS MOVIMIENTOS FANTASMA EN VIDEOS CAPTURADOS - MKCHASER” cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Abaroa Duque Thomas Eduardo

Fuertes Troya Ronnie Alexander

DEDICATORIA

El presente trabajo lo dedico a Dios, quien siempre me ha bendecido y ayudado en todo el trayecto de mi vida, sin él no lograría nada.

Quiero dedicar este trabajo a las personas que me han querido toda mi vida, siempre me han brindado amor, han estado viendo por mi bien y ayudándome en todo momento, mi familia. En especial a mi abuelita Aida Tobar, ya que sin ella no hubiera podido llegar a donde estoy, y sin ella no sería nadie. A mi hermanita Stephanie Abaroa, quien ha estado conmigo, y a mi madre Margarita Duque, quien ha sido mi más grande ejemplo a seguir en todos los aspectos de mi vida, a quien siempre extrañaremos, que en paz descansa.

Thomas Abaroa Duque

DEDICATORIA

Dedico este trabajo a mi padre y a mi madre, ya que parte de todos sus esfuerzos, sacrificios, trabajo duro, dedicación, paciencia, y amor incondicional se encuentran reflejados en este trabajo.

Ronnie Fuertes Troya

AGRADECIMIENTOS

Quiero agradecer a Dios por haberme apoyado a completar esta meta en mi vida, a mis familiares por su apoyo incondicional, y a mi abuelita Aidita porque siempre ha estado allí conmigo cuando más la he necesitado.

A mis amigos, con quienes hemos compartido muchos momentos tristes y felices, y quienes me han acompañado en este arduo camino.

Y a todos nuestros profesores, por encaminarnos en este trayecto profesional.

Thomas Abaroa Duque

AGRADECIMIENTOS

Agradezco a Dios por todas las bendiciones y enseñanzas que derrama sobre mí.

A mi Familia por todo el amor y apoyo incondicional que me brinda día a día.

A mis Amigos por todo su apoyo y consejos que me brindaron a lo largo de esta etapa de la universidad y desarrollo del proyecto.

A los docentes por todas sus enseñanzas.

Ronnie Fuertes Troya

TABLA DE CONTENIDO

Certificación de la Elaboración del Proyecto.....	ii
Autoría de Responsabilidad	¡Error! Marcador no definido.
Autorización	¡Error! Marcador no definido.
Dedicatoria.....	iv
Dedicatoria.....	vi
Agradecimientos	vii
Agradecimientos	viii
Tabla de contenido	ix
Índice de Figuras	xiv
Índice de cuadros.....	xix
índice de tablas.....	xx
RESUMEN	xxi
ABSTRACT.....	xxii
Capítulo 1.....	1
1.1. Planteamiento del Problema	1
1.1.1. Especificaciones Técnicas.....	2
1.2. Antecedentes	3
1.3. Justificación.....	6
1.4. Objetivos	10
1.4.1. General.....	10
1.4.2. Específicos	10
1.5. Alcance del Proyecto.....	11
1.5.1. Diseño Mecánico	11
1.5.2. Diseño Electrónico y Eléctrico	11

1.5.3.	Diseño del sistema de control.....	12
1.6.	Metodología	12
1.7.	Estructura del documento	13
Capítulo 2.....		15
2.1.	Sistemas Aéreos no tripulados a base de Rotores (RUAS)	15
2.1.1.	Clasificación de las RUAS	16
2.1.2.	Autopilot.....	18
2.1.3.	Estaciones de control	20
2.2.	Plataforma estabilizada inercialmente (ISP).....	26
2.2.1.	Estructura	28
2.2.2.	Amortiguadores	30
2.2.3.	Sensores.....	31
2.2.4.	Motores.....	32
2.2.5.	Controlador de gimbal	36
2.2.6.	Dispositivo de captura de imágenes.....	36
2.3.	Ordenadores on-board	37
2.3.1.	Single-Board Computer (SBC).....	38
2.3.2.	Procesador multicore asimétrico.....	38
2.4.	Protocolos de comunicación	39
2.4.1.	Minicom	39
2.4.2.	UART	40
2.4.3.	WiringPi	40
2.4.4.	ROS.....	40
2.4.5.	SSH	40
2.5.	Estabilización de vídeo.....	41

2.5.1.	Filtro de Kalman.....	43
2.5.2.	OpenCV.....	44
2.5.3.	RANSAC.....	44
2.5.4.	SURF.....	44
2.5.5.	FAST.....	44
2.5.6.	BRIEF.....	45
2.5.7.	ORB.....	45
2.5.8.	RobustMatcher.....	45
2.6.	Resumen.....	46
Capítulo 3.....		48
3.1.	Sistema Aéreo no tripulado.....	48
3.1.1.	Selección del Autopilot.....	48
3.1.2.	Selección de la estación de control.....	51
3.1.3.	Configuración del Autopilot.....	52
3.1.4.	Diagrama de conexión de los periféricos del Pixhawk.....	60
3.1.5.	Conexión de los motores al Autopilot.....	61
3.1.6.	Conexión del receptor de radio control al Autopilot.....	61
3.2.	Componentes del sistema de estabilización.....	62
3.2.1.	Plataforma estabilizada inercialmente (ISP).....	62
3.2.2.	Ordenador on-board.....	80
3.2.3.	Diseño de la base total del sistema.....	89
3.3.	Conexiones del sistema de estabilización.....	95
3.3.1.	Diagrama de conexión del SimpleBGC.....	95
3.3.2.	Conexión de la cámara a la SBC.....	96
3.3.3.	Conexiones para la envío y recepción señal de control.....	97

3.3.4.	Conexión del Odroid XU4 con el Pixhawk	99
3.3.5.	Comunicación estación de tierra - Odroid XU4.....	99
3.3.6.	Conexión del R9D al SimpleBGC	100
3.4.	Software del sistema de estabilización.....	102
3.4.1.	Controlador de la gimbal.....	102
3.4.2.	Sistema Operativo del Odroid XU4.....	114
3.5.	Algoritmos del sistema de estabilización.....	120
3.5.1.	Filtro de Kalman.....	120
3.5.2.	Planta del dron.....	120
3.5.3.	Algoritmo de estabilización de video.....	134
3.5.4.	Implementación del algoritmo	137
3.6.	Resumen.....	140
Capítulo 4.....		142
4.1.	Pruebas mecánicas.....	142
4.1.1.	Peso máximo soportado por el dron	142
4.1.2.	Prueba dimensional de las placas base	143
4.1.3.	Prueba de compresión de los amortiguadores	146
4.2.	Comparación de software	147
4.2.1.	Mission Planner	147
4.2.2.	QGroundControl	148
4.3.	Tiempo de funcionamiento del sistema	149
4.4.	Pruebas del Algoritmo	151
4.4.1.	Obtención del RMSE	159
4.4.2.	Obtención del PSNR.....	165
4.4.3.	Obtención del ITF	168

4.5. Resumen.....	170
Capítulo 5.....	171
5.1. Análisis financiero	171
5.1.1. Costos directos	171
5.1.2. Costos totales directos	176
5.2. Costos indirectos.....	176
5.2.1. Costos totales indirectos.....	176
5.3. Costos totales	177
Capítulo 6.....	178
6.1. Conclusiones.....	178
6.2. Recomendaciones.....	179
Bibliografía	180
ANEXOS	187

ÍNDICE DE FIGURAS

Figura 1 Pruebas de Estabilización del algoritmo de estabilización.....	8
Figura 2 “Rolling Shutter” o bamboleo	9
Figura 3: Categorías de los RUAS.....	16
Figura 4 Sistema de control de vuelo del UAV	19
Figura 5: Ejemplo de Autopilots	20
Figura 6: Vehículos de Mission Planner.....	21
Figura 7: Mission Planner	22
Figura 8: Logotipo de APM Planner	23
Figura 9: QGroundControl	24
Figura 10: Versiones de UgCS	25
Figura 11: Planificación de misiones.....	26
Figura 12: Ejemplos de configuraciones de ISP	27
Figura 13: Causas de rotación inercial de la línea de visión	28
Figura 14: Configuraciones de plataformas de 3 grados de libertad.....	29
Figura 15: Movimientos de la gimbal y gimbal lock.....	29
Figura 16: Ejemplo de amortiguadores para gimbal	30
Figura 17: Errores comunes de las IMU	32
Figura 18: Estructura de un motor eléctrico sin escobillas.....	33
Figura 19: Secuencia de conmutación del motor sin escobillas.....	34
Figura 20: Ejemplo de controladores de gimbal.....	36
Figura 21: Ejemplo de Single-Board Computers.....	38
Figura 22: Modo de configuración del comando minicom.....	40
Figura 23: Pixhawk PX4.....	51
Figura 24: Conexión entre el Pixhawk y computadora.....	53
Figura 25: Instalación del Firmware de Pixhawk PX4	54
Figura 26: Tipo de vehículo	54
Figura 27: Calibración del radio control	55
Figura 28: Calibración de sensores	56
Figura 29: Indicador de calibración	56
Figura 30: Modos de vuelo	57

Figura 31: Calibración Final del Pixhawk	58
Figura 32: Extracción del Micro-SD del Pixhawk	59
Figura 33: Acople de la tarjeta micro-SD al computador.....	59
Figura 34 Conexión de los periféricos al Pixhawk.....	60
Figura 35 Conexión de los motores al Autopilot.....	61
Figura 36 Conexión del receptor de radio control al Autopilot	61
Figura 37: Ensamble del dron	62
Figura 38: Estructura de la gimbal	63
Figura 39: Placa base del motor del eje Yaw.....	66
Figura 40: Desplazamiento en la placa del motor generado por la carga	68
Figura 41: Tensión en la placa del motor generado por la carga	68
Figura 42: Amortiguadores	69
Figura 43: Carga en el amortiguador	70
Figura 44: Desplazamiento del amortiguador generado por la carga	70
Figura 45: Tensión en el amortiguador generado por la carga	71
Figura 46: Controlador de gimbal.....	73
Figura 47: IMU MPU6050	73
Figura 48: Pines para control de motores	74
Figura 49: Estructura y posición de los motores e IMU.....	74
Figura 50: Posición de la IMU en la estructura	75
Figura 51: Fuerzas aplicadas en el sistema gimbal	78
Figura 52: Diagrama de fuerzas de la gimbal	79
Figura 53: Posición del peso adicional.....	80
Figura 54: Desempeño de CPU/RAM de varias SBC	82
Figura 55: Memorias eMMC y microSD de Odroid XU4	83
Figura 56: Diferencia de rendimiento entre SD y eMMC.....	83
Figura 57: Adaptador para memoria eMMC.....	84
Figura 58 Dimensiones del Odroid XU4.....	84
Figura 59: Adaptador Shifter Shield	85
Figura 60: Dimensiones del regulador de voltaje DROK.....	85
Figura 61: Energización del Odroid XU4.....	86

Figura 62: Diseño de la carcasa para el Odroid XU4 – Vista isométrica 1...	87
Figura 63: Diseño de la carcasa para el Odroid XU4 – Vista isométrica 2...	87
Figura 64: Puntos de sujeción del Odroid XU4	88
Figura 65: Carcasa del Odroid XU4	89
Figura 66: Ubicación de los componentes en la base - 1	90
Figura 67: Ubicación de los componentes en la base - 2	91
Figura 68: Fuerzas aplicadas en la base	91
Figura 69: Ubicación de las fuerzas aplicadas en la base	92
Figura 70: Desplazamiento en la placa del motor generado por la carga	93
Figura 71: Tensión en la placa del motor generado por la carga	93
Figura 72: Diagrama de fuerzas de la base	94
Figura 73: Diagrama de conexión de los motores al controlador	95
Figura 74: Conexión del cable USB	96
Figura 75: Ensamble de la gimbal.....	97
Figura 76: Conexión entre el R9D y el Odroid XU4.	98
Figura 77: Conexión entre el Odroid XU4 y el SimpleBGC.....	98
Figura 78: Puertos de comunicación entre Pixhawk y el Odroid XU4	99
Figura 79: Comunicación Estación de Tierra-Odroid XU4	100
Figura 80: Señal de control del RC hacia el eje Yaw de la gimbal.....	101
Figura 81: Ancho de pulso del valor mínimo de la señal de control	102
Figura 82: Ancho de pulso del valor máximo de la señal de control	102
Figura 83: Interfaz del programa SipleBGC GUI v2	104
Figura 84: Instalacion del Firmware del controlador de gimbal	105
Figura 85: Posiciones del sensor para calibración.....	106
Figura 86: Posición de los ejes de la IMU	106
Figura 87: Indicadores de la posición de la IMU	107
Figura 88: Utilización del cubo nivelador de 3 ejes	107
Figura 89: Posición de la cámara para calibración	108
Figura 90: Ángulos de compensación para la cámara	108
Figura 91: Error I2C	109
Figura 92: Núcleo de ferrita para en la IMU	109

Figura 93: Habilitación del 3er eje del controlador	110
Figura 94: Calibración de parámetros PID del controlador	111
Figura 95: Calibración del voltaje suministrado a los motores	112
Figura 96: Frecuencia del PWM de los motores	112
Figura 97: Datos en tiempo real de los acelerómetros y giroscopios.....	112
Figura 98: Extracción del archivo de arranque del sistema operativo.....	114
Figura 99: Instalación del archivo de arranque de sistema operativo	115
Figura 100: Tarjetas de Memoria del Odroid XU4.....	115
Figura 101: Conexión computador-Odroid XU4	116
Figura 102: Opciones del Programa Minicom.....	117
Figura 103: Configuración del Puerto Serial del programa Minicom	117
Figura 104: Proceso de instalación del programa de arranque	117
Figura 105: Instalación del sistema operativo para el Odroid XU4.	118
Figura 106: Posición de la eMMC en la placa Odroid XU4	119
Figura 107: Conexión de los dispositivos al Odroid XU4.	119
Figura 108: Diagrama de flujo para almacenar la señal de control del RC	122
Figura 109: Datos almacenados en el archivo .txt	123
Figura 110: Descarga del programa c_uart_interface_example	124
Figura 111: Pseudocódigo para la extracción de fotogramas	125
Figura 112: Selección de muestreos correspondientes a los fotogramas..	126
Figura 113: Pseudocódigo en VBA para eliminación de tramos	127
Figura 114: Pseudocódigo para la transformación de fotogramas.....	129
Figura 115: Estimación de la planta.....	130
Figura 116: Estimación de la planta basada en la acción de control	131
Figura 117: Pseudocódigo para la transformación de la función	132
Figura 118: Pseudocódigo para la obtención de la función	133
Figura 119: Estimación de la planta basada en el filtro de Kalman.....	134
Figura 120: Pseudocódigo del programa de estabilización de video	140
Figura 121: Pruebas de carga del dron.....	143
Figura 122: Unión de la gimbal con la placa base total.....	144
Figura 123: Disposición de los dispositivos en la placa base total.....	145

Figura 124: Unión del sistema de estabilización de video al dron	145
Figura 125: Peso de la batería.....	146
Figura 126: Deformación de los amortiguadores	146
Figura 127: Calibración de los sensores del Pixhawk.....	147
Figura 128: Calibración de los sensores del Pixhawk.....	148
Figura 129: Programa c_uart_interface_example	149
Figura 130: V6 frame 6 y 7.	152
Figura 131: V6 frame 7 y 8.	152
Figura 132: V3 frame 11 y 12.....	153
Figura 133: V6 frame 47 y 48.....	153
Figura 134: V6 frame 48 y 49.....	154
Figura 135: V4 frame 54 y 55.....	154
Figura 136: V6 frame 81 y 82.....	155
Figura 137: V6 frame 82 y 83.....	155
Figura 138: V5 frame 84 y 85.....	156
Figura 139: V6 frame 103 y 104.....	156
Figura 140: V6 frame 104 y 105.....	157
Figura 141: V2 frame 3 y 4.	157
Figura 142: Video de cámara estática sin estabilizar, frame 119 y 120.	158
Figura 143: Video de cámara con gimbal sin estabilizar, frame 133 y 134.	158
Figura 144: Video offline del algoritmo de estabilización, frame 133 y 134.	159
Figura 145: Comparación del RMSE de los fotogramas originales.....	161
Figura 146: Comparación del RMSE total de los videos originales.....	162
Figura 147: Porcentaje de mejora del uso del algoritmo de estabilización	163
Figura 148: Pseudocódigo del programa “mse”, que compara fotogramas	164
Figura 149: Comparación del PSNR de los fotogramas originales	166
Figura 150: Pseudocódigo del programa “psnr”, que compara fotogramas	167
Figura 151: Comparación del ITF total de los videos originales.....	169
Figura 152: Mejora del uso del algoritmo de estabilización	169

ÍNDICE DE CUADROS

Cuadro 1	Especificaciones técnicas STORM Drone 6 V3	2
Cuadro 2	Especificaciones técnicas de las baterías Floureon.....	3
Cuadro 3	Comparación de motores BLDC con otros motores.....	34
Cuadro 4	Especificaciones de controladores de vuelo	49
Cuadro 5	Características de las estaciones de control	51
Cuadro 6	Especificaciones técnicas motor DYS BGM4108-130T-8.5	64
Cuadro 7	Especificaciones técnicas motor Brushless Motor 2804-100T	65
Cuadro 8	Propiedades del acrílico	66
Cuadro 9	Características del acrílico	67
Cuadro 10	Selección del controlador de Gimbal.....	71
Cuadro 11	Especificaciones técnicas de la cámara ComQtech.....	75
Cuadro 12	Especificaciones técnicas de la cámara G-vision Inkfish A1	76
Cuadro 13	Especificaciones técnicas de SBC's.....	81
Cuadro 14	Propiedades de materiales para impresión 3D	88
Cuadro 15	Cargas del sistema.....	150

ÍNDICE DE TABLAS

Tabla 1 Métricas de Evaluación.....	7
Tabla 2 Parámetros PID y motores para cámara ComQtech	113
Tabla 3 Parámetros PID y motores para cámara G-vision Inkfis h A1.....	113
Tabla 4 Carga máxima soportada por el dron	142
Tabla 5 Calculo del tiempo de funcionamiento del sistema.....	151
Tabla 6 RMSE por fotograma.....	159
Tabla 7 RMSE total y Porcentaje de mejora.....	161
Tabla 8 PSNR por fotograma	165
Tabla 9 ITF y Porcentaje de mejora	168
Tabla 10 Costos de materiales mecánicos.....	172
Tabla 11 Costos de materiales eléctricos y electrónicos	172
Tabla 12 Costos de materiales para estructura	172
Tabla 13 Accesorios eléctricos y electrónicos	173
Tabla 14 Accesorios para estructura	173
Tabla 15 Costos de accesorios	174
Tabla 16 Costos directos totales de materiales.....	174
Tabla 17 Mano de obra directa.....	175
Tabla 18 Costo de mano de obra directa	176
Tabla 19 Costos totales directos	176
Tabla 20 Costos totales indirectos.....	177
Tabla 21 Costo total del proyecto	177

RESUMEN

El principal objetivo de este trabajo es desarrollar un sistema de estabilización para cámaras a bordo de mini RUAVs que minimice el efecto de los movimientos fantasma en videos capturados, como parte del Proyecto de Investigación VisualNavDrone 2016-PIC-024 de la Universidad de las Fuerzas Armadas ESPE dirigido por el Dr. Wilbert G. Aguilar. Los capítulos 1 y 2 hablan sobre los conceptos básicos de temas que conforman al proyecto y la teoría que interviene en la misma. Contiene temas como protocolos de comunicación entre dispositivos y elementos que permiten el acceso a parámetros para la estabilización de video mediante el código que se emplea, entre otros. El capítulo 3 se centra en el desarrollo del sistema de estabilización, donde se involucra la parte mecánica que soporta los diferentes componentes electrónicos, la programación e interrelación entre el dron, el ordenador a bordo y la gimbal SimpleBGC. El programa funciona mediante conexión Wifi entre la estación de tierra y el Odroid XU4, el cual recibe los mandos de ejecución. Después de la implementación se realizaron pruebas funcionales del video resultante, donde se puede verificar que el video procesado se aproxima al movimiento real, es más efectivo y tiene mejor rendimiento de estabilización que el video original. El análisis económico detalla el coste invertido en el cumplimiento de este proyecto. El documento finaliza con conclusiones y recomendaciones para futuros trabajos.

PALABRAS CLAVE:

- UAV MULTIROTOR
- GIMBAL
- ESTABILIZACIÓN DE VIDEO
- ESTIMACIÓN DE MOVIMIENTO
- SOFTWARE EMBEBIDO

ABSTRACT

The main goal of this work is the development of a stabilization system for cameras onboard mini Rotor UAVs that minimizes the effect of the undesired movements on captured videos. Chapters 1 and 2 talk about the basic concepts of subjects that are related to the project and related theory. There are different topics like communication protocols between devices and elements for accessing to video stabilization parameters. Chapter 3 focuses in the development of the stabilization system, including the mechanical structure, electronic components, onboard processor, gimbal, communication interface, and algorithms. The Ground Station is communicated with the Odroid XU4 by Wi-Fi connection, for sending the commands of program execution. Experimental result show that the obtained video is more stable than the original without reducing the motion fidelity. The system is optimal because depends only on the last frame. The economic analysis details the implementation cost of this project. The document ends with conclusions and future works.

KEYWORDS:

- MULTIROTOR UAV
- GIMBAL
- VIDEO STABILIZATION
- MOTION ESTIMATION

EMBEDDED SYSTEM

CAPÍTULO 1

INTRODUCCIÓN

En el presente capítulo, se presenta la necesidad del desarrollo de un Sistema de Estabilización de video para minimizar el efecto de los movimientos fantasma. De igual manera, se justifica el desarrollo e importancia del presente trabajo. Se define el objetivo general y los objetivos específicos. Por último, se presenta la metodología utilizada para la realización óptima del trabajo.

1.1. Planteamiento del Problema

Desde los años noventa, ha habido un creciente interés en el desarrollo de sistemas aéreos no tripulados (UAS), siendo utilizados en aplicaciones como vigilancia aérea, reconocimientos e inspección de ambientes complejos y peligrosos. (Kendoul, 2011)

Los dispositivos de captura de video que se han empleado para estos trabajos, han sido cámaras digitales, cuyo uso se ha extendido en años recientes. Pero como resultado de su montaje, se ha exhibido un gran aumento de sacudimientos en el video resultante debido a las vibraciones generadas por el efecto del viento sobre el vehículo aéreo. Adquiriendo a partir del movimiento no amortiguado, el efecto “Rolling Shutter”, donde cada fila de la imagen está expuesta en tiempos ligeramente diferentes, a proyectar una oscilación inestable en la salida de video. (Karpenko, Jacobs, Baek, & Levoy, 2011)

Adicional a esto, se han hallado movimientos fantasma, los cuales son un falso desplazamiento generado en parámetros de escala y/o traslación debido a la compensación y suavizado de movimientos de alta frecuencia, donde a veces el proceso de suavización de movimientos, remueve movimientos reales y/o

introduce un retraso en ellos. Ambos casos son definidos como movimientos fantasma.(Aguilar & Angulo, 2014b)

Por esta razón que es necesario desarrollar un sistema de estabilización, que permita una óptima distinción de video, y como consecuencia de esto, lograr un mejor desempeño en la aplicación a la que se pone en práctica.

1.1.1. Especificaciones Técnicas

El dron utilizado para el montaje del sistema de estabilización de video es “STORM Drone 6 V3 NAZA Lite AT9”, cuyas especificaciones se presentan en el Cuadro 1.

Cuadro 1

Especificaciones técnicas STORM Drone 6 V3

Autopilot	Sistema DJI NAZA-M Lite GPS
Sistema de Radio Control	RadioLink 2.4Ghz AT9 w/ Receptor R9D 9-Ch
Peso	850g (sin batería)
Peso con batería	1270g (con batería de 11.1V 5500mAh 30C)
Dimensiones	450mm x 375mm x 215mm (sin antena GPS)
Motor	Motor de alto poder sin escobillas Storm M2208 1260kv
Hélices	Gemfan 8045C de fibra de carbón relleno con nylon (3 x CW y 3 x CCW)
Batería	Storm 11.1V 5500mah 30C
Tiempo de Vuelo	19 min (@11.1V 5500mAh, sin equipo de filmación)
Rango de Control del Transmisor RadioLink AT9	0.9km en tierra, 1.5km en el aire

Fuente: [Helipal, 2016]

La batería que se utilizó para la energización de los actuadores es “Floureon 11.1V 3S Lipo Battery 5500mAh 35C (30C-40C Discharge Rate) Deans Plug for

DJI F450 Quadcopter RC Helicopter Airplane Hobby”, cuyas especificaciones se presentan en el Cuadro 2.

Cuadro 2

Especificaciones técnicas de las baterías Floureon

Peso	402.2 g
Dimensiones	15.5 x 4.5 x 2.5 cm
Marca	Floureon
Voltaje	11.1 Voltios
Capacidad	5500mAh
Tasador de descarga continua	35C

Fuente: [Floureon, 2016]

1.2. Antecedentes

La historia de los primeros UAVs (vehículos aéreos no tripulados) empezó justo después del primer vuelo pilotado por el hombre, remontando a los años de 1916 cuando Elmer Sperry de la compañía Sperry Gyroscope, junto con Peter Hewitt, desarrollaron exitosamente un sistema de control automático para el bote volador Curtiss.

Después de que los Estados Unidos se involucraran en la segunda guerra mundial en 1917, se inició el desarrollo de la bomba voladora. El contrato fue otorgado a Charles Kettering para desarrollar éstas aeronaves no tripuladas. Se construyeron burdamente veinticinco biplanos pequeños y baratos, nombrados "Kettering Bug" en 1918. Vuelos que cumplían su objetivo, a la final siempre se estrellaban, pero el éxito adquirido era dramáticamente suficiente para la armada, que compró 75 aeronaves más. El vuelo más extenso realizado por un "Kettering Bug", fue de solo 16 millas, y jamás fueron pilotados operacionalmente en su tiempo de vida.

La era moderna de los UAVs empezó a finales de 1959. Apremiada por el derribamiento de la aeronave Power U-2 de Gary Francis sobre Rusia, donde el primer UAV moderno consistía en el posicionamiento de cámaras en blancos teledirigidos.

En Julio de 1960, la fuerza aérea de Estados Unidos asignó doscientos mil USD al proyecto de nombre código “Red Wagon”, cuyo contrato perteneció a la Compañía Aeronáutica Ryan, que consistía en mejorar un modificado blanco teledirigido “Firebee” para que desempeñe misiones de vigilancia fotográfica a control remoto.(Cook, 2007)

A inicios de los años setenta, la visión computacional fue vista como un componente de la percepción visual de una agenda ambiciosa para imitar la inteligencia humana y para dotar a los robots con comportamientos inteligentes. En ese tiempo, se creía que algunos de los primeros pioneros de la inteligencia artificial y robótica, tales como el MIT, Stanford y CMU, podían resolver el problema de la “entrada visual” fácilmente mientras resolvían problemas difíciles tales como un razonamiento y planeamiento de alto nivel.

De acuerdo a una historia bien conocida de 1996, Marvin Minsky del MIT solicitó a su estudiante Gerald Jay Sussman a dedicar sus vacaciones de verano a enlazar una cámara a su computador, haciendo que la computadora describiera lo que veía, sabiendo en la actualidad que ese dilema era más difícil que de lo que parecía. (Boden 2006, p. 781)

Lo que distinguió a la visión computacional del ya existente campo de procesamiento digital de imagen (Rosenfeld & Pfaltz, 1966), fue el deseo de recuperar la estructura tridimensional del mundo a partir de imágenes y encaminarlos hacia un entendimiento completo de una escena.

En los ochentas, la atención fue enfocada en técnicas matemáticas sofisticadas para desempeñarse en imágenes cuantitativas y análisis de escenas.

Pirámides en la imagen empezaron a ser utilizadas ampliamente para realizar tareas tales como combinación de imágenes y buscar correspondencias (Anandan, 1989; Burt & Adelson, 1983a, 1983b; Rosenfeld & Pfaltz, 1966). Se desarrollaron varias versiones de pirámides usando el concepto de procesamiento escala-espacio (Lindeberg, 1990; A. Witkin, 1984; Andrew Witkin, Terzopoulos, & Kass, 1987).

A finales de los ochenta, la investigación en mejor detección de esquinas y contornos fue muy activa durante este periodo (Canny, 1986; Nalwa & Binford, 1986), incluyendo la introducción de rastreadores de contornos que evolucionaban dinámicamente, tales como “snakes” (Kass, Witkin, & Terzopoulos, 1988), y el modelo físico de tres dimensiones (D. Terzopoulos A. Witkin & D Terzopoulos, 1988; D Terzopoulos & Fleischer, 1988; Demetri Terzopoulos, Witkin, & Kass, 1987).

En los noventa, emergió el uso de invariantes descriptivos para el reconocimiento, el cual evolucionó en un esfuerzo para resolver el problema de la estructura de movimiento. Mucha de la actividad inicial se dirigió a las “reconstrucciones proyectivas”, lo que no requirió de conocimiento en calibración de cámaras (Faugeras, 1992; R I Hartley, 1992; Richard I. Hartley, 1994).

En la década pasada se ha continuado profundizando en la interacción entre los campos de visión y gráficos, en particular en muchos temas introducidos bajo la rúbrica de renderizado de imágenes, tales como “image stitching”, captura y renderizado en el campo de luz, y en captura de imagen en HDR (alto rango dinámico) a través de la exposición de horquillado (Debevec &

Malik, 1997), nombrado “fotografía computacional” dando reconocimiento al constante uso de dichas técnicas en la fotografía digital de todos los días.

La tendencia actual que ahora demanda mucha investigación de reconocimiento visual en nuestra comunidad, es la aplicación de técnicas de aprendizaje en máquinas sofisticadas para solventar problemas de visión computacional.(Szeliski, 2010)

1.3. Justificación

Los problemas que se presenta durante la adquisición de imágenes con cámaras a bordo del sistema para la navegación, autónoma o teleoperada de los UAVs en aplicaciones de rescate, transporte o vigilancia es la presencia de movimientos indeseados entre fotogramas consecutivos y los denominados movimientos fantasma.

Se consideran movimientos indeseados aquellos generados durante el vuelo como resultado de complejas características aerodinámicas del UAV, y que el usuario no tuvo la intención de realizarlos. Se pueden distinguir entre dos tipos de movimientos generados en las cámaras:

- “Weak Shaking” o Sacudimiento débil (aproximadamente de $\pm 10^\circ$ de variación en las direcciones horizontal y vertical y/o algunas pequeñas fracciones de Hertz).
- “Strong Shaking” o Sacudimiento fuerte (más de $\pm 10^\circ$ de variación en las direcciones horizontal y vertical y/o décimos de Hertz).”(Drahanský, Orság, & Hanáček, 2010)

En el caso de Weak Shaking, el problema puede ser resuelto usando solo estabilización de imagen por software (digital). Mientras que para el Strong Shaking, no es posible lograr una estabilización únicamente por software, sino que se requiere de un hardware adicional. Éste hardware puede basarse en un

una unidad de servomotor o una unidad neumática/hidráulica, que sea capaz de compensar el movimiento del sistema de la cámara en la dirección opuesta.”(Drahanský et al., 2010)

Hay múltiples algoritmos que se han diseñado para contrarrestar los movimientos indeseados de Weak Shaking, tales como “L1 Optimal” y “Parrot’s Director Mode”, que pueden ser utilizados para post-procesamiento de videos. En el artículo “Real-time video stabilization without phantom movements for micro aerial vehicles” de los autores Aguilar W. & Angulo, se presenta un algoritmo de estabilización capaz de ser aplicado en tiempo real y minimizar el efecto de los movimientos fantasma. Este algoritmo ha sido comparado con “L1 Optimal” usando como métricas de evaluación el ITF¹ y el RMSE², obteniendo los resultados de la Tabla 1 y la Figura 1.

- El ITF es una medida ampliamente utilizada en la literatura para determinar el grado de estabilidad de un video, entre mayor sea su valor significa que el video es más estable.
- El RMSE compara el movimiento estimado con el movimiento observado, es decir que es una métrica de evaluación del realismo del movimiento, entre menor sea su valor el movimiento estimado se asemejará más al movimiento observado.

Tabla 1

Métricas de Evaluación

<i>Nombre de Video</i>	<i>Métrica de Evaluación</i>	<i>Original</i>	<i>L1-Optimal</i>	<i>Aproximación</i>
Video 1	ITF (dB)		19.62	19.48
	RMSE	14.09	0.046	0.028
Video 2	ITF (dB)	13.43	19.57	19.52

¹ ITF: Fidelidad de Transformación entre Fotogramas

² RMSE: La Raíz del Error Medio Cuadrático

Continúa



	RMSE		0.051	0.023
Video 3	ITF (dB)		20.16	19.89
	RMSE	14.65	0.047	0.029
Video 4	ITF (dB)		21.24	21.12
	RMSE	16.96	0.036	0.017

Fuente: (Aguilar & Angulo, 2014b)

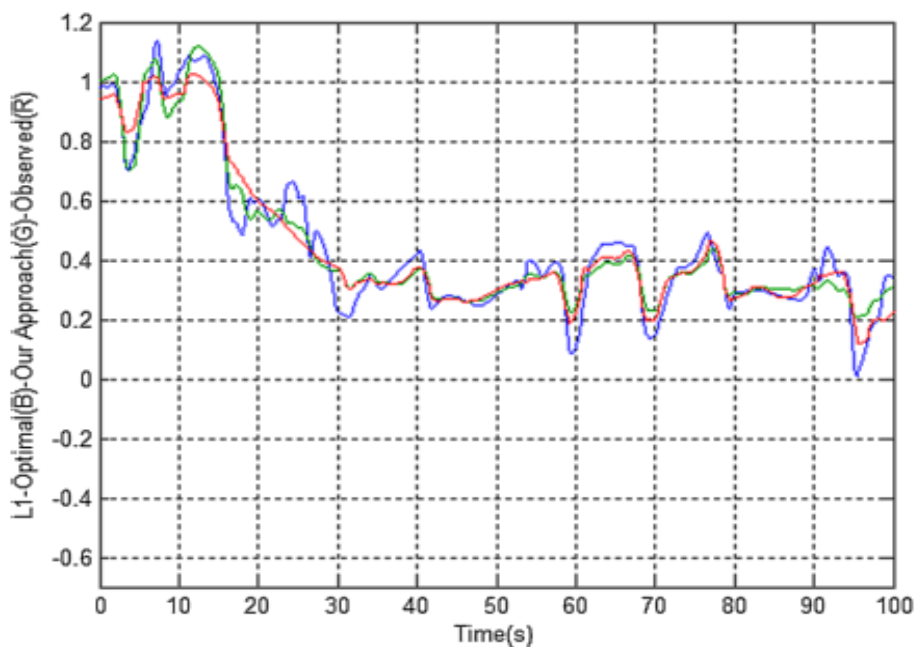


Figura 1 Pruebas de Estabilización del algoritmo de estabilización (L1-Optimal [Azul], Algoritmo en Tiempo Real [Verde], Señal Original [Rojo])

Fuente: (Aguilar & Angulo, 2014b)

Se puede observar que el algoritmo en tiempo real con respecto a L1-Optimal, respecto al ITF, es inferior por un valor promedio de 0.72%, lo que indica que tiene menor estabilidad que “L1 Optimal”, y en función del RMSE, es inferior en un valor promedio de 46.28%, es decir que el movimiento estimado se asemeja más al movimiento observado. (Aguilar & Angulo, 2014b)

En el tema de la cámara existe un efecto óptico llamado “Rolling Shutter”, que se origina debido al movimiento no atenuado del dispositivo, combinado con la captura de imágenes, en la que cada fila de la ilustración se expone a desplazarse ligeramente a diferentes intervalos de tiempo, resultando en un desagradable “bamboleo” en el video resultante.

Para la corrección del mismo, se utilizan técnicas digitales y mecánicas, que se basan en giroscopios para medir la rotación de la cámara en altas frecuencias y con alta precisión, siendo el sistema Mecánico de Estabilización de Imagen (MIS), el de uso más común. Éstas técnicas se fundamentan en hallar la orientación relativa del giroscopio con respecto a la cámara, analizando la relación de sus tres ejes de rotación. (Karpenko et al., 2011)

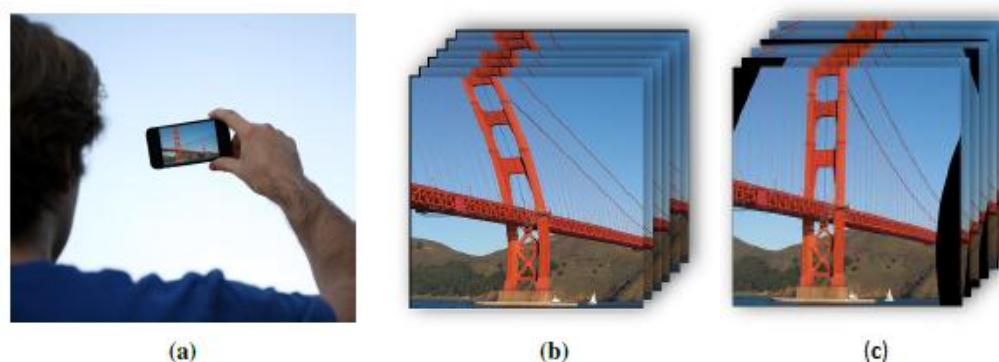


Figura 2 “Rolling Shutter” o bamboleo
a) Video Capturado, b) “Rolling Shutter”, c) Imagen procesada

Fuente: (Karpenko et al., 2011)

El vehículo que transportará el sistema de estabilización es un dron aéreo, por lo tanto, toda la estructura estará sujeta a perturbaciones generadas por el viento, por lo que habrá información incierta del comportamiento del dron durante el vuelo debido a la compleja configuración aerodinámica. Debido al efecto “Rolling Shutter”, se empleará una gimbal de tres ejes de rotación. Se implementará un algoritmo de estabilización de video similar al que se comparó con el algoritmo L1 Optimal, debido a su funcionalidad en tiempo real, lo que

permitirá su implementación en aplicaciones futuras, las cuales pueden ser vigilancia, rescate o transporte, como se enunció anteriormente

Para la obtención del movimiento deseado en la secuencia de video, que suprima la inestabilidad de altas frecuencias, se utilizará el Filtro de Kalman (KF), ya que es uno de los métodos más utilizados para rastreo y estimación debido a que es óptimo, simple, y robusto.

El vehículo sobre el cual se montará el sistema de estabilización mecánico es un RUAV de mini escala, por tanto, el conjunto gimbal-cámara deberá liviano.

Realizado este sistema se podrá implementar en drones del proyecto de investigación VisualNavDrone, que contribuirá a futuras investigaciones del mismo.

1.4. Objetivos

1.4.1. General

Desarrollar un sistema de estabilización para cámaras a bordo de mini RUAVs que minimice el efecto de los movimientos fantasma en videos capturados.

1.4.2. Específicos

- Diseñar una estructura mecánica de soporte para un sistema de estabilización para cámaras capaz de ser montada en la base de un RUAV de mini escala, para la compensación de los movimientos indeseados.
- Seleccionar los elementos mecánicos y electrónicos adecuados para el acondicionamiento del vehículo con base en trabajos relacionados en la literatura de RUAVs.

- Elaborar un sistema de control servo visual, que permita estabilizar los movimientos de la cámara mediante la implementación de un algoritmo de estabilización en tiempo real que minimice el efecto de los movimientos fantasma.
- Obtener un modelo no lineal del sistema RUAV-Gimbal.
- Validar el diseño de la estructura mecánica mediante información visual e inercial obtenida experimentalmente.
- Evaluar el sistema de estabilización mecánico propuesto mediante el RMSE del movimiento del gimbal respecto al movimiento real y el ITF.

1.5. Alcance del Proyecto

El Proyecto a desarrollarse tendrá las siguientes capacidades y alcances.

1.5.1. Diseño Mecánico

Se dispondrá de motores dc sin escobillas, los cuales brindan precisión en el control del movimiento angular, para la rotación el mecanismo. Se diseñará la base sobre la cual estarán montados todos los elementos del sistema, y la unión entre el sistema de estabilización mecánico de la cámara con la base. En caso de ser necesario se usará algún tipo de sistema de transmisión de movimiento.

1.5.2. Diseño Electrónico y Eléctrico

Para la adquisición de datos se contará con una unidad de medición inercial (IMU), que provea con la medición de la aceleración lineal y velocidad angular, parámetros necesarios para el control del Gimbal en problemas de estabilización de imagen.

El procesamiento de datos se llevará a cabo en una placa Odroid-XU4, la cual permite la obtención y control de los parámetros para controlar la Gimbal, ya que cuenta con puertos/pines de entrada y salida de datos.

Para el control de bajo nivel del sistema de vuelo del vehículo, se utilizará el Autopilot 3DR Pixhawk, que es un controlador comercial estable y robusto, ampliamente utilizado en distintos drones de mini escala.

Se usará baterías de litio para maximizar la autonomía energética durante los vuelos.

1.5.3. Diseño del sistema de control

Para el control de rotación para los ejes Roll, Pitch y Yaw, se usará un controlador de gimbal junto con la señal de control que enviará el algoritmo de estabilización de video a través del Odroid XU4, los cuales utilizan la IMU para la adquisición de parámetros de rotación del gimbal.

Se extrapolará un algoritmo de estabilización de video en tiempo real que elimine los movimientos indeseados y minimice los movimientos fantasma manteniendo el realismo del movimiento. Dentro de este algoritmo estarán contemplados los filtros necesarios, así como las técnicas estadísticas para minimizar errores.

Los programas y software que se dispondrán para la implementación, análisis y simulación del proyecto, son: SolidWorks, MATLAB, OpenCV, Linux, C y C++.

1.6. Metodología

El proyecto se desarrollará en base a las etapas de análisis, diseño, implementación y evaluación de resultados de (González, 2015).

En la etapa de análisis, se pretende estudiar las posibles alternativas para el diseño del sistema, así como la estructura de estabilización, los actuadores, la cámara a utilizarse, el sistema de control, y las limitaciones de cada una, para posteriormente elegir la que mejor rendimiento y desempeño ofrezca.

Se realizará:

- Observando videos del funcionamiento de cámaras en drones y sistemas de estabilización.
- Leyendo artículos relacionados a movimientos fantasma, sistemas de control de alto procesamiento, estructuras gimbal, motores eléctricos y cámaras.
- Consultando a profesionales que tengan conocimiento sobre la elaboración de estructuras mecánicas, sistemas de control y estabilización de video.

En la etapa de diseño, se determinará la forma específica en que cada componente y sistema escogido se posicionará y aplicará para lograr los objetivos planteados, con la utilización de las herramientas de simulación como SolidWorks, y de cálculo como MATLAB, además, el sistema operativo Linux, y software de programación como Python, C, C++ y OpenCV.

Al concluir la etapa de diseño, en la etapa de implementación, se procederá a la construcción de la estructura y el desarrollo del sistema de estabilización.

Culminando con la comprobación del desempeño y efectividad del sistema mediante la realización de pruebas de campo.

1.7. Estructura del documento

El presente documento se estructura en 7 capítulos y los anexos.

- El primer capítulo hace referencia a las generalidades del proyecto, define objetivos y muestra el alcance del mismo.
- El segundo capítulo presenta la descripción de los programas, componentes y conceptos utilizados en el sistema de estabilización.
- El tercer capítulo abarca lo referente a las especificaciones de diseño y selección de componentes e implementación de los mismos en la estabilización de video en el RUAV.

- El cuarto capítulo hace referencia a las pruebas realizadas del sistema, la comprobación del desempeño mediante ITF y RMSE, presentado los resultados obtenidos.
- En el capítulo cinco, se presenta un análisis económico y financiero, el cual permitirá demostrar que el proyecto es realizable y rentable.
- El capítulo seis, se presentan las conclusiones y recomendaciones que se generaron al realizar el diseño, construcción e implementación de sistema de estabilización del presente trabajo.
- Por último, se adjunta como anexos los planos del sistema.

CAPÍTULO 2

MARCO TEÓRICO

En el presente capítulo, se da a conocer información referente a los sistemas aéreos no tripulados, las plataformas inercialmente estabilizadas, los ordenadores on-board, los protocolos de comunicación de estos dispositivos, y la estabilización de video.

2.1. Sistemas Aéreos no tripulados a base de Rotores (RUAS)

El segmento de las aeronaves a base de rotores está evolucionando rápidamente y dinamizan al sector del mercado de las UAS. Los RUAS son adecuados para aplicaciones civiles en comparación a aeronaves tripuladas y UAS de alas fijas. Aplicaciones potenciales de las UAS incluyen inspección de infraestructuras, imposición de la ley, vigilancia de fronteras costeras y tránsito, manejo de crisis y desastres, y agricultura y silvicultura. La mayor parte de estas aplicaciones requieren de bajas altitudes de vuelo con planeamiento y la capacidad de despegar y aterrizar verticalmente (VOTL). (Kendoul, 2011)



Figura 3: Categorías de los RUAS

Fuente: (Kendoul, 2011)

2.1.1. Clasificación de las RUAS

Los RUAS pueden ser categorizados en cinco clases, basadas en atributos tales como tamaño y carga útil

2.1.1.1. 1ra Categoría

Helicópteros no tripulados de escala completa o helicópteros autónomos opcionalmente tripulados. La característica principal de esta categoría de RUAS es su significativa carga útil, resistencia y rango, además de la posibilidad de llevar un piloto a bordo del vehículo de una manera segura, de forma que provee un excelente banco de pruebas para vuelos con presencia de peligro. Un buen ejemplo de esta categoría es el helicóptero Boeing Unmanned Little Bird (ULB).

2.1.1.2. 2da Categoría

Helicópteros UAS de mediana escala que están disponibles en plataformas autónomas o semiautónomas que tienen una carga útil de más de 10 kg, siendo el peso total de la aeronave de 30kg. Algunos ejemplos son el Yamaha RMAX (Japón), Shiebel S-100(Austria) y el Rotomotion SR500(USA). Estas plataformas son generalmente bien diseñadas con un grado de fiabilidad.

2.1.1.3. 3ra Categoría

RUAS de escala pequeña están basadas en helicópteros de Radio Control (RC), con piloto automático integrado opcionalmente. Tienen una carga útil de 2 a 10 kg y su peso total es menor a 30 kg. Helicópteros tales como el Vario Benzin Trainer, Bergen Industrial Twin, Rotomotion SR100, y entre otros, caen en esta categoría. Se caracterizan por ser de bajo costo y disponibilidad, pero requieren de un mantenimiento considerable y trabajo ingenieril para convertirlos de helicópteros RC a UAS dependientes.

2.1.1.4. 4ta Categoría

Los Mini RUAS son portables y pueden volarse en entornos abiertos, así como también en espacios cerrados. Su carga útil es de menos de 2kg y su peso total puede estar entre los cientos de gramos, a unos pocos kilogramos. La mayor parte de ellos se energizan eléctricamente, con un tiempo de vuelo que varía de 5 minutos a 1 hora, dependiendo de la carga útil. Su bajo costo, mantenimiento fácil y operabilidad segura los hacen excelentes bancos de prueba para la investigación.

2.1.1.5. 5ta Categoría

Vehículos Micro Aéreos (MAVs) con menos de 100g de carga útil, tales como el micro robot volador de Epson. Sensores de navegación y aviónicas son difíciles de llevar en estas máquinas. Los retos de investigación incluyen soluciones de sensamiento y navegación basados en principios bio miméticos.

Estos RUAS son principalmente diseñados para aplicaciones en ambientes cerrados y pueden despegar y ser recuperados con la palma de la mano.

Las categorías I, II, y III son generalmente adecuados para aplicaciones en entornos abiertos y requieren de un área para despegar y aterrizar, y en el caso de los mini y micro UAS de las categorías IV y V son plataformas de multirrotores que pueden volar tanto en entornos abiertos como cerrados, y pueden despegar desde espacios reducidos. (Kendoul, 2011)

Sin importar la categoría de las UAV, estas incluyen dos sistemas diferentes integrados. El uno es el sistema Autopilot, conjunto de actuadores que realiza el control de movimiento y circuitos de control que hacen que estos actuadores se muevan. El segundo es el director de vuelo (FD), es el cerebro del sistema Autopiloto.(Federal Aviation Administration, 2009)

Actualmente, no existe una definición establecida entre Autopilot y controlador de vuelo, ya que muchas personas usan estas dos palabras indistintamente. Uno se refiere a un sistema que mantiene al UAV estable mientras vuela y el otro a un sistema más completo que permite al UAV realizar vuelos autónomos. (Dronetrest, 2015)

Por ese motivo, en este documento referiremos al Autopilot como aquel sistema completo que permite realizar vuelos autónomos.

2.1.2. Autopilot

Es un sistema MEMS cuyo objetivo principal es guiar constantemente al UAV para que siga las rutas de referencia o navegue a través de puntos de referencia sin la asistencia de operadores humanos. Los controladores de vuelo más completos aparte de cumplir lo antes mencionado pueden guiar al UAV en todas sus etapas incluyendo despegue, ascenso, descenso, seguir la trayectoria y aterrizaje.

Forma parte del sistema de control de vuelo del UAV, ya que se comunica con la estación de control para realizar el cambio del modo de control, recibe los datos del GPS para actualizar su posición y envía las señales de control a los motores del UAV.(Chao, Cao, & Chen, 2010)

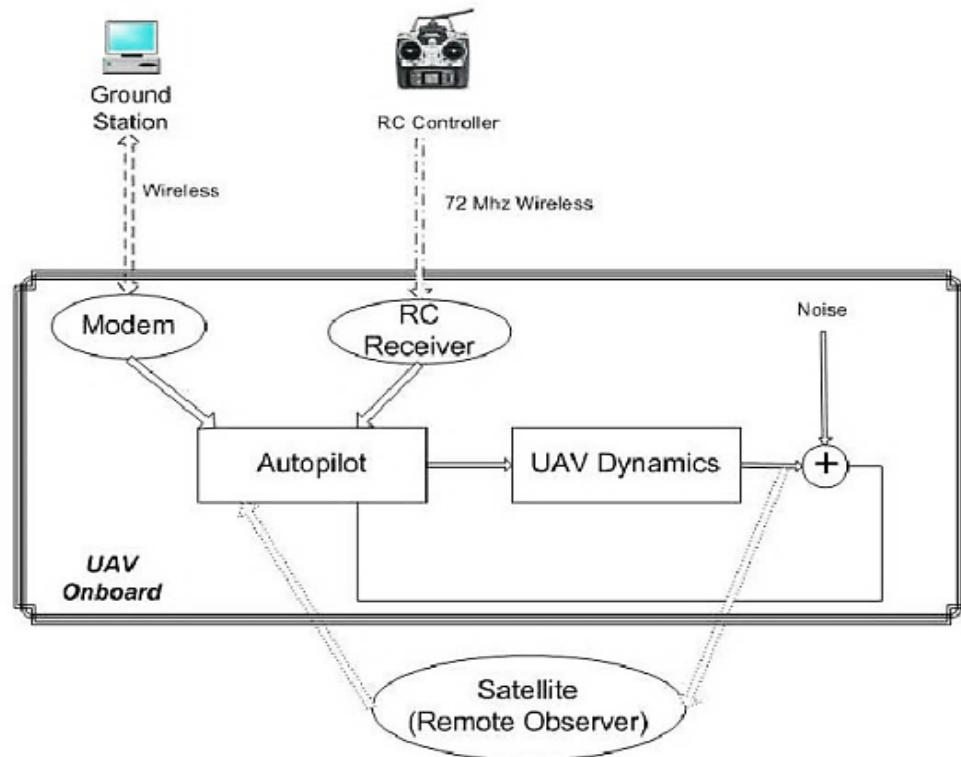


Figura 4 Sistema de control de vuelo del UAV

Fuente: (Chao et al., 2010)

La mayoría están enfocados en navegación por GPS basada en puntos de referencia. Hay dos controladores básicos para el control de vuelo del UAV: Controlador de altitud y Controlador de velocidad y dirección. El controlador de altitud sirve para que el UAV vuele a una altitud deseada incluyendo la etapa de aterrizaje y despegue. El controlador de velocidad y dirección para que el UAV vuele a través de los puntos de interés deseados. Para lograr esto se pueden usar diferentes estrategias de control incluyendo PID, Redes neuronales adaptativas, Lógica Fuzzy, entre otros. (Chao, Luo, Di, & Chen, 2009)

Este posee sensores para determinar su posición, procesadores integrados para usos de estimación y control, y circuitos periféricos para enviar el control a los motores y comunicarse con la estación de control y el RC. Debido a las limitaciones físicas de los UAV pequeños, el hardware del mismo debe poseer tamaño reducido, peso ligero y tener bajo consumo de energía. (Chao et al., 2010)

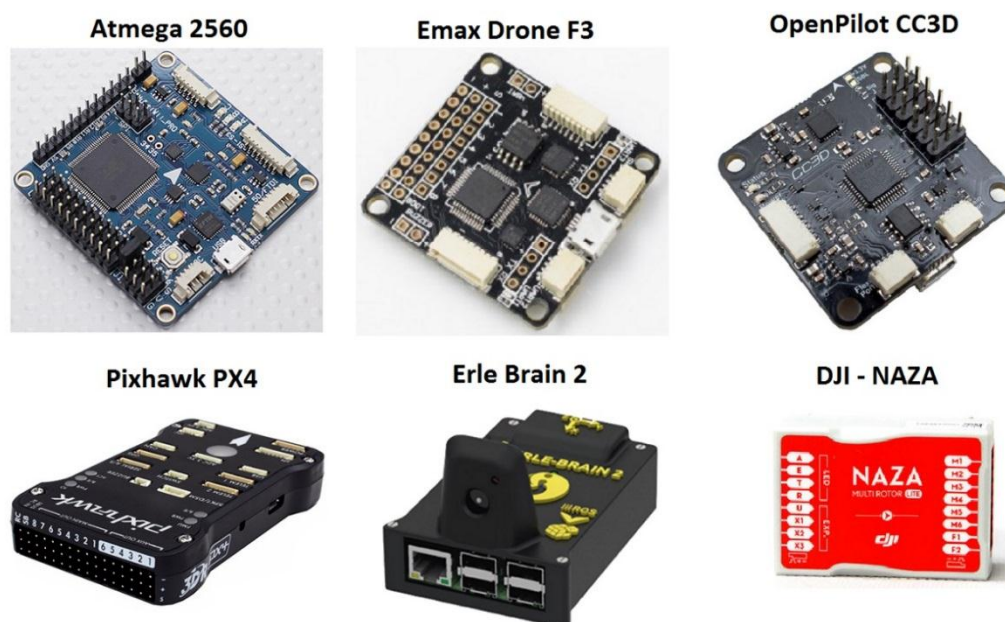


Figura 5: Ejemplo de Autopilots

La configuración y la calibración de los sensores del Autopilot se lo realiza a través de la estación de control, el cual es un software que corre en computadoras o dispositivos móviles dependiendo del tipo y modelo Autopilot. (Dronetrest, 2015)

2.1.3. Estaciones de control

Actualmente existen múltiples estaciones de control que ofrecen diferentes prestaciones. A continuación, se mencionan varias estaciones de control comúnmente usadas.

2.1.3.1. Mission Planner

Mission Planner es una aplicación completa de estación de control para el proyecto Autopilot de código abierto ArduPilot. Esta aplicación sirve para drones, helicópteros, aviones, carros, el cual es compatible con Windows, pero también se puede correr en Linux y Mac OS. Puede ser usada como utilidad de configuración o suplemento de control dinámico para el vehículo autónomo. (ArduPilot, 2016)

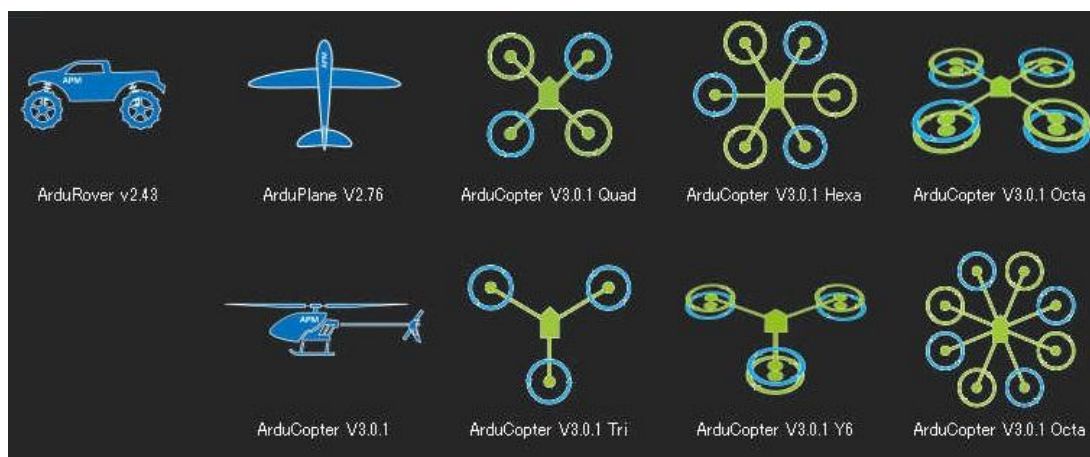


Figura 6: Vehículos de Mission Planner

Fuente: (ArduPilot, 2016)

Su principal característica es realizar la planificación de misiones utilizando mapas de Google Maps, Bing, entre otros. Permite realizar la descarga de archivos de registro de misión y analizarlos, y posee una interfaz con un simulador de vuelo. (ArduPilot, 2016)



Figura 7: Mission Planner

Con un dispositivo de telemetría adecuado esta aplicación permite controlar el estado del vehículo mientras está en funcionamiento, registra los datos de telemetría, los cuales pueden ser vistos y analizados. También se puede utilizar el vehículo en primera persona (FPV). (ArduPilot, 2016)

2.1.3.2. APM Planner 2.0

APM Planner 2.0 es una aplicación de código abierto de ArduPilot para los controladores de vuelo basados en MAVlink incluyendo APM y Pixhawk, el cual se puede ejecutar en Windows, Mac OS y Linux. Posee una interfaz más simple y tiene menos características que Mission Planner. (ArduPilot, 2016)



Figura 8: Logotipo de APM Planner

Fuente: (ArduPilot, 2016)

2.1.3.3. MAVProxy

MAVProxy es una estación de control funcional para UAV's, comúnmente utilizada por desarrolladores ya que funciona a base de comandos, este software complementa las estaciones de control gráficas como Mission Planner, APM Planner, entre otras. Una característica clave de MAVProxy es su capacidad para reenviar los mensajes del UAV a través de la red vía UDP a diferentes estaciones de control. (ArduPilot, 2016)

2.1.3.4. QGroundControl

QGroundControl proporciona un control de vuelo completo y configuración de vehículos equipados con ArduPilot o PX4. El objetivo principal es mejorar la facilidad de uso para los nuevos usuarios, así como soporte de funciones de gama alta para usuarios experimentados. Es compatible con Windows, Mac OS y Linux, y también se ejecuta en dispositivos iOS y Android, permitiendo una transmisión de video con la visualización de los instrumentos de vuelo. Su interfaz cuenta con la visualización del mapa de vuelo donde se muestra la posición del vehículo, la pista de vuelo, los instrumentos del vehículo, etc.

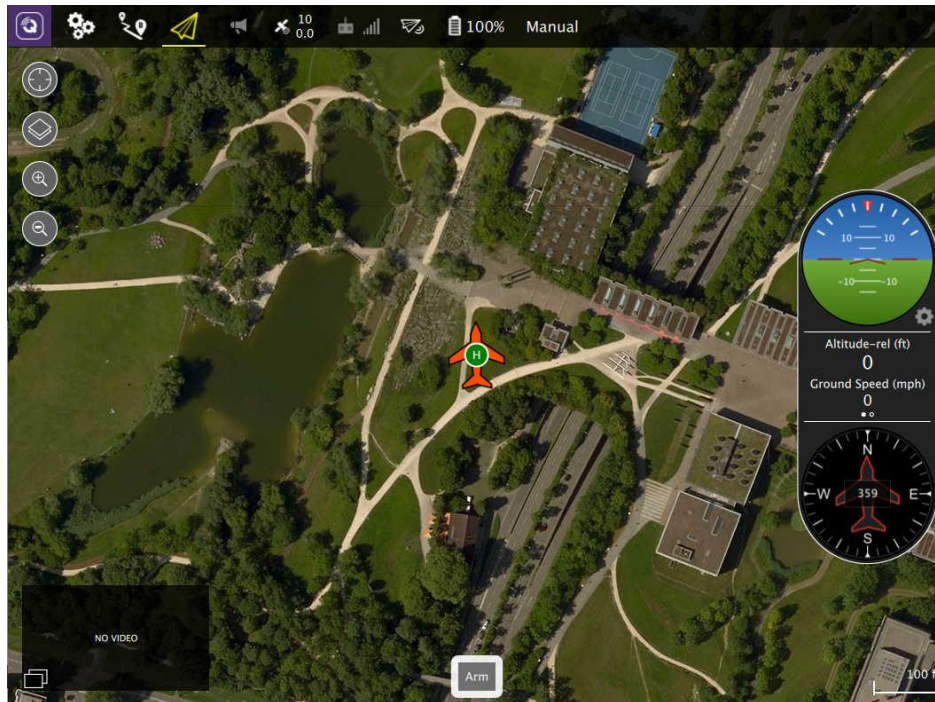


Figura 9: QGroundControl

Fuente: (QGroundControl, 2016)

Entre sus características principales también se encuentra la planificación de misiones para el vuelo autónomo, y cuenta con soporte de vuelo para cualquier vehículo capaz de usar MAVlink. Esta estación de control ha sido probada en competiciones como IMAV (The International Micro Air Vehicles Competition) y EMAV (The European Micro Aerial Vehicle Competition). (QGroundControl, 2016)

2.1.3.5. UgCS

Es una estación de control desarrollada por SPH Engineering, el cual cuenta con 3 versiones, las cuales son: open (gratuita), one y pro. El software es fácil de usar para la planificación y ejecución de misiones de todo tipo de vehículos no tripulados.



Figura 10: Versiones de UgCS

Fuente: (SPH Engineering, 2016)

Entre sus prestaciones tiene realizar misiones de largas distancias con cambios de baterías, entorno inmersivo en 3D, pantalla de telemetría. Soporta gran variedad de tipo de drones, entre los cuales se encuentran aquellos compatibles con MAVlink (ArduPilot, Pixhawk).

En su versión pro permite realizar el análisis y procesamiento de imágenes después del vuelo, tiene la opción de mapas personalizados los cuales cuentan con las zonas de acceso no permitido, cuenta con vigilancia dependiente automática (ADS-B). Su sistema soporta el monitoreo y control de múltiples operadores y múltiples drones al mismo tiempo. (SPH Engineering, 2016)



Figura 11: Planificación de misiones

Fuente: (SPH Engineering, 2016)

Debido al creciente interés del uso de cámaras a bordo de drones, en la actualidad existe gran variedad de plataformas estabilizadas inercialmente, conocidas como gimbal, la cual mediante el control de actuadores mantiene un objeto estabilizado, esta ayuda a corregir los movimientos bruscos y vibraciones.

2.2. Plataforma estabilizada inercialmente (ISP)

Las ISP generalmente están compuestas por una estructura, amortiguadores, motores y el sensor o grupo de sensores. En algunas configuraciones el sensor o la carga a estabilizar se monta directamente en la gimbal, mientras que, en otras se montan espejos u otros elementos ópticos en esta y el sensor se fija al vehículo. Generalmente, este debe estar diseñado para apuntar y estabilizarse alrededor de dos o más ejes.

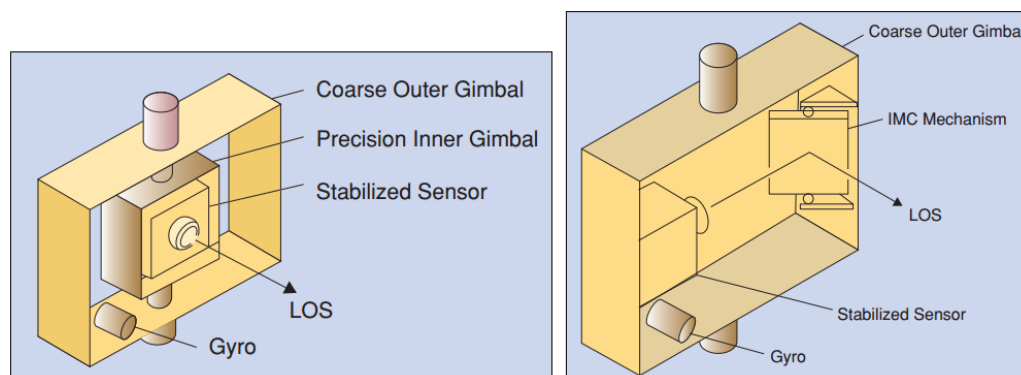


Figura 12: Ejemplos de configuraciones de ISP

Fuente: (Hilkert, 2008)

Las configuraciones electromecánicas del ISP son tan diversas como las aplicaciones para las cuales están diseñadas, el peso de la misma depende ante todo del tamaño de la carga que va a ser estabilizada, su tamaño y peso aumentan dependiendo del número de ejes que se le adicione a esta. Aunque los requerimientos de la ISP varían dependiendo de la aplicación, su objetivo principal es mantener el control de la línea de visión (LOS) de un objeto con relación a otro objeto o al espacio inercial.

Las causas de rotación inercial de la línea de visión de una plataforma estabilizada se pueden dar ya sea por una perturbación del par, flexibilidad en el sistema, o una entrada errónea a los actuadores de la gimbal, las cuales se pueden ver en la **Figura 13.**(Hilkert, 2008)

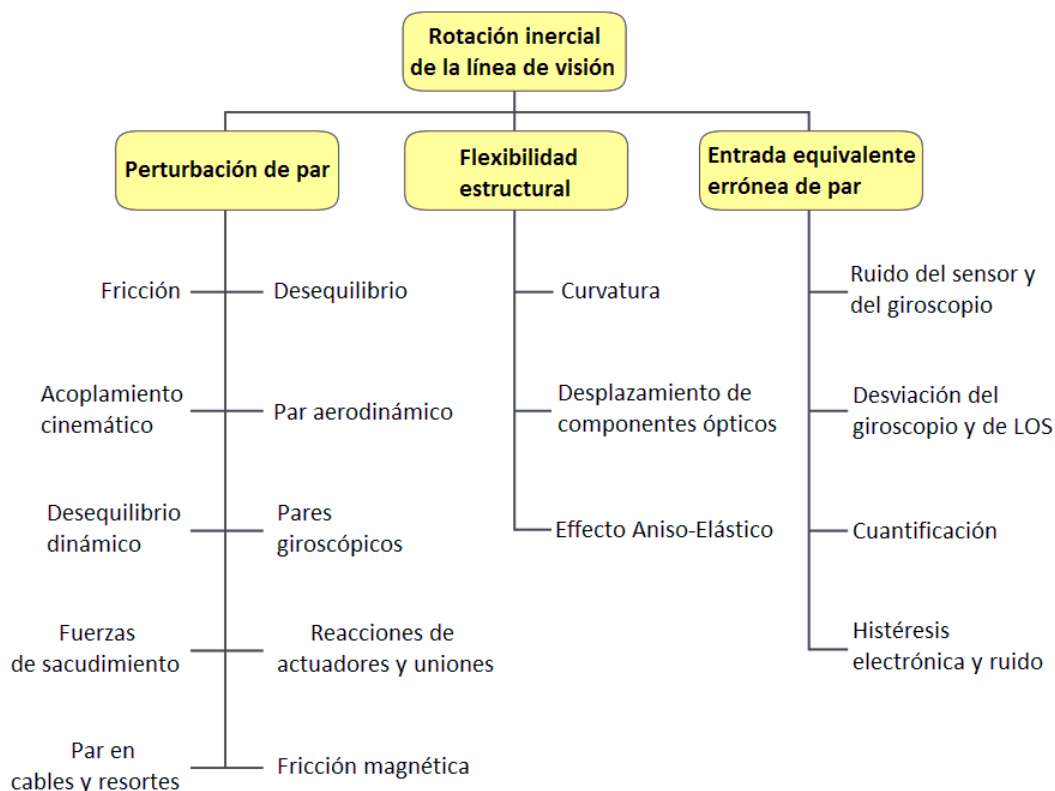


Figura 13: Causas de rotación inercial de la línea de visión

Fuente: (Hilkert, 2008)

2.2.1. Estructura

Generalmente las gimbals usadas en los UAV son de 2 o 3 ejes. Hay 12 posibles configuraciones de estructuras de 3 ejes, pero estas se reducen en 2, indicadas en la **Figura 14**, debido a que son una representación simétrica de estas. La ventaja principal que posee la primera configuración sobre la segunda es que el momento de inercia y el peso que se tiene en el eje Yaw se puede distribuir entre dos actuadores, la otra es que reduce problemas mecánicos como el efecto de bamboleo. (Mendez, Rodriguez, Brady, & Tosunoglu, 2010)

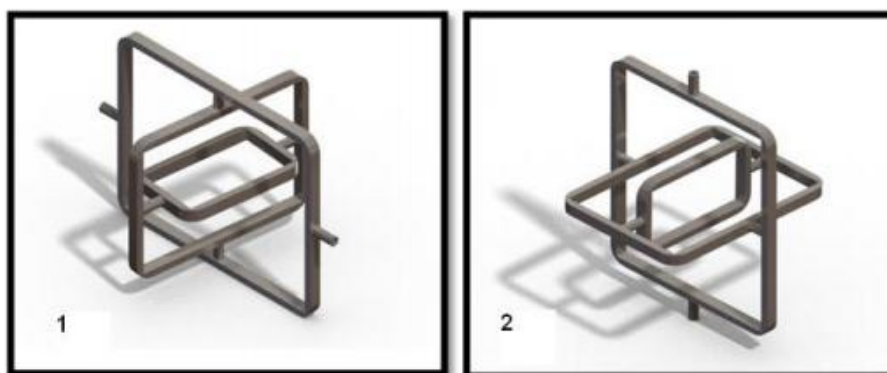


Figura 14: Configuraciones de plataformas de 3 grados de libertad

Fuente: (Mendez et al., 2010)

Una ISP de 3 ejes pierde un grado de libertad si los 3 ejes se encuentran en el mismo plano, ya que no hay manera de "desenrollar" las rotaciones perpendiculares a ese plano. Cuando esto sucede se le conoce como gimbal lock y el sistema de referencia se pierde. Esto sucede por lo general a plataformas que poseen una estructura mecánica y modelos de control simples, y además de estabilizar una carga pequeña. (Yu, Wu, Zong, & Kong, 2016)

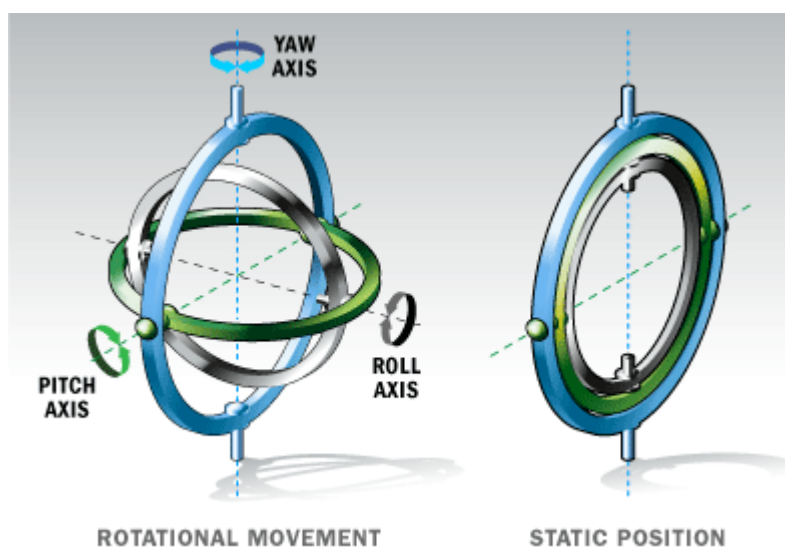


Figura 15: Movimientos de la gimbal y gimbal lock

Fuente: (Strickland, 2016)

Para evitar situaciones de gimbal lock que surgen inesperadamente se puede colocar topes físicos para limitar el movimiento del mismo o electrónicamente detener el movimiento la gimbal. En caso de poseer espacio disponible se puede añadir otro eje o a su vez utilizar algoritmos para aprovechar las capacidades de ordenadores a bordo para reconocer que se va a producir un gimbal lock y reorientarla.

2.2.2. Amortiguadores

Generalmente ya sea por el movimiento del dron o por fuerzas externas se generan vibraciones indeseadas, las cuales pueden generar fallas o daños en las estructuras. Se pueden emplear muchos procedimientos para tratar de controlar y mantener estas vibraciones dentro de ciertos límites, uno de los más utilizados es un control pasivo, el cual no requiere de una fuente de energía externa para su funcionamiento e involucra una reducción de las estas por medio de resortes, materiales elásticos y amortiguadores que se adicionan a la estructura desde a etapa de diseño. (Martínez & Portillo, 2013)

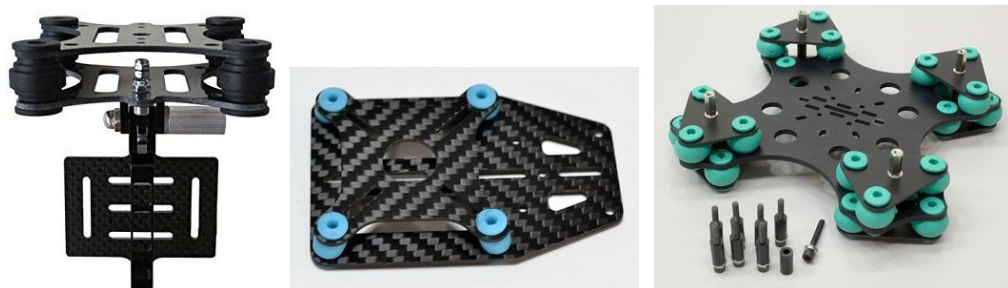


Figura 16: Ejemplo de amortiguadores para gimbal

La desventaja principal de este tipo de amortiguadores está en la limitación de eliminar sólo las vibraciones en el rango de frecuencias para el cual fue calculado, por lo que puede resultar ineficiente o inestable si el rango de operación del sistema cambia. Adicionalmente, los aisladores pasivos tienen

dos restricciones negativas. La primera de ellas es que el aislante seleccionado debe soportar la carga estática de la máquina. La segunda es que el soporte debe tener una rigidez lo suficientemente baja para que la frecuencia de resonancia de la máquina montada sobre los soportes aislantes sea considerablemente menor que la frecuencia de operación de la máquina. (Martínez & Portillo, 2013)

2.2.3. Sensores

Los sensores son uno de los componentes más críticos en una gimbal. Dos requerimientos importantes para una ISP son: ancho de banda y ruido eléctrico, ya que los sensores al encontrarse generalmente en la parte de retroalimentación del bucle de control su ancho de banda o las características de respuesta dinámica afectan directamente al ancho de banda del bucle de control global. La exactitud del factor de escala y el bias son consideraciones secundarias, pero su nivel de importancia puede variar dependiendo de la aplicación a realizar. (Hilkert, 2008)

Un dispositivo muy usado en las gimbals para drones es la unidad de medición inercial (IMU), el cual mide la posición y orientación. Este dispositivo posee diferentes sensores como giroscopios, magnetómetros y acelerómetros para combinar sus mediciones de forma inteligente y realizar estimaciones de orientación, aceleración lineal y velocidad angular en los tres ejes del espacio. (Pierre Vigouroux Cavolina, 2010)

Estos datos no se encuentran tan afectados por los factores ambientales como en el caso de los receptores GPS. No obstante, las IMU presentan errores a corto, medio y largo plazo temporal, debido a su carácter acumulativo. (Preckler Clemente, 2012)

La exactitud de los datos generados por la IMU depende de la precisión de los acelerómetros y los giroscopios, se debe tener en cuenta que no todos los

errores son relevantes para las IMU, ya que algunos son tan pequeños que no representan una gran diferencia en los datos obtenidos, es por eso, que algunos fabricantes no los describen en sus especificaciones. (Novatel, 2014)

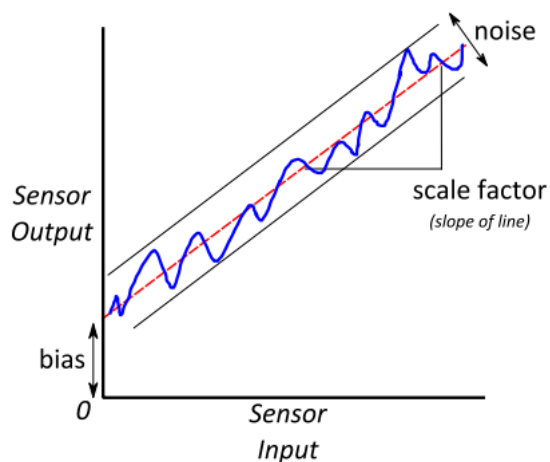


Figura 17: Errores comunes de las IMU

Fuente: (Novatel, 2014)

2.2.4. Motores

Las ISP requieren actuadores que proporcionen el par y la velocidad adecuados sin excesivo cogging o histéresis. El par del actuador debe ser suficiente para suprimir las distorsiones de par y para responder a las entradas de comando para mover a las ISP y a la línea de visión como se requiera en la aplicación. También el par debe ser capaz de responder al ruido de alta frecuencia sin saturar o sobrecargar el actuador. (Hilkert, 2008)

La mayor parte de las gimbals utilizadas en drones usan motores dc sin escobillas (brushless motors - BLDC), los cuales combinan la larga vida útil del motor de inducción, la linealidad del motor de imán permanente, además aumentan la capacidad de rango de velocidad (productividad), reducen el peso (diseño compacto) y mejoran la capacidad de torsión (precisión). (Motor Consultants, 2015)

Estos motores no incorporan un conmutador mecánico en su diseño, esto es lo que los hace diferente de la tecnología de los otros motores, ya que utiliza un conmutador electrónico, ya que es realizada por la parte de control, siendo esta la que provee la energía al motor. (Motor Consultants, 2015)

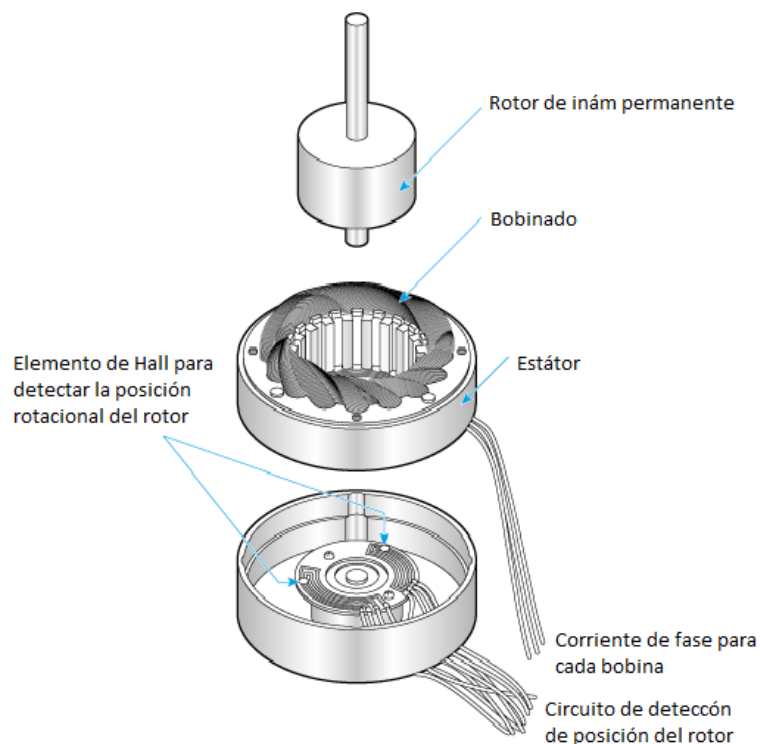


Figura 18: Estructura de un motor eléctrico sin escobillas

Fuente: (NIDEC Corporation, 2016)

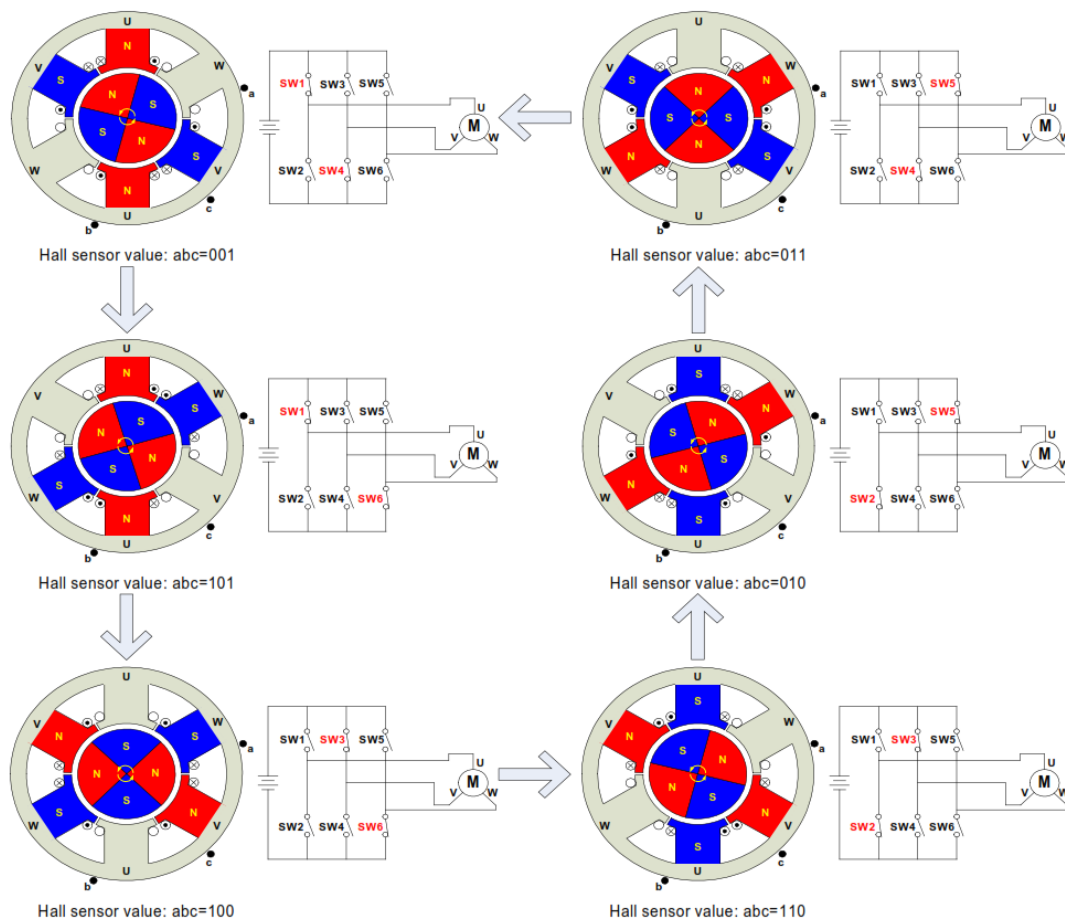


Figura 19: Secuencia de conmutación del motor sin escobillas

Fuente: (Zhao & Yangwei, 2011)

Cuadro 3

Comparación de motores BLDC con otros motores

<i>Característica</i>	<i>Motor sin escobillas</i>	<i>Motor con escobillas</i>	<i>Motor de inducción</i>
-----------------------	-----------------------------	-----------------------------	---------------------------

Continúa



Estructura mecánica	Campos magnéticos del rotor y estator están hechos de imanes permanentes	Campos magnéticos del rotor y estator están hechos de imanes permanentes o electroimanes	El rotor y el estator tienen devanados, pero las líneas AC están conectadas al estator
Mantenimiento	Bajo o nada	Periódico	Bajo
Características Velocidad-Torque	Funcionamiento a todas las velocidades con carga nominal	Perdidas de torque en altas velocidades debido a las pérdidas en las escobillas	No lineal
Eficiencia	Alta	Moderada	Baja
Método de conmutación	Utiliza interruptores de estado sólido	Contactos mecánicos entre escobillas y conmutador	Se requiere de circuito especial para encendido
Rango de velocidad	Alto	Moderado	Bajo
Ruido Eléctrico	Bajo	Alto	Bajo
Método de detección de la posición del rotor	Sensores Hall, codificadores ópticos, etc.	Detección automática por escobillas y conmutador	No tiene
Inversión de dirección	Invirtiendo la secuencia de conmutación	Invirtiendo el voltaje en la terminal	Cambiando las dos fases de entrada del motor
Requerimientos de control	Un controlador siempre es requerido para realizar la secuencia de	Para velocidades fijas no se requiere controlador, para velocidades variables si se	Para velocidades fijas no se requiere controlador, para velocidades variables si se

	conmutación	requiere	requiere
Costo del sistema	Alto, debido al controlar	Bajo	Bajo

Fuente: (Madaan, 2013)

2.2.5. Controlador de gimbal

Debido a que los motores sin escobillas necesitan su propio controlador para realizar la conmutación necesaria para su funcionamiento, lo más recomendable es la utilización de los controladores de gimbal, los cuales existen de diferentes modelos y marcas, así mismo, la cantidad de ejes que permiten controlar.

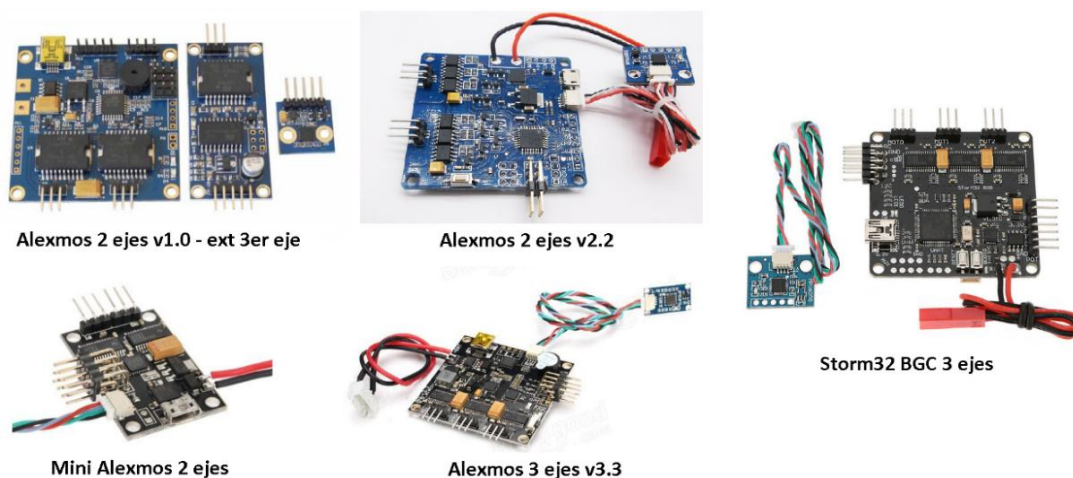


Figura 20: Ejemplo de controladores de gimbal

2.2.6. Dispositivo de captura de imágenes

Las cámaras deportivas, tales como la serie GoPro, están diseñados para la documentación de los eventos deportivos en primera persona, como el ciclismo y montañismo, entre otros. Debido a su pequeño tamaño son cada vez más populares en los robots móviles pequeños y vehículos de radio control, esto se debe a que poseen una buena relación rendimiento/tamaño. (Ovr & Forssén, 2015)

Como se mencionó anteriormente, los UAVs se utilizan para una gran variedad de tareas, siendo controlados de manera manual o remota, mientras recolectan información e imágenes de su entorno. Estas imágenes, generalmente son analizadas después de haber realizado el vuelo, debido a que se posee una potencia de procesamiento limitado. Para operar de manera más robusta y autónoma, los UAVs deberían estar equipados con más poder de procesamiento y así procesar las imágenes on-board, permitiendo al UAV analizar y reaccionar en tiempo real.

Una plataforma de hardware puede ser un procesador embebido (por ejemplo, un Raspberry Pi) o incluso una computadora pequeña como una laptop, dependiendo de la potencia de procesamiento que se necesite. El uso de estos en un UAV impone limitaciones en estas plataformas, ya que deben ser ligeros, pequeños y tener una potencia de procesamiento adecuada con bajo consumo de energía para mantener largos tiempos de vuelo (Hulens, Verbeke, & Goedemé, 2016)

2.3. Ordenadores on-board

Debido a los grandes avances tecnológicos se ha podido reducir las dimensiones de las computadoras, llegando así a las Mini PC (por ejemplo: las NUC de Intel), siendo estas un gran complemento para los UAVs. Sin embargo, poseen peso y consumo de energía elevados, por lo que su utilización es limitada.

Avances recientes relacionados con plataformas embebidas muestran un aumento continuo en la potencia de procesamiento con consumo de energía y peso razonables, por lo que son complementos ideales para las UAVs, a estos dispositivos se los conoce por el nombre de Single-Board Computer (SBC).

2.3.1. Single-Board Computer (SBC)

Son computadoras completas construidas en una sola placa, las ventajas principales de las SBC son sus dimensiones y peso reducidos, y costo. Hoy en día son ampliamente utilizadas en electrónica de consumo, control industrial, instrumentación médica, redes de comunicación, automóviles y otros dispositivos de mano. (Alee, Rahman, & Ahmad, 2011)

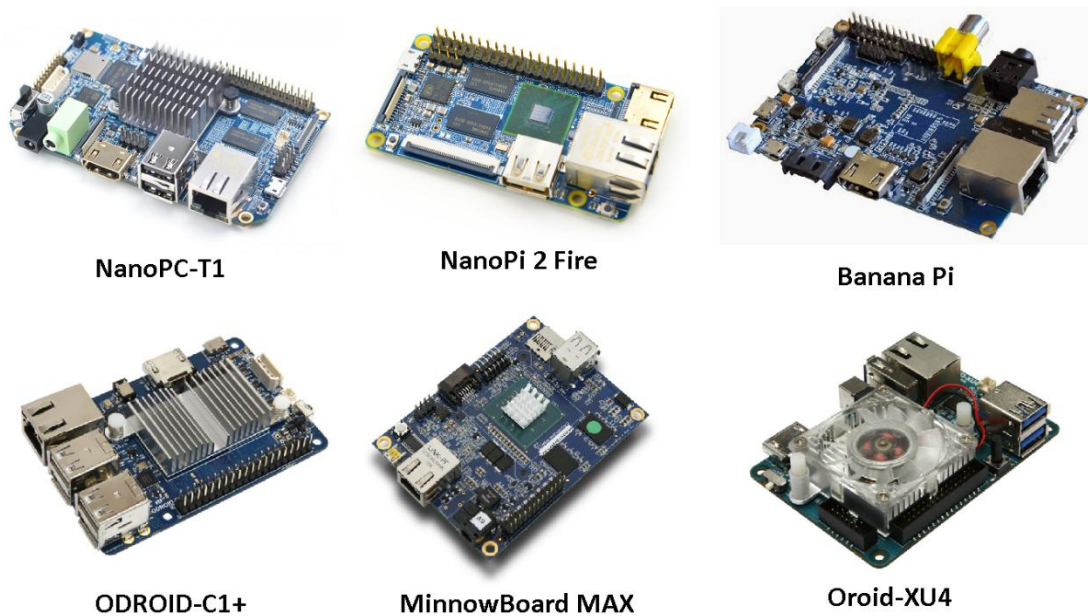


Figura 21: Ejemplo de Single-Board Computers

Estos dispositivos pueden contener uno o varios procesadores, aumentando así su capacidad de procesamiento y sus prestaciones.

2.3.2. Procesador multicore asimétrico

Un procesador multicore asimétrico (Asymmetric Multicore Processor - AMP) integra cores con distintas características en un mismo chip. Por un lado, los AMPs integran un grupo de cores rápidos, de alto rendimiento y consumo, que trabajan a alta frecuencia e implementan complejas técnicas micro arquitectónicas. Por otra parte, estos procesadores incluyen un grupo de cores

Figura 22: Modo de configuración del comando minicom.

Fuente: (Negus & Caen, 2008)

2.4.2. UART

UART (Transmisor Recibidor Asíncrono Universal), es un tipo de protocolo de comunicación serial, usado comúnmente para intercambio de información entre una computadora y periféricos a corta distancia, baja velocidad y bajo costo. Incluye tres módulos kernel los cuales son: generador de velocidad de baudios, receptor y transmisor. (Fang & Chen, 2011)

2.4.3. WiringPi

Es una librería de control extenso de GPIO (Entradas y Salidas de propósito general) para la plataforma Raspberry Pi que ha sido escrita y mantenida por Gordon Henderson. Usa el “sysfs” y técnicas de mapeo de memoria para crear una librería y colección de comandos altamente eficientes que han sido desarrolladas a la medida de la plataforma RPi. (Molloy, 2016)

2.4.4. ROS

Es un código abierto para el sistema operativo de robots. El cual provee una capa de comunicaciones estructuradas sobre el host de sistemas operativos de un clúster computacional heterogéneo. Permite la comunicación de puerto a puerto, está basado en herramientas que cumplen varias tareas y abarca varios lenguajes de programación. Es un código ligero, gratuito y abierto. (Quigley et al., 2009)

2.4.5. SSH

SSH (secure shell), es un protocolo de inicio de sesión remota y segura, el cual utiliza canales multiplexados en un solo túnel encriptado. Este protocolo ha sido diseñado para correr encima de la capa de transporte SSH y protocolos de identificación de usuario. (Lonvick, 2006)

Para poder realizar la estabilización de video por software, se requirió de códigos de programación empleados en el campo profesional para obtener el mejor video resultante.

2.5. Estabilización de vídeo

Las técnicas de estabilización de vídeo han sido estudiadas durante décadas, y muchos esquemas se han propuesto para solucionar los problemas implícitos. Estos métodos pueden clasificarse en dos categorías: Estabilización de vídeo mecánica y Estabilización de video por procesamiento de imágenes.

La estabilización de vídeo mecánica emplea un grupo de sensores inerciales incorporados en un dispositivo conocido como IMU (Unidad Medición Inercial) para detectar el desplazamiento de roto-traslación de la cámara para compensar los movimientos y mantener estable la cámara. Aunque este método es capaz de obtener buenos resultado en algunas aplicaciones, no puede eliminar todas las vibraciones del vídeo, además, su capacidad puede ser limitada por su complejo hardware, peso, costo y el requisito de módulos especiales.

La estabilización de video por procesamiento de imágenes se logra a través de técnicas de procesamiento de imágenes para estimar el movimiento de la cámara calculando el grado de los parámetros de transformación geométrica entre cuadros consecutivos de vídeo, suavizando los parámetros y compensando la desviación de las imágenes. Comparando estos dos métodos, el procesamiento de imágenes en la estabilización de vídeo es una técnica de estabilización de vídeo ideal, más preciso, flexible, barata y fácil de operar. (Yue Wang, Hou, Leman, & Chang, 2011). Algunos de los problemas que se presentan en distintos métodos de estabilización de vídeo son:

- Presencia de objetos cercanos

Es uno de los principales problemas, ya que la mayor parte de los puntos de interés detectados son en la región de este objeto.

- Objetos en movimiento

Algunos objetos con muchos puntos de interés durante la estimación de movimiento hacen que se tenga un seguimiento no deseado de estos objetos.

- Cámaras en drones

Escenas fotograma³ a fotograma, desplazamientos significativos, bajas frecuencias de video, congelamientos, desplazamientos a alta velocidad son problemas frecuentes debido a la complejidad dinámica de los drones durante un vuelo. En todos estos casos, los cambios entre dos fotogramas consecutivos pueden ser considerables, produciendo un problema crítico para la estabilización de video.

Para solucionar estos problemas, a cambio de un coste computacional mayor, se utiliza un proceso de emparejamiento de puntos de interés en base al algoritmo RANSAC y la diferencia de nivel de gris entre fotogramas consecutivos. En consecuencia, el proceso de estimación de movimiento no se realiza en los puntos de interés de los objetos sino de todo el fotograma.

Varios métodos de suavizado de movimiento están disponibles para la estimación de la intención de movimiento tales como filtro de partículas, filtro de Kalman, filtro adaptativo, entre otros. Una combinación de un filtro paso bajo de segundo orden, usando el menor número de fotogramas como sea posible, y la

³ Fotograma: Se conoce como fotograma a cada una de las imágenes que suceden en una película

entrada de acción de control son empleados para estimar una intención movimiento fiable. (Aguilar & Angulo, 2014)

2.5.1. Filtro de Kalman

El filtro de Kalman consiste en un conjunto de ecuaciones matemáticas que proveen una solución recursiva óptima, por el método de mínimos cuadrados. La meta de esta solución consiste en calcular un estimador lineal, y óptimo del estado⁴ de un sistema en un tiempo t con base en la información disponible en $t-1$, y actualizar, con la información adicional disponible en t , dichas estimaciones.

El filtro se desempeña suponiendo que el sistema puede ser descrito a través de un modelo estocástico lineal, en donde el error asociado tanto al sistema como a la información adicional que se incorpora en el mismo tiene una distribución normal con media cero y varianza determinada. La solución es óptima por cuanto el filtro combina toda la información observada y el conocimiento previo acerca del comportamiento del sistema para producir una estimación del estado de tal manera que el error es minimizado estadísticamente.

Lo que hace al filtro tan interesante es precisamente su habilidad para predecir el estado de un sistema, aun cuando la naturaleza precisa del sistema modelado es desconocida. En la práctica, las variables estado individuales de un sistema dinámico no pueden ser determinadas, de forma exacta, por una medición directa. Dado lo anterior, su medición se realiza por medio de procesos estocásticos que involucran algún grado de incertidumbre en la medición. (Solera Ramírez, 2003)

⁴ El estado debe contener la información más relevante del sistema en cada momento del tiempo, tratando de considerar el menor número de variables posible.

2.5.2. OpenCV

Es una librería de visión computacional de código abierto escrito en C y C++ que corre bajo Linux, Windows y Mac OS X. Además, existe un desarrollo activo en interfaces para Python, Ruby, Matlab y otros lenguajes.

Provee una infraestructura de visión computacional de simple uso, contiene más de 500 funciones que abarcan varias áreas de visión, las cuales incluyen inspección de productos industriales, imágenes médicas, seguridad, interfaz de usuario, calibración de cámaras, visión estéreo y robótica. También contiene una librería de propósito general para el aprendizaje de máquina (MLL) que trabaja en conjunto con la visión computacional, la cual está enfocada en reconocimiento de patrones estadísticos y agrupamientos.(Bradski & Kaehler, 2008)

2.5.3. RANSAC

RANSAC (Muestras Aleatorias de Consenso) interpreta/suaviza información que contenga un importante porcentaje de error. Ideal para aplicaciones de análisis de imagen automatizado, donde la interpretación se basa en información proveída por un detector de características propenso al error. (Choi, Kim, & Yu, 2009)

2.5.4. SURF

SURF (Aceleradas Características Robustas), es un detector y descriptor de puntos de interés invariante en escala y rotación. Se basa en la medición basada en la matriz Hessiana para el detector, y en un descriptor basado en distribución, el cual a su vez simplifica estos métodos. (Bay et al., 2006)

2.5.5. FAST

FAST (Puntos de interés de prueba de segmento acelerados), es un detector de puntos de interés donde el centro del área circular en una imagen,

es usada para determinar claridad u oscuridad en pixeles cercanos. Sin embargo no toda el área del círculo es evaluada, sino solo los pixeles discretizados en el círculo describiendo el segmento.(Mair, Hager, Burschka, Suppa, & Hirzinger, 2010)

2.5.6. BRIEF

BRIEF (Características Elementales Independientes Binarias Robustas), es un descriptor de puntos de interés eficiente que usa líneas binarias. Es altamente discriminatoria incluso usando pocos bits y puede ser procesada por computador usando simples pruebas de diferencia de intensidad. Resultando en ser rápido en construirse y en emparejar. (Calonder, Lepetit, Strecha, & Fua, 2010)

2.5.7. ORB

ORB (FAST orientado y BRIEF rotado), es un descriptor binario de puntos de interés, basado en BRIEF, el cual es resistente al ruido e invariante a la rotación. Es dos veces más rápido que SIFT en desempeño, así como también en varias situaciones. Su eficiencia es probada en varias aplicaciones del mundo real, incluyendo detección de objetos, y rastreo de parches en smartphones. (Rublee, Rabaud, Konolige, & Bradski, 2011)

2.5.8. RobustMatcher

Es un algoritmo del estado de arte que realiza emparejamiento entre un par de imágenes, los cuales observan una misma imagen en diferentes puntos de vista. Este emparejamiento es realizado de manera bidireccional, si la distancia medida entre descriptores varía significativamente, solo el primer emparejamiento es aceptado, y el segundo es rechazado, el resultado son dos conjuntos de emparejamientos, y solo los emparejamientos que están en ambos conjuntos son aceptados, el resto son rechazados. El conjunto resultante es

comprobado usando RANSAC, llegando a ser un algoritmo robusto que entrega resultados de alta calidad. (Wasenm, 2015)

2.6. Resumen

- Los RUAS se pueden clasificar en 5 categorías, de acuerdo a su carga útil y peso total. Siendo los Mini RUAS (4ta Categoría) excelentes bancos de pruebas para investigación, debido a su bajo costo, portabilidad, carga útil, ya que puede llegar cerca de los 2kg con un tiempo de vuelo de 5 minutos a 1 hora.
- Indistintamente del tipo de UAV que utilice, estos poseen un sistema que realiza el control de movimiento, conocido como Autopilot, el cual lo mantiene estable mientras vuela y puede llegar a realizar vuelos autónomos. Siendo este calibrado y configurado por una estación de control, que puede estar instalado en una computadora o dispositivo móvil.
- Debido al creciente interés del uso de cámaras a bordo de drones, existe gran variedad de gimbals, que permiten mantener una línea de visión estable. Estas están compuestas por una estructura, amortiguadores, motores y el sensor o grupo de sensores.
- Generalmente los UAV utilizan una gimbal de 2 o 3 ejes. Las gimbals de 3 ejes sufren de “Gimbal Lock”, que es la pérdida de un grado de libertad y el sistema de referencia, esto ocurre cuando los ejes se encuentran en el mismo plano. Esto sucede a plataformas que poseen una estructura mecánica y modelos de control simples, y además de estabilizar una carga pequeña.
- Algunos tipos de Autopilots permiten complementar su sistema mediante ordenadores on-board, para poder operar de manera más robusta y realizar vuelos autónomos procesando información de su entorno y así reaccionar en tiempo real.

- Debido a los avances tecnológicos relacionados con las plataformas embebidas, existe un aumento de potencia de procesamiento con consumo de energía y peso razonables, lo que permite que estos dispositivos sean complementos ideales para las UAVs, a estos dispositivos se los conoce como Single-Board Computer.
- Las técnicas de estabilización de vídeo pueden clasificarse en dos categorías: Estabilización de vídeo mecánica, la cual está limitada por su hardware, peso, costo y módulos especiales, y la Estabilización de vídeo por procesamiento de imágenes, la cual a través de procesamiento de imágenes estima la intensidad de movimiento de la imagen.
- Existen varios protocolos de comunicación empleados para la comunicación de los ordenadores on-board con los controladores de vuelo y las gimbal, pero dependiendo de los sistemas y dispositivos que se tenga, se deben emplear los más adecuados y compatibles.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN

El capítulo se centra en el diseño y desarrollo de un sistema de estabilización para cámaras a bordo del dron mini RUAV “STORM Drone 6 V3”. Inicialmente se selecciona el Autopilot y la estación de control para su calibración y configuración, después de ello, los elementos que conforman la gimbal y ordenador on-board. Finalmente se diseña y construye la base donde se colocarán dichos elementos y se implementa el algoritmo de estabilización de vídeo.

3.1. Sistema Aéreo no tripulado

En esta subsección se explican las razones para la selección del Autopilot y la estación de control, para su posterior calibración, configuración y ensamble del dron.

3.1.1. Selección del Autopilot

El modelo de la planta del dron se basa en información obtenida por el Autopilot. Se utilizó un autopiloto reconfigurable que permite la extracción de datos inerciales.

Varios de los controladores de vuelo más robustos que permiten vuelos autónomos y trabajan con software libre son: Pixhawk, Erle-Brain, ArduPilot Mega, Arsov AUAV-X2. Estos autopilotos poseen soporte de ArduPilot. A continuación, se presenta especificaciones técnicas importantes:

Cuadro 4
Especificaciones de controladores de vuelo

	<i>Pixhawk</i>	<i>Erle-Brain 2</i>	<i>ArduPilot Mega</i>	<i>Arsov AUAV-X2</i>
CPU	-32-bit STM32F427 Cortex M4 core con FPU	-900MHz quad- core ARM	-AtMega 2560	-STM32F427VI ARM
	-32 bit STM32F103 coprocesador a prueba de fallos	Cortex-A7 CPU		-STM32F100C8 T6 ARM
Sensores	-Giroscopio 3- ejes 16-bit -Acelerómetro/ Magnetómetro 3-axis 14-bit -Acelerómetro/ Giroscopio 3- ejes -Barómetro	-Sensor de gravedad -Giroscopio -Magnetómetro digital -Sensor de Temperatura y presión -ADC para medición de batería	-Giroscopio 3- ejes -Acelerómetro -Barómetro de alto rendimiento	-Giroscopio digital 3-ejes -Acelerómetro/ Magnetómetro compacto 6-ejes -Giroscopio/ Acelerómetro 6- ejes -Barómetro

Continúa



Interfaces	-5x UART	-12x PWM		
	-2x CAN	-1x Entrada RC		-3x UART
	-Spektrum DSM/ DSM2/ DSM-X	-1x Módulo de poder	-Entrada GPS	-1x CAN
	-Futaba S.BUS	-1x I2C connector	-I2C	-1x I2C
	-Codificador PPM	-1x I2C connector	-Módulo de poder	-1x SPI
	-Entradas RSSI	-1x UART	-Conector Micro USB	-2x ADC
	-I2C	-4 Puertos USB	-Entrada 3.3v	-8x PWM
	-SPI	-1 Puerto HDMI	ADC	-2x JTAG
	-Entradas 3.3 y 6.6V ADC	-10/100 Internet -3.5mm conector de audio	-Entradas PWM	-Entrada tarjeta micro SD
	-Puerto micro USB		-Entrada RC	-Conector micro USB
		Con cámara		
		A: 70 mm		
		L: 96 mm		
	Dimensiones	A: 50 mm	E: 20 mm	A: 70.5 mm
L: 81.5 mm		Sin cámara	L: 45 mm	L: 50 mm
E: 15.5 mm		A: 70 mm	E: 13.5 mm	
		L: 96 mm		
	E: 58.3 mm			
Peso	38 g	100 g	31 g	10.9g

Fuente: (ArduPilot, 2016), (Acutronic Robotics Company, 2017), (PX4 Dev Team, 2017)

Como se puede observar, en el **Cuadro 4**, el Autopilot Pixhawk posee mejores características sobre los otros controladores, y por recomendación del Prof. Wilbert G. Aguilar Ph.D. se acordó en la utilización del este para el proyecto.



Figura 23: Pixhawk PX4

Fuente: (PX4 Dev Team, 2017)

3.1.2. Selección de la estación de control

A continuación, se presenta una tabla con base en las características que debe poseer la estación de control.

Cuadro 5

Características de las estaciones de control

	<i>Mission Planner</i>	<i>APM Planner 2.0</i>	<i>MAVProxy</i>	<i>QGroundControl</i>	<i>UgCS</i>
Facilidad de uso	Si	Si	No	Si	Si
De código abierto	Si	Si	Si	Si	No
Compatibilidad con Pixhawk	Si	Si	Si	Si	Si
Compatible con Linux	No	Si	Si	Si	No
Compatible con Windows	Si	Si	Si	Si	Si
Gratuita	Si	Si	Si	Si	Si
Permitir comunicación por MAVlink	No	Si	Si	Si	No

Poseer una configuración avanzada	Si	No	Si	Si	No
Planificación y ejecución de misiones	Si	Si	Si	Si	Si

La aplicación UgCS en su versión gratuita al no permitir un control y configuración más avanzada al Autopilot, y al no ser de código abierto no sirve para la realización del proyecto. Las prestaciones que ofrece esta aplicación sirve para otros propósitos ya sea de pasatiempo o negocio, más no para investigación y desarrollo.

La facilidad de uso, y la variedad de configuraciones que se puede realizar al Autopilot, son razones por las cuales no se eligió MAVProxy y APM Planner 2.0 respectivamente.

Para tener una mejor apreciación de cuál es la aplicación que mejor se adapta a los requerimientos del proyecto se acordó realizar pruebas tanto de vuelo, de configuración y comunicación del Autopilot utilizando las estaciones de control Mission Planner y QGroundControl. Teniendo como resultado que QGroundControl permite realizar más configuraciones, y permite una comunicación directa con MAVLink, por lo que se eligió este programa.

Los enlaces de descarga y los pasos para la instalación de la aplicación QGroundControl, se los puede encontrar en: https://donlakeflyer.gitbooks.io/qgroundcontrol-user-guide/content/download_and_install.html

3.1.3. Configuración del Autopilot

A continuación, se presenta el procedimiento realizado para la instalación del Firmware, calibración y configuración del Autopilot Pixhawk.

3.1.3.1. Instalación 1er Firmware

Se descargó e instaló el programa QGroundControl v3.0.0 (dagar, 2016), y la versión de firmware que se utilizó en la instalación y calibración inicial fue el “nuttx-px4fmu-v2-default” del 25 de Agosto del 2016, debido a que esta versión calibra los ESC correctamente. (LorenzMeier, GitHub, 2016)

Se conectó el Pixhawk a la computadora mediante un cable micro-USB a USB.



Figura 24: Conexión entre el Pixhawk y computadora

En la pestaña de configuración, representada por el símbolo con engranes, se selecciona la opción “Firmware”, se sigue los pasos indicados por el programa, después se selecciona la opción de “configuraciones avanzadas” y

se elige “Firmware personalizado”, en la cual se escoge el archivo previamente descargado.

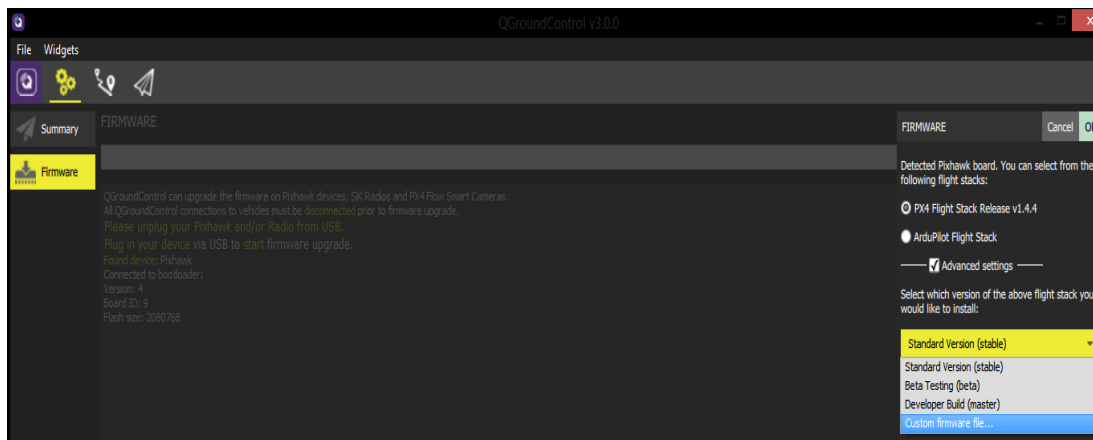


Figura 25: Instalación del Firmware de Pixhawk PX4

Después, se procede a calibrar y configurar el Pixhawk

3.1.3.2. Calibración y configuración

Lo primero que se realizó fue la se seleccionó el tipo de vehículo a usar, el cual es el Hexacoptero tipo X.

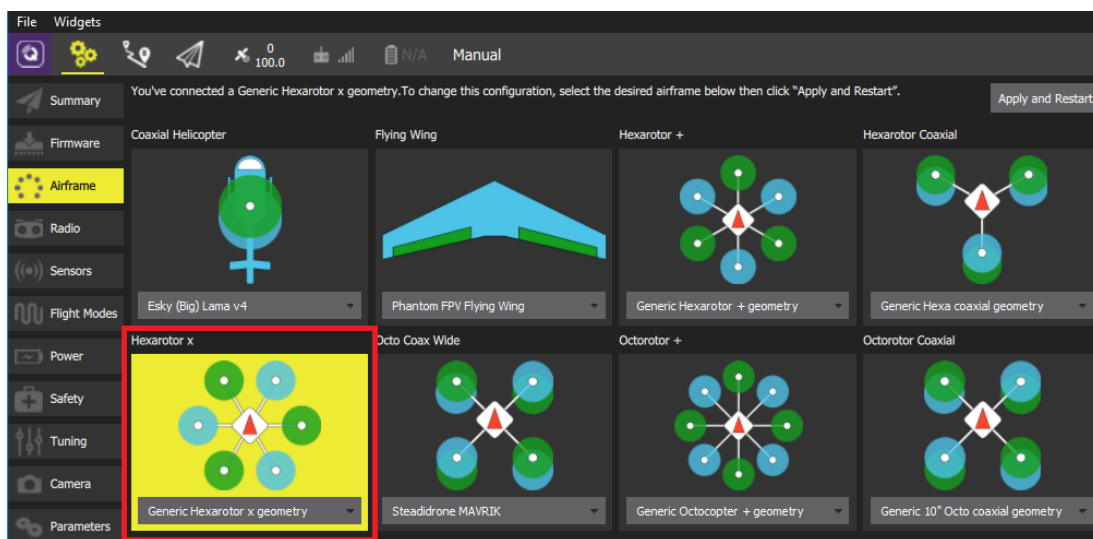


Figura 26: Tipo de vehículo

La calibración del radio control se utilizó para configurar los canales del mismo, y para determinar el comportamiento de cada palanca de mando.

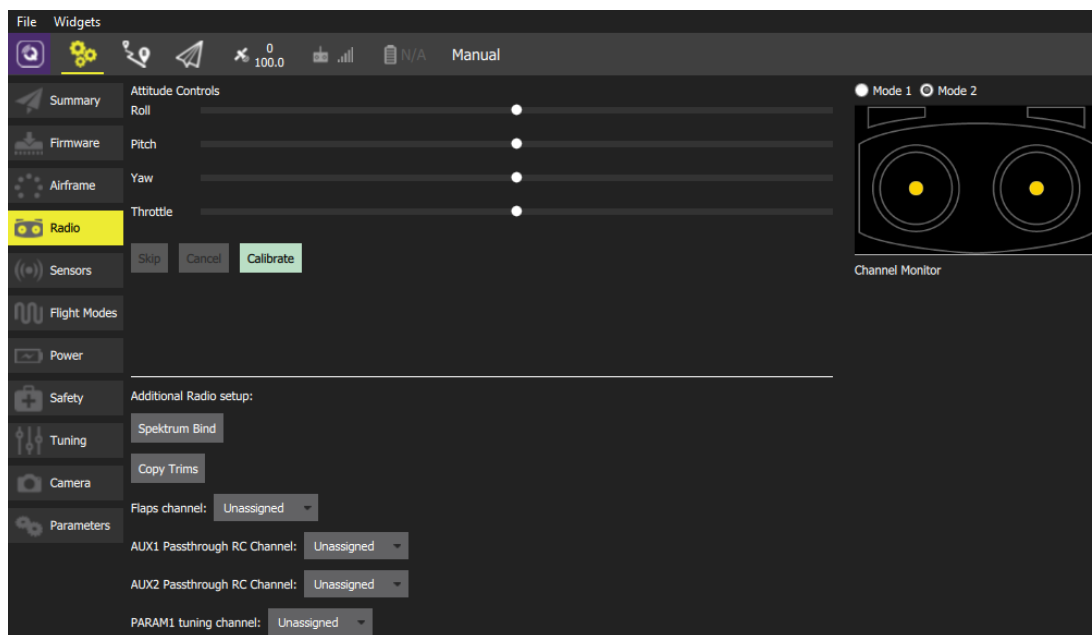


Figura 27: Calibración del radio control

Finalizada la calibración, se asignó Roll, Pitch, Yaw y Throttle, y se configuró automáticamente los máximos, mínimos, reversa y trim de los mismos para todos los canales.

Se calibró cada uno de los sensores, realizando las instrucciones indicadas en el programa conectado el GPS al controlador.



Figura 28: Calibración de sensores

Se comprobó que la calibración de cada sensor estaba correcta mediante los indicadores de colores que posee el programa.

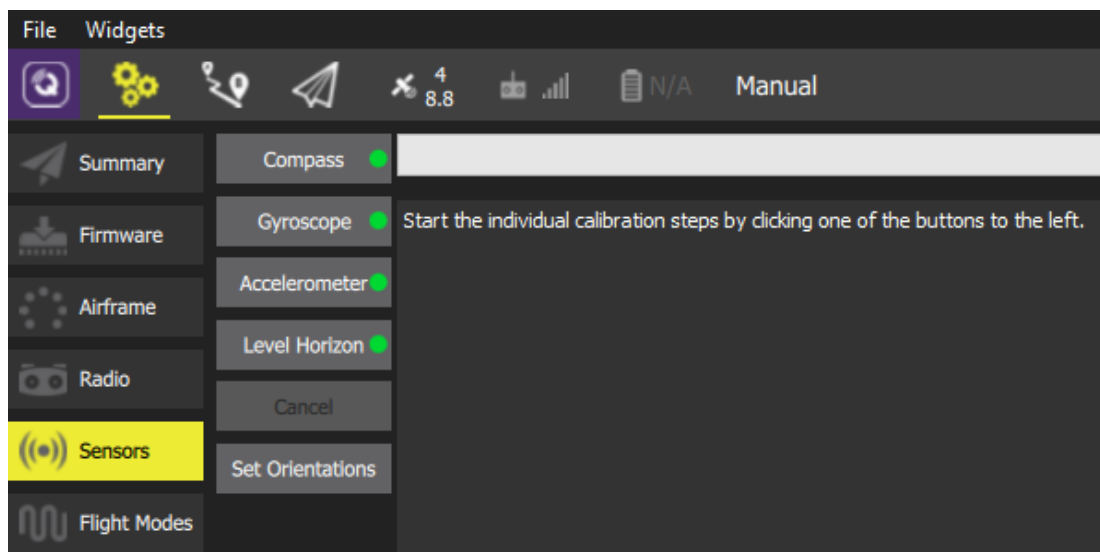


Figura 29: Indicador de calibración

Se asignó el modo de vuelo estabilizado, ya que este es un modo de vuelo recomendado para personas no experimentadas en volar drones, debido a que permite volarlo de manera manual pero este se auto-nivela en los ejes de Roll y Pitch.

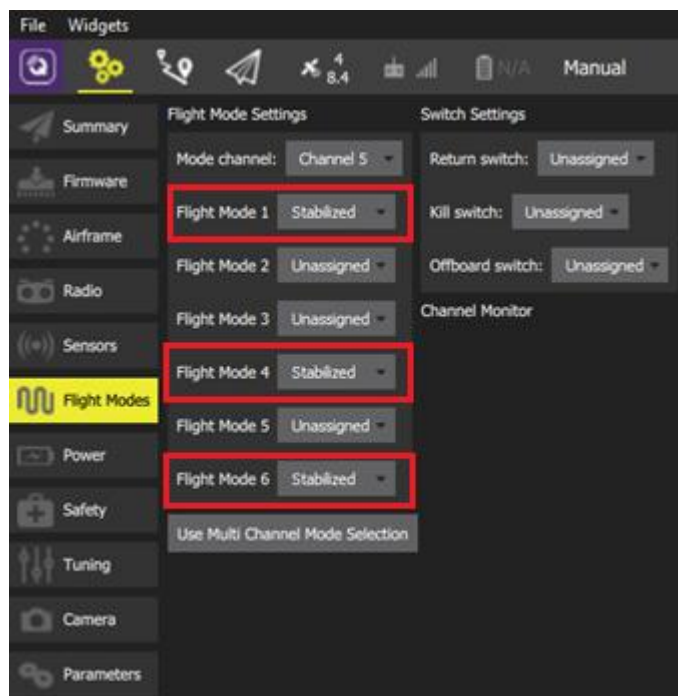


Figura 30: Modos de vuelo

Las condiciones que se deben cumplir para poseer un buen control del dron en este modo de vuelo son:

- El piloto tendrá que introducir con regularidad los mandos de Roll y Pitch para mantener el vehículo en su sitio cuando el viento lo empuje.
- La entrada de Yaw del piloto controla la tasa de cambio del mismo, ya que cuando este es soltado el vehículo mantendrá su rumbo actual.
- La entrada de altitud del piloto controla la velocidad media del motor, lo cual requiere un ajuste constante del mismo para mantener la altitud. Si se

coloca el acelerador completamente abajo los motores irán a su tasa mínima y si el vehículo está volando perderá el control y caerá.

Finalizada la calibración, se desconecta el Pixhawk, se cierra el programa QGroundControl, para realizar la instalación del nuevo firmware.

3.1.3.3. Instalación del 2do Firmware

Se descargó el firmware “Release v1.3.4” perteneciente al 03 de Julio del 2016. Y se realizó el mismo procedimiento instalación indicado previamente. Después se procedió a calibrar únicamente el giroscopio.

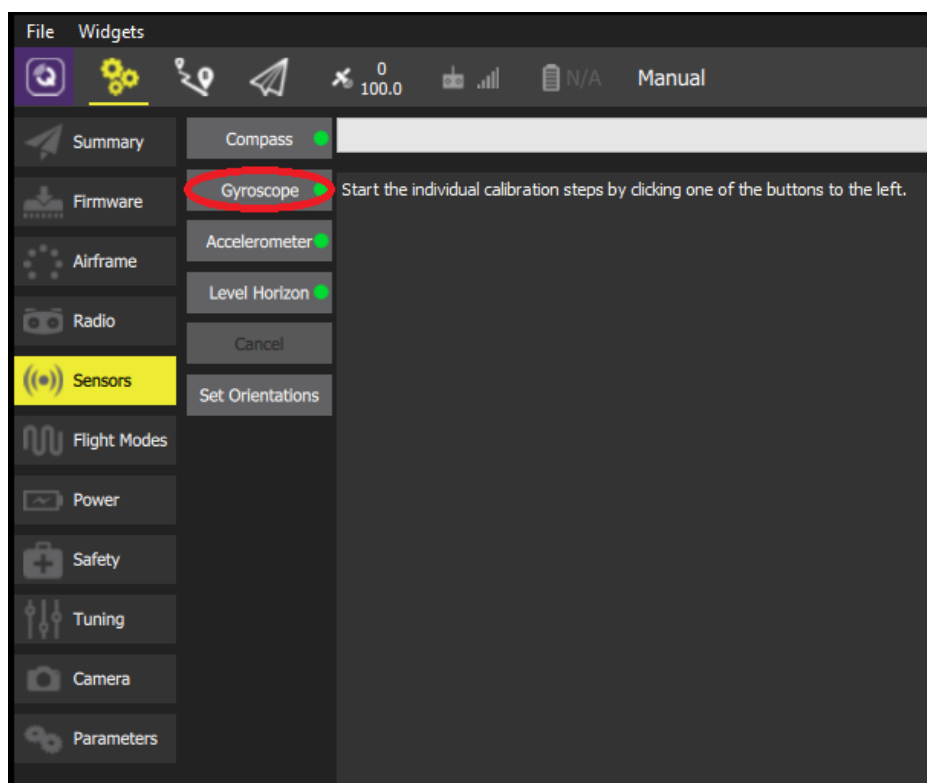


Figura 31: Calibración Final del Pixhawk

Se desconectó el Pixhawk y se extrajo la memoria micro-SD del mismo.



Figura 32: Extracción del Micro-SD del Pixhawk

Se acopló a un adaptador SD, y se conectó a la computadora para formatearla y se volvió a colocar en el Pixhawk.



Figura 33: Acople de la tarjeta micro-SD al computador.

Realizadas las configuraciones y calibraciones previamente indicadas se obtiene un correcto funcionamiento del Pixhawk.

3.1.4. Diagrama de conexión de los periféricos del Pixhawk

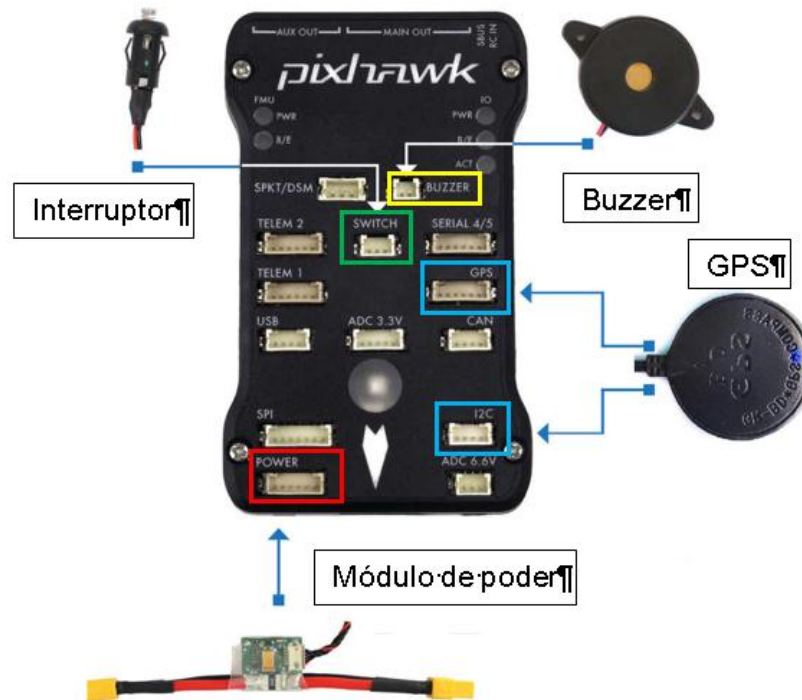


Figura 34 Conexión de los periféricos al Pixhawk

3.1.5. Conexión de los motores al Autopilot

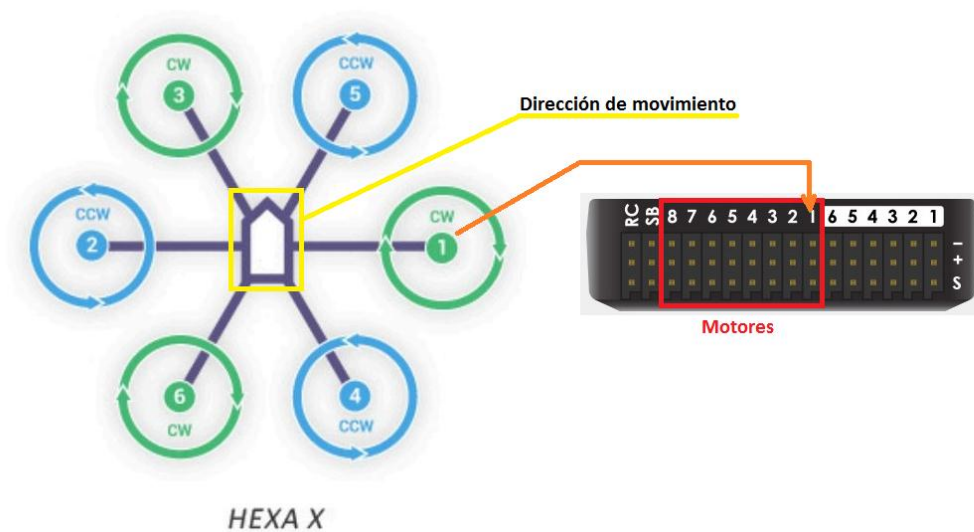


Figura 35 Conexión de los motores al Autopilot

3.1.6. Conexión del receptor de radio control al Autopilot

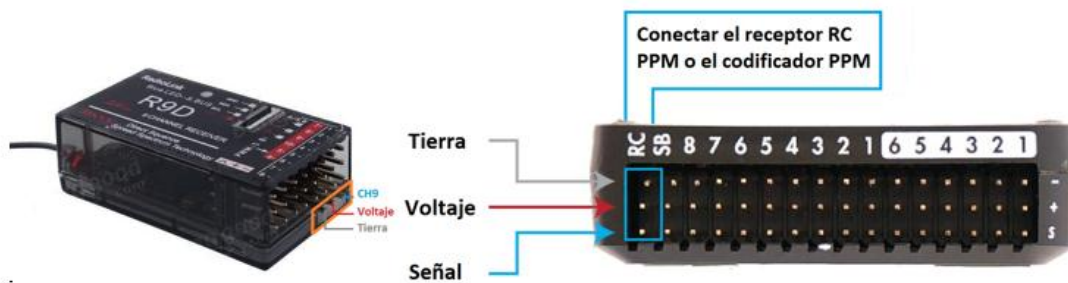


Figura 36 Conexión del receptor de radio control al Autopilot

El dron ensamblado queda como se indica en la **Figura 37**: Ensamble del dron.



Figura 37: Ensamble del dron

3.2. Componentes del sistema de estabilización

El sistema de estabilización está compuesto por: la plataforma estabilizada inercialmente (estabilización mecánica), el dispositivo de captura de imágenes, el ordenador on-board (estabilización por procesamiento de imágenes) y la base donde se montarán todos estos dispositivos.

3.2.1. Plataforma estabilizada inercialmente (ISP)

Como se mencionó en el capítulo anterior una ISP está compuesta por: estructura, sensores, motores, controlador y amortiguadores, los cuales se dan a conocer a continuación.

3.2.1.1. Estructura de la gimbal

Debido a la gran acogida que han tenido los drones y a la utilización de las cámaras en ellos, existe una amplia variedad de estructuras para gimbal tanto en dimensiones, como en peso y carga máxima a soportar. Para este proyecto se usó una estructura liviana y de dimensiones pequeñas, ya que se cuenta con un límite de carga máxima del dron de 1.3kg.

Entre las estructuras más utilizadas se encuentran aquellas fabricadas con aleación de aluminio, ya que este material es fácil de mecanizar, es liviano y de acuerdo a la aleación, posee diferentes propiedades y características térmicas y mecánicas.

La estructura que se seleccionó, se muestra en la **Figura 38**, la cual fue seleccionada teniendo en cuenta la disponibilidad en el mercado, que sea de bajo costo y que posea los requisitos mencionados anteriormente. Esta estructura posee un peso total de 27.6g.



Figura 38: Estructura de la gimbal

3.2.1.2. Selección de motores

Los motores montados en el sistema de estabilización son los “DYS BGM4108-130T-8.5 Gimbal Brushless Outrunner Motor for RC Aircraft-Black” y “Brushless Motor 2804-100T”.

Siendo los motores BGM4108 los que se escogió originalmente para el sistema, ya que se tenía como requerimiento que puedan soportar una amplia variedad de cámaras fotográficas para un futuro mejoramiento del sistema, sin embargo, debido a fallas humanas quedo solo un motor disponible, el cual se designó para ser usado en el eje Yaw, ya que el control sobre el mismo es de mayor importancia para lograr una mejor minimización de movimientos fantasma.

Para la selección de los motores restantes se acordó buscar aquellos que sean de bajo costo, que estén disponibles en el mercado y que soporten el peso de la estructura y la cámara, por lo que se escogió los 2804-100T, los cuales son muy utilizados por las gimbals debido a su bajo costo y a su buen desempeño con estructuras pequeñas y livianas, ya que soportan cámaras deportivas de estilo GoPro. A continuación, se presentan las especificaciones de los motores BGM4108-130T-8.5 y 2804-100T en los Cuadros 6 y 7 respectivamente.

Cuadro 6

Especificaciones técnicas motor DYS BGM4108-130T-8.5

Vueltas	130 T
Polos y Ranuras	24N22P
Diámetro de Cable	0.15mm
Diámetro del Eje	Hueco 8.5mm
Peso	93g
Máximo Peso de Cámara	700g
Tamaño de Motor (mm)	46 x 46 x 24

Ri (ohm)	15.5 ohm
----------	----------

Fuente: [DYS, 2016]

Cuadro 7

Especificaciones técnicas motor Brushless Motor 2804-100T

Vueltas	100 T
Polos y Ranuras	12N14P
Diámetro de Cable	0.19mm
Diámetro del Eje	Hueco 4mm
Peso	41.5g
Máximo Peso de Cámara	340g
Tamaño de Motor (mm)	34.5 x 16 x 24
Ri (ohm)	10 ohm

Fuente: Caja del motor

3.2.1.3. Placa base del motor

La estructura de la gimbal esta soportada por una base total en la cual se colocan todos los dispositivos del sistema, para ello se necesita una base que sirva de unión entre el motor del eje Yaw y la base total.

La consideración que se tomó en cuenta para la elaboración del diseño CAD, es que este debe poseer las dimensiones de los orificios del motor del eje Yaw exactas, para poder obtener una buena unión entre la base y el motor. De la misma manera, que las distancias de los orificios donde se colocarán los amortiguadores sean asimétricas, ya que de esta manera se tendrá una carga distributiva igual para todos ellos. Teniendo así una base como se muestra en la **Figura 39**.

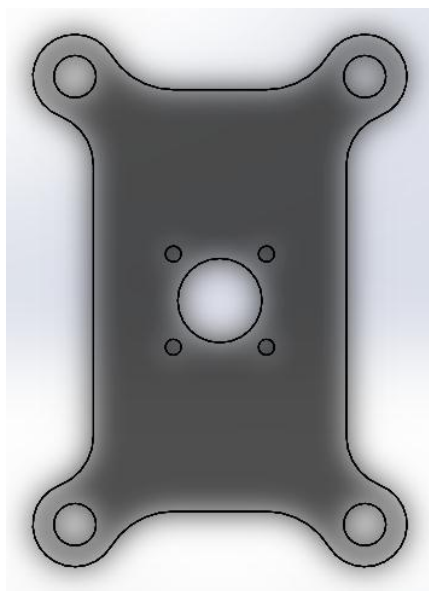


Figura 39: Placa base del motor del eje Yaw

El material seleccionado para la elaboración de la placa es de acrílico debido a su bajo costo, sus propiedades y características.

Cuadro 8

Propiedades del acrílico

	<i>Unidades</i>	<i>Norma ASTM</i>	<i>Colado</i>	<i>Extruido</i>
Peso específico	g/cm ³	D-792	1.19-1.20	1.19-1.20
Resistencia a la rotura	kg/cm ²	D-638	562-773	492-773
Resistencia en el límite elástico	kg/cm ²	D-790	840-1300	740-1330
Resistencia a la compresión	kg/cm ²	D-695	773-1330	740-1260
Resistencia al impacto	kg/cm ²	D-256 A	1.9	2.4

Continúa



Conductividad térmica	10^{-4} cal·cm /sec·cm ² ·°C	C-177	4.6-6.0	4.0-6.0
------------------------------	---	-------	---------	---------

Fuente:(IdePlas, 2012)

Cuadro 9

Características del acrílico

Resistencia química	Resistente a la mayoría de las sustancias, incluyendo soluciones de álcalis y ácidos. El contacto con algunos selladores y juntas puede causar el cuarteamiento de la lámina bajo ciertas condiciones.
Resistencia a la intemperie	Resistente a la exposición de radiación solar intensa, frío extremo, a cambios súbitos de temperatura, a la brisa salada y a otras condiciones meteorológicas.
Estabilidad dimensional	La lámina de acrílico se expande y contrae a consecuencia de cambios de temperatura y niveles de humedad, pero no existe reducción en sus dimensiones con el paso del tiempo.
Conductividad eléctrica	Excelente aislante eléctrico, sin embargo, causa una carga estática en la superficie la cual atrae partículas de polvo y pelusa.

Fuente: (Ultra Plas S.A., 2017)

Se sometió a un análisis estático para analizar el comportamiento de la placa cuando es sometido a la carga máxima.

$$W = m * g \quad \text{Ec.0-1}$$

$$m_{m\acute{a}x} = 246g$$

$$Carga\ m\acute{a}x\ (N) = 0.246\ kg * 9.81\ m/s^2$$

$$Carga\ m\acute{a}x\ (N) = 2.41326\ N$$

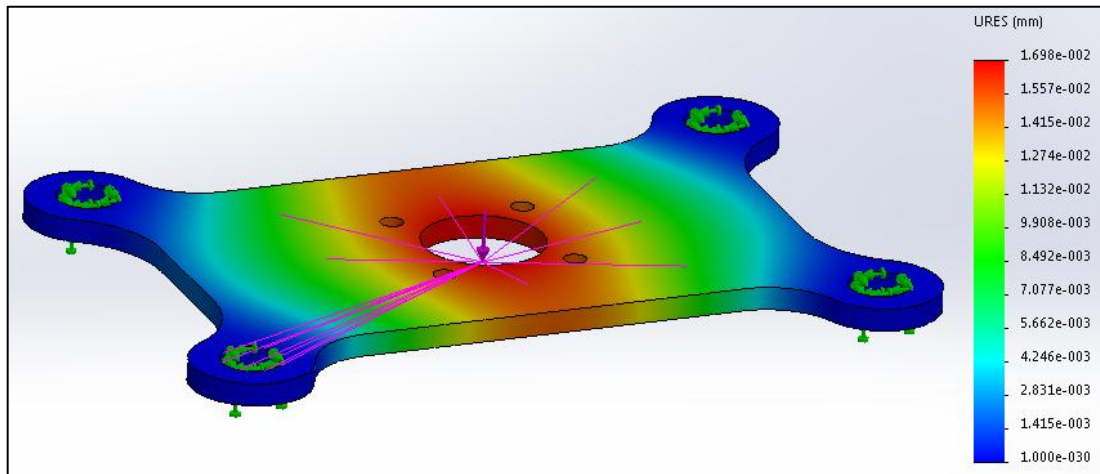


Figura 40: Desplazamiento en la placa del motor generado por la carga

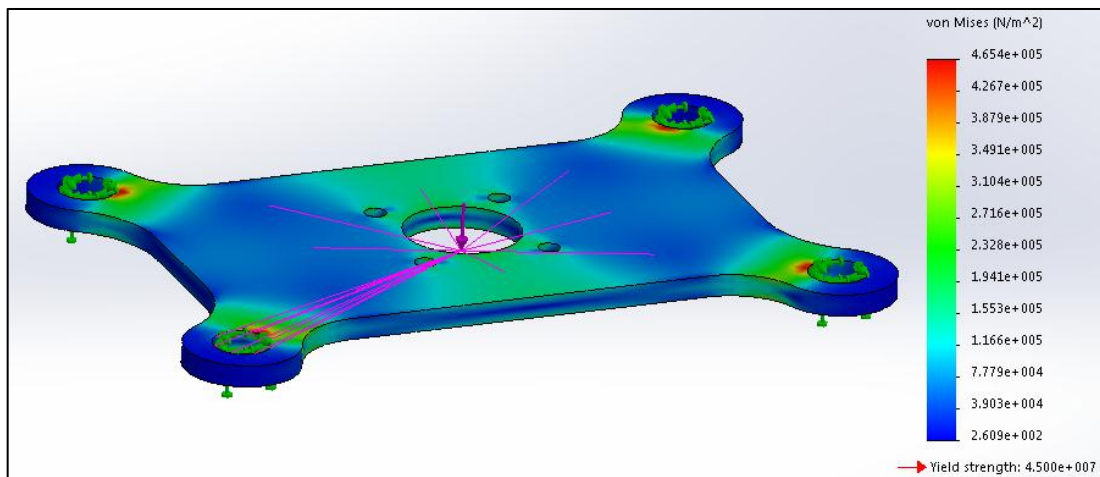


Figura 41: Tensión en la placa del motor generado por la carga

Como se puede observar, el desplazamiento máximo generado por la carga es de $1.698e^{-2} \text{ mm}$ en su punto más crítico, y la **Figura 41** nos indica que el material no sufrirá un fallo elástico cuando este sea sometido a esta carga.

3.2.1.4. Amortiguadores

La consideración principal a tomó en cuenta para la selección de los amortiguadores, es que soporten el peso de alrededor de los 300g, se adiciona un peso máximo de la base del motor de 40g a los 260g que pesa la gimbal.

La disponibilidad de los amortiguadores no es muy alta, ya que se encuentran a precios muy elevados, pero se seleccionó amortiguadores que soportan una carga de 400g cada uno.



Figura 42: Amortiguadores

Se sometió a un análisis estático para comprobar el desplazamiento máximo generado por el peso que debe soportar, utilizando la Ec.3-1.

$$m_{m\acute{a}x} = 300g$$

$$Carga\ m\acute{a}x\ (N) = 0.30\ kg * 9.81\ m/s^2$$

$$Carga\ m\acute{a}x\ (N) = 2.943\ N$$

Dado que se utilizan 4 amortiguadores, la carga aplicada a cada amortiguador es de:

$$Carga\ individual\ (N) = \frac{2.943}{4} = 0.7357\ N$$

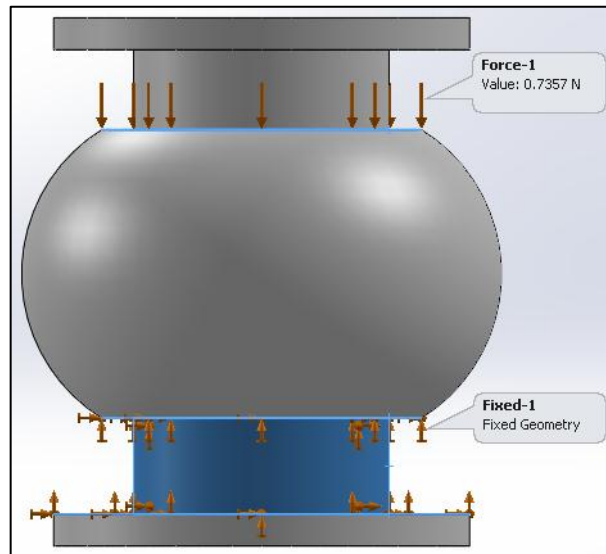


Figura 43: Carga en el amortiguador

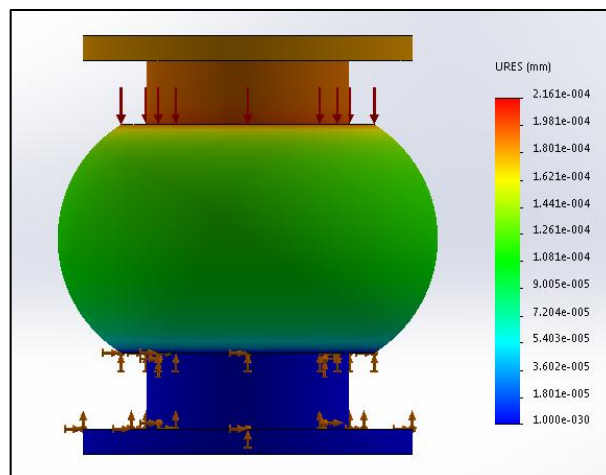


Figura 44: Desplazamiento del amortiguador generado por la carga

Como se puede observar, en la **Figura 44**, el desplazamiento máximo generado por la carga es de $2.161e^{-4} \text{ mm}$ en su punto más crítico, por lo que los amortiguadores seleccionados cumplen con el requisito de soportar la carga a la que son sometidos.

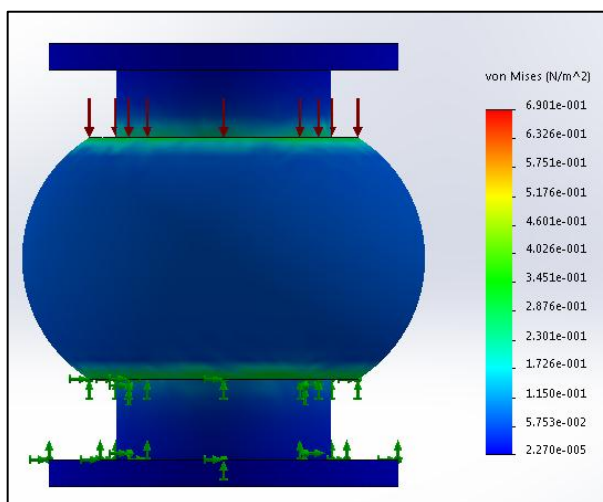


Figura 45: Tensión en el amortiguador generado por la carga

La tensión máxima generada por la carga en el amortiguador es de 0.6901 N/m^2 , lo que indica que el amortiguador no sufrirá un fallo elástico al someterlo a esta carga.

3.2.1.5. Selección del controlador de gimbal

En la actualidad existe una gran variedad de controladores de gimbal, cada uno con sus diferentes prestaciones, dimensiones, fabricantes, tamaños, etc. Debido a eso, a continuación, se presenta una tabla de con los controladores de gimbal más utilizados con base en los requerimientos que se tomaron en cuenta para la selección del mismo.

Cuadro 10

Selección del controlador de Gimbal

	<i>Storm32</i>	<i>VR Gimbal</i>	<i>AlexMos SimpleBGC</i>	<i>Tarot</i>	<i>EvvGC</i>
Código libre	Si	Si	No	No	Si
Compatible con Pixhawk	Si	Si	Si	Si	Si

Continúa



Permitir el control de los motores mediante una SBC	Si	Si	Si	Si	Si
Controlador de 3 ejes	Si	Si	Si	Si	Si
Facilidad de uso	Fácil	Fácil	Fácil	Fácil	Fácil
Disponibilidad en Ecuador	No	No	Si	No	No
Dimensiones	50x50 mm	51x70.5x13 mm	50x50 mm	99x88.5x105 .6 mm total gimbal	50x50x10 mm
Peso	13g	24g	17g y 15g	178g peso total gimbal	18g

Fuente: (OlliW's Bastelseiten, 2017), (Quatum, 2017), (VR Gimbal, 2017), (**Tarot Aviation Technology, 2015**)

Como se puede observar, en el **Cuadro 10**, los controladores de gimbal actuales poseen grandes prestaciones, por lo que elegir cualquiera de los mencionados anteriormente no representa ningún problema para la realización del proyecto, es por eso, que se tomó en cuenta como factor decisivo la disponibilidad del mismo en Ecuador, siendo el único que se encontró, al momento de realizar esta búsqueda, el controlador AlexMos SimpleBGC.

El modelo de controlador que se dispone es AlexMos SimpleBGC V2.1 de Basecam Electronics, siendo este una versión antigua de los modelos actuales, pero cumple los requerimientos necesarios para la realización del proyecto, ya que cuenta con el control para 2 ejes en su placa principal y una expansión para el 3er eje.

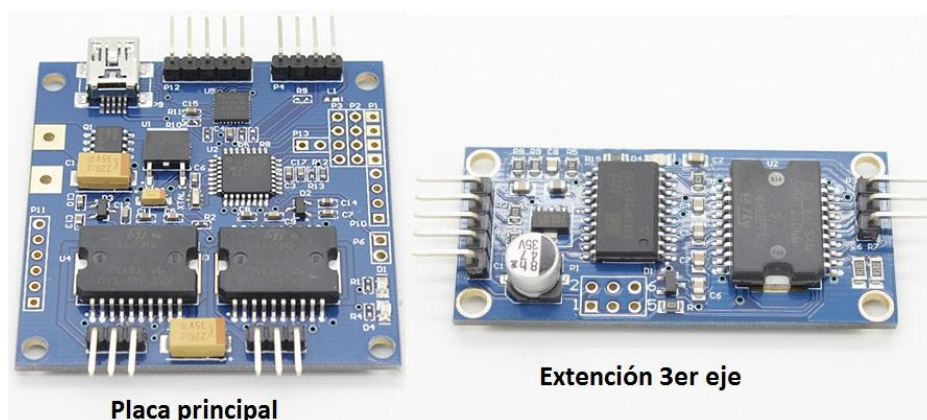


Figura 46: Controlador de gimbal

El dispositivo viene con una IMU MPU6050, el cual fue el primer dispositivo de rastreo de movimiento integrado de 6 ejes del mundo, que combina un giroscopio y un acelerómetro de 3 ejes cada uno, además de un procesador de movimiento digital (DMP), teniendo dimensiones de 4x4x0.9mm. Debido a su sensor de bus I2C dedicado, acepta directamente entradas de un compás externo de 3 ejes, para de esta manera proporcionar una salida de información de 9 ejes. (InvenSense, 2013)



Figura 47: IMU MPU6050

La ventaja de este dispositivo es que, a pesar de no ser de código libre, permite realizar el control de los motores ya sea a través de la estación de control, del radio control o de una placa de ordenador pequeña, ya que posee pines de entrada colocados para ese propósito.

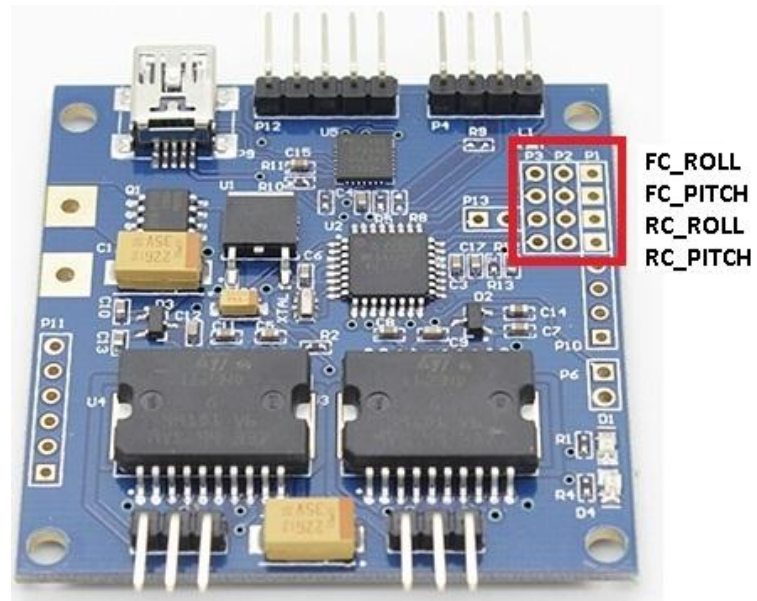


Figura 48: Pines para control de motores

3.2.1.6. Ensamble de la gimbal

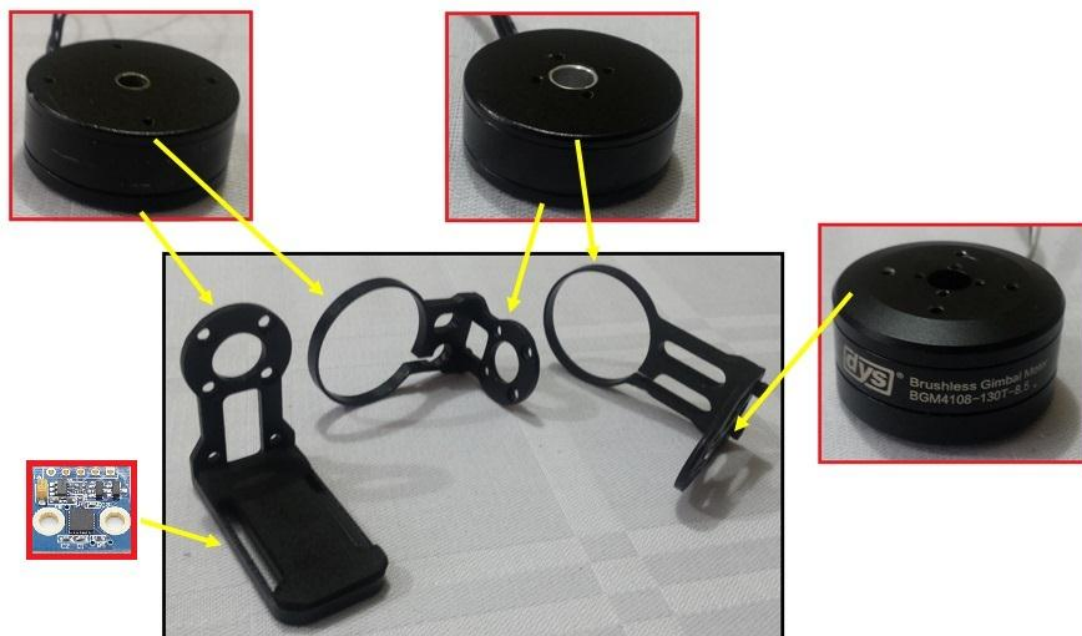


Figura 49: Estructura y posición de los motores e IMU

Debido a que posee un tamaño reducido la IMU, esta se colocó debajo de la pieza que sirve como base para la cámara. De esta manera se evitó que el cableado de la misma obstruya en el funcionamiento de la estructura.

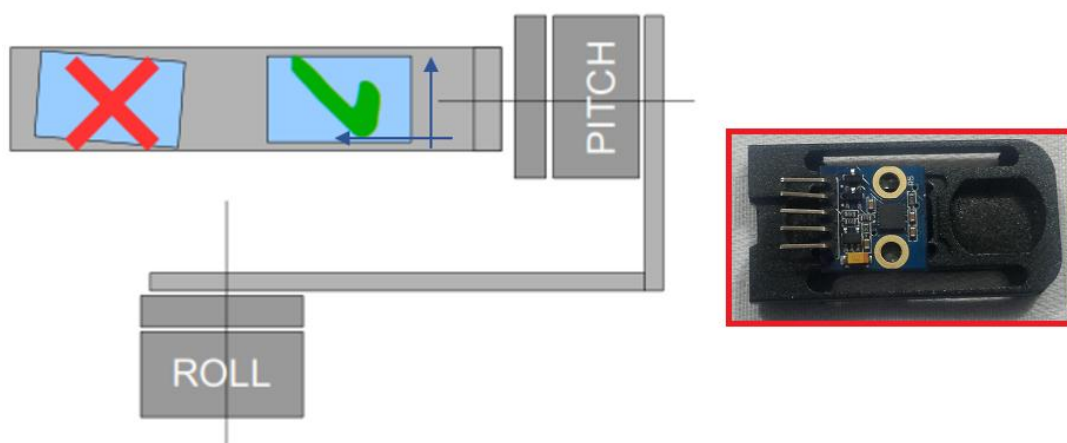


Figura 50: Posición de la IMU en la estructura

3.2.1.7. Dispositivo de captura de imágenes

La captura de video para evaluación del algoritmo de minimización de movimientos fantasma realizado por el Prof. Wilbert G. Aguilar Ph.D. se lo realizó con una cámara VGA 640x480 a 15fps, por lo que se tomó en cuenta esas especificaciones para la selección de la cámara.

Los requerimientos que se tomaron en cuenta para la selección de la cámara son: disponibilidad en el mercado, precio, dimensiones, peso y resolución. Siendo la cámara “Webcam ComQtech Serie WCC21”, que se montó en el dron, cuyas especificaciones se presentan en el **Cuadro 11**.

Cuadro 11

Especificaciones técnicas de la cámara ComQtech

Conexión	USB - Plug and play
Dimensiones	52.65 x 51.5

Continúa



Salida de Video	640 x 480 hasta 30fps
	Lente ajustable
Formato de Video	VGA
Resolución máxima	8MP
Ajuste de luz y compensación de color	Automático
Micrófono	Incorporado
Sistemas Operativos	Todo sistema de operativo de PC
Peso	20.9g

Fuente: Caja de la cámara

También, se utilizó la cámara “G-vision Inkfish A1 Sports Camera”, para realizar capturas de video usadas para sacar la planta del dron y realizar la prueba de funcionamiento del filtro de Kalman, cuyas especificaciones se presentan en el **Cuadro 12**.

Cuadro 12

Especificaciones técnicas de la cámara G-vision Inkfish A1

Método Impermeable	Estuche plástico a prueba de agua
Profundidad de inmersión bajo el agua	30m
DSP (Procesamiento de Señal Digital)	Sunplus SPCA6330
Sensor CMOS	OV2710
Ángulo de Visión	170°
Enfoque	12cm ~ infinito
Audio de Entrada	Micrófono
Audio de Salida	Altavoz
HDMI de salida	Tipo D
Interface USB	USB2.0, 5 Pin USB
Almacenamiento	Soporta TF(TransFlash) (Hasta 32GB)
LCD	1.5' LTPS LCD
Batería	3.7V, 900mAh li-ion batería recargable
Tiempo de duración	<120 mins

Continúa



Dimensiones	59 x 41 x 24.5 mm
Control WIFI	Wifi
Salida de Video	1920 x 1080 30fps, 1280 x 720 60fps
Formato de Video	H.264 MOV
Modo de Grabación de Video	Grabación por bucle
Pixeles Efectivos	12M/8M/5M/2M
Formato de Imagen	JPEG
Audio	WAV (taza de bits: 176kbps)
Ráfaga	3 fotos por segundo
Reproducción	Soportado
Modos de tomar fotos	Una toma/ cronómetro / Tomas continuas
Cámara trabaja con computador	Soportado
Rotación de 180°	Soportado
Sistemas Operativos	Windows98/me/XP/vista/7 MacOS/Linux

Fuente: [G-vision S&T Limited, 2016]

Para sostener la cámara, se utilizó una cinta de velcro, la cual permite colocar una amplia variedad de tipos de cámaras de forma sencilla. La cámara que se utilizó es más liviana que las cámaras deportivas de estilo GoPro, debido a esto se tiene un desequilibrio en el eje Roll. Los motores seleccionados son más sensibles a un desequilibrio debido a su tamaño reducido, es por eso que se procedió a equilibrar el sistema añadiendo un peso extra, de esta manera se evita un sobreesfuerzo al motor ya que a largo plazo podría causar un daño, así como generar dificultad en la estabilización de la gimbal.

3.2.1.8. Cálculo para equilibrar la estructura de la ISP

Se tiene las siguientes fuerzas en el sistema:

$$F1 = \text{Peso pieza 4}$$

$$F2 = \text{Peso pieza 3} + \text{Peso tornillos} + \text{cables}$$

$$F3 = \text{Peso motor}$$

$$F4 = \text{Peso pieza 2} + \text{Peso tornillos}$$

$$F5 = \text{Peso cámara} + \text{Peso velcro}$$

$$F6 = \text{Peso pieza 1} + \text{Peso IMU}$$

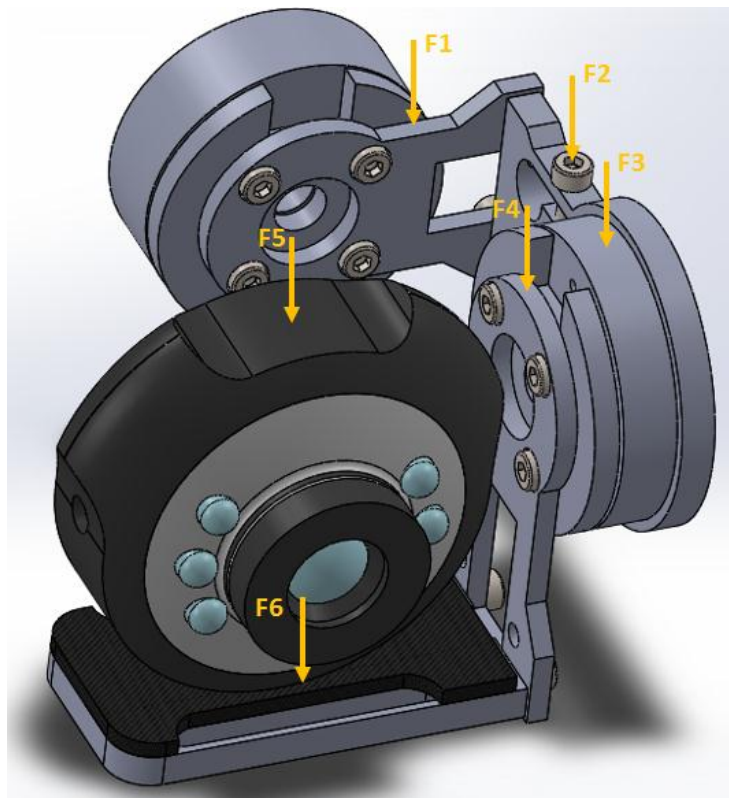


Figura 51: Fuerzas aplicadas en el sistema gimbal

Donde:

$$F1 = 0.0028 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.0274 \text{ N}$$

$$F_2 = (0.0041 \text{ kg} + 0.00135 \text{ kg} + 0.0078 \text{ kg}) \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.1299 \text{ N}$$

$$F_3 = 0.0415 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.4071 \text{ N}$$

$$F_4 = (0.0038 \text{ kg} + 0.0027 \text{ kg}) \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.0637 \text{ N}$$

$$F_5 = (0.0209 \text{ kg} + 0.0023 \text{ kg}) \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.2275 \text{ N}$$

$$F_6 = (0.0058 \text{ kg} + 0.0015 \text{ kg}) \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.0716 \text{ N}$$

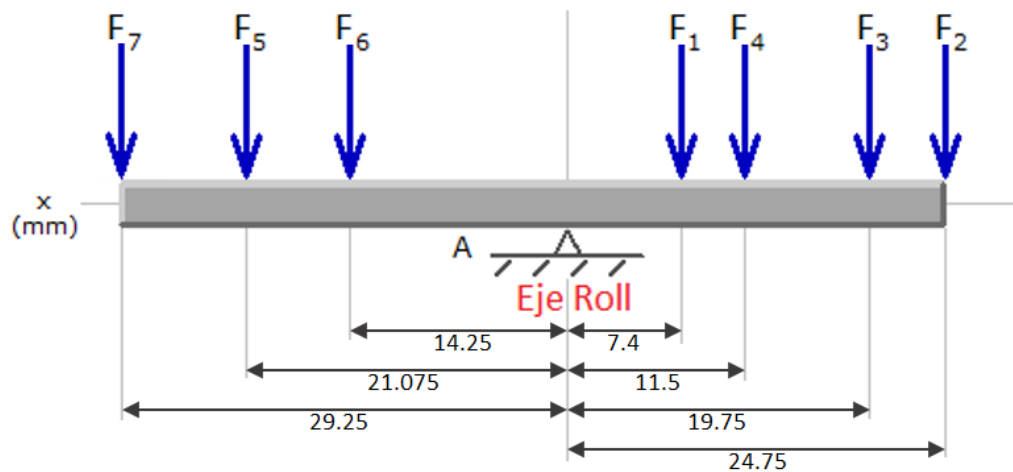


Figura 52: Diagrama de fuerzas de la gimbal

Se procedió a calcular la F7, la cual representa el peso adicional que se necesita para equilibrar al sistema.

Se tiene la siguiente sumatoria de momentos:

$$\sum M_A = 0 \quad \text{Ec.0-2}$$

$$F_1 \times D_1 + F_2 \times D_2 + F_3 \times D_3 + F_4 \times D_4 - F_5 \times D_5 - F_6 \times D_6 - F_7 \times D_7 = 0$$

$$0.0274 \times 0.0074 + 0.1299 \times 0.02475 + 0.4071 \times 0.01975 + 0.0637 \times 0.0115 \\ - 0.2275 \times 0.021075 - 0.0716 \times 0.01425 - F_7 \times 0.02925 = 0$$

$$0.006377 - F_7 \times 0.02925 = 0$$

$$F_7 = 0.2180 \text{ N}$$

$$F_7 = \text{Peso} \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.2180 \text{ N}$$

$$\text{Peso} = 0.02222 \text{ kg}$$

$$\text{Peso} = 22.22 \text{ g}$$

El peso adicional se lo colocó en la parte inferior de la pieza que sirve como base de la cámara, en la posición que se indica en la **Figura 53**.

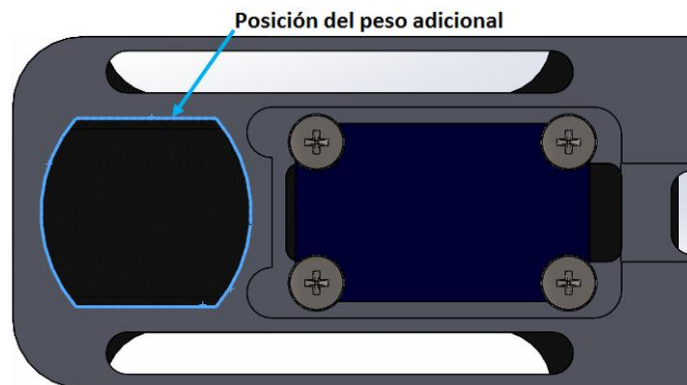


Figura 53: Posición del peso adicional

3.2.2. Ordenador on-board

El algoritmo de minimización de movimientos fantasma realizado por el Prof. Wilbert G. Aguilar Ph.D. fue evaluado en una laptop que posee las siguientes especificaciones: Procesador Intel Core i7-2670QM, velocidad de procesamiento de 2.2 GHz con Turbo Boots hasta 3.1 GHz y 16Gb de RAM. (Aguilar & Angulo, 2014b)

3.2.2.1. Selección del Single-Board Computer

La investigación de los diferentes modelos de SBC se la realizó en el 1er trimestre del año 2016, teniendo en consideración que este debe poseer especificaciones similares a la laptop mencionada anteriormente, ya que este programa no ha sido probado con especificaciones inferiores.

A continuación, se presentará una tabla con las especificaciones técnicas más importantes de algunos ordenadores de placa pequeña que poseen características similares a este requisito.

Cuadro 13
Especificaciones técnicas de SBC's

	<i>Raspberry Pi 3</i>	<i>Nvidia Jetson TX1</i>	<i>Odroid C2</i>	<i>Odroid XU4</i>	<i>Arndale Octa</i>
Sistema en chip (SoC)	Broadcom BCM2835	Nvidia Tegra X1	Amlogic S905	Samsung Exynos 5 Octa	Samsung Exynos 5420 Octa
CPU	ARM Cortex-A53	ARM Cortex-A57 ARM Cortex-A53	ARM Cortex-A53	ARM Cortex-A15 ARM Cortex-A7	ARM Cortex-A15 ARM Cortex-A7
	4	4 - 4	4	4 - 4	4 - 4
	1.2 GHz	1.9 GHz 1.3 GHz	1.5 GHz	2 GHz 1.4 GHz	1.8 GHz 1.3 GHz
GPU	VideoCore IV 400MHz	Nvidia GM20B 1000MHz	ARM Mali-450MP 750MHz	ARM Mali-T628 600 MHz	ARM Mali T-628 MP6
RAM	1 GB – LPDDR2	4 GB – 64 bit LPDDR4	2 GB – 64 bit DDR3	2 GB – 64 bit DDR3L	3GB LPDDR3
Precio	35 \$	599.99 \$	40 \$	74 \$	199 \$

Fuente: (Premier Farnell Company, 2017), (NVIDIA Corporation, 2017), (Hardkernel co., Ltd, 2017), (Arndaleboard, 2017)

Como se puede observar en el **Cuadro 13**, la SBC Odroid XU4 es la placa que posee mejores prestaciones a un bajo costo, además, posee la última generación de Octo-Core 5422 ultra rápida. Es una computadora SoC construida por una compañía de Hardkernel de Corea del Sur. Se introdujo en el mercado en verano del 2015 como la tarjeta de computadora ARM Octa-Core big.LITTLE de alto rendimiento más asequible. (Ivković, Veljović, Ranđelović, & Veljović, 2016)

En la **Figura 54** se muestra el desempeño de varios ordenadores de placa pequeña, con el fin de conocer cuan significativa es la diferencia existente entre la placa Odroid XU4 con su competencia.

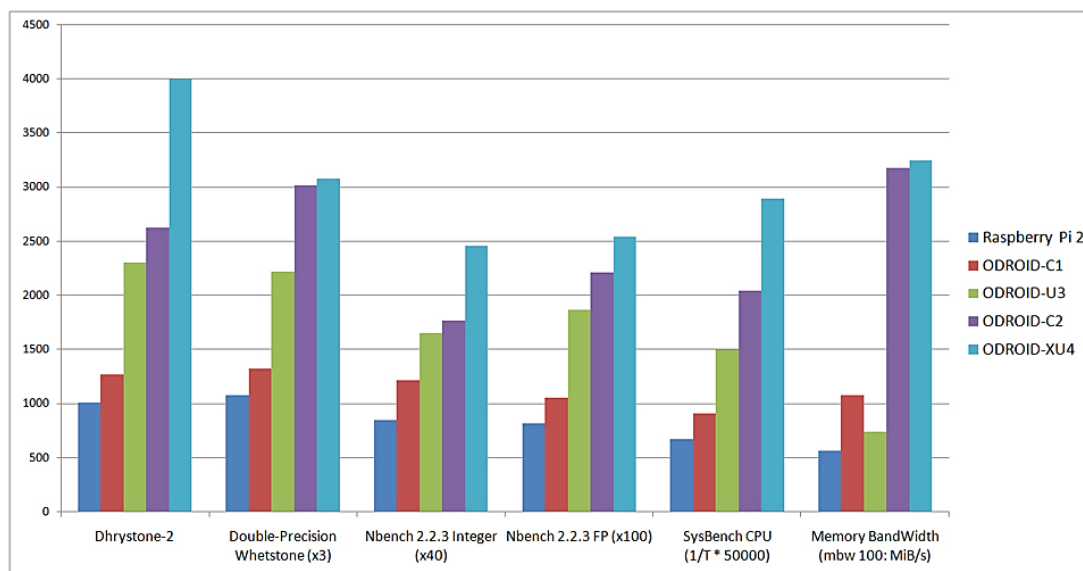


Figura 54: Desempeño de CPU/RAM de varias SBC

Fuente: (Hardkernel co., Ltd, 2017)

Este dispositivo permite instalar en su sistema los sistemas operativos Ubuntu, Android, Fedora, ARCHLinux, Debian, Openelec. Para ello cuenta con dos tipos de memoria que pueden ser usados, ya sea la tarjeta microSD o en la memoria eMMC.

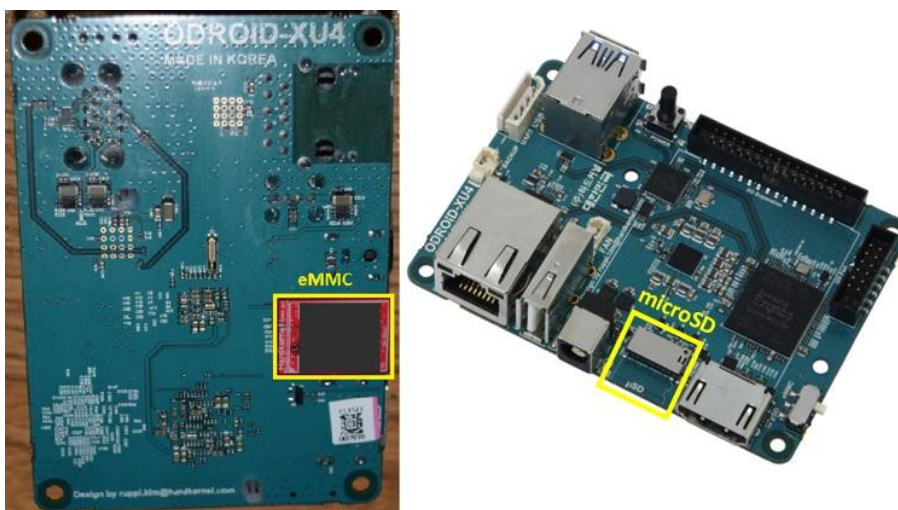


Figura 55: Memorias eMMC y microSD de Odroid XU4

Tiene un incremento en su tasa de lectura y escritura en la memoria eMMC comparada con la memoria SD, como se muestra en la **Figura 56**.

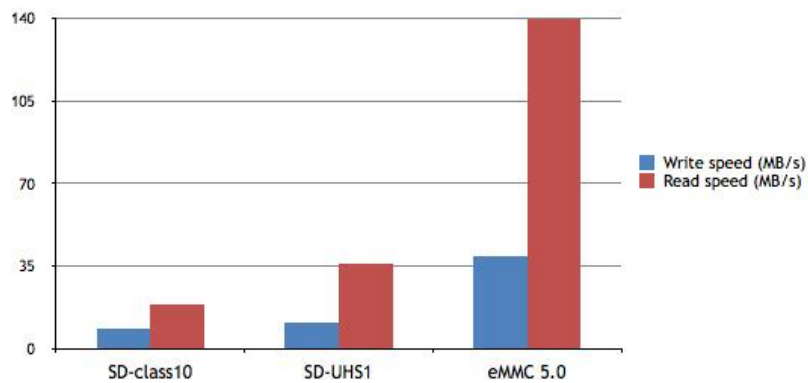


Figura 56: Diferencia de rendimiento entre SD y eMMC

Fuente: (Hardkernel co., Ltd, 2017)

Adicional a esto, la memoria eMMC viene con un adaptador, el cual se utilizó para la instalación del sistema operativo.

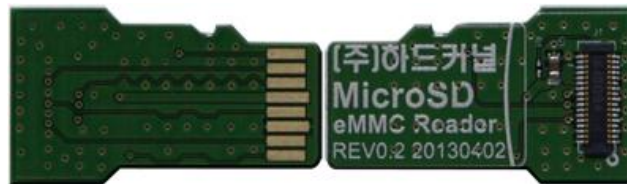


Figura 57: Adaptador para memoria eMMC

A pesar de poseer dimensiones reducidas y de contar con un peso de 60.7g, el dispositivo cuenta con 2 puertos USB 3.0, 1 puerto USB 2.0, 1 puerto de Ethernet y 1 puerto HDMI 1.4a.

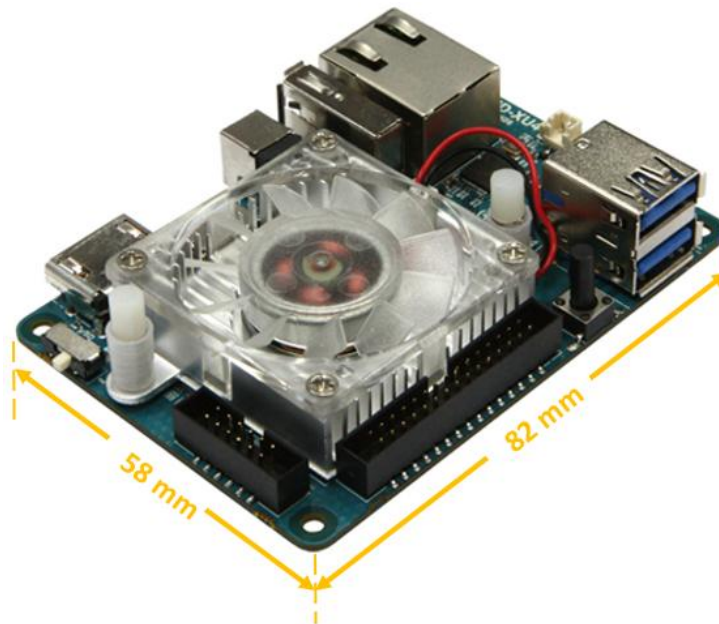


Figura 58 Dimensiones del Odroid XU4

También cuenta con 42 pines nativos GPIO, los cuales operan a 1.8V, sin embargo, debido a que muchos dispositivos trabajan a 3.3V o 5V, se cuenta con un adaptador llamado Shifter Shield, que convierte a estos en 40 pines, los cuales pueden trabajar a 3.3V o 5V.

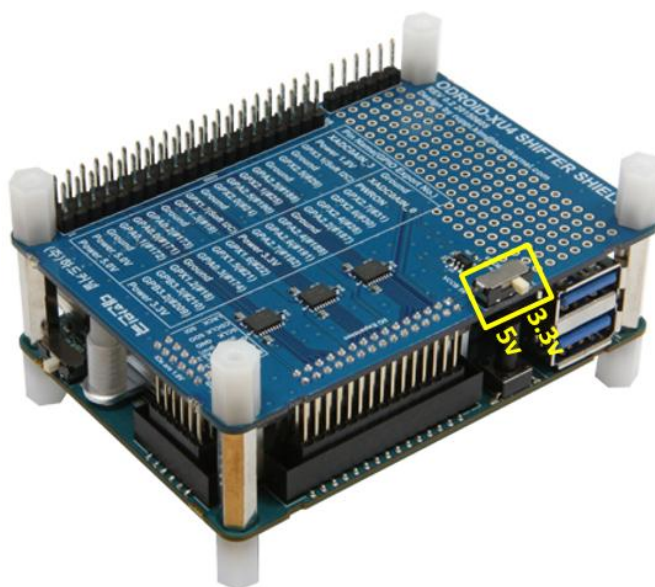


Figura 59: Adaptador Shifter Shield

La alimentación de esta placa es de 5V a 4A, debido a esto, la energización de este dispositivo se la realizó mediante la regulación del voltaje de la batería a través del DROK, el cual realiza la regulación de voltaje de 12V/24V a 5V - 5A/25W. Este regulador del voltaje posee un peso de 104.3g.



Figura 60: Dimensiones del regulador de voltaje DROK

La **Figura 61** muestra la conexión que se realizó para energizar el Odroid XU4.

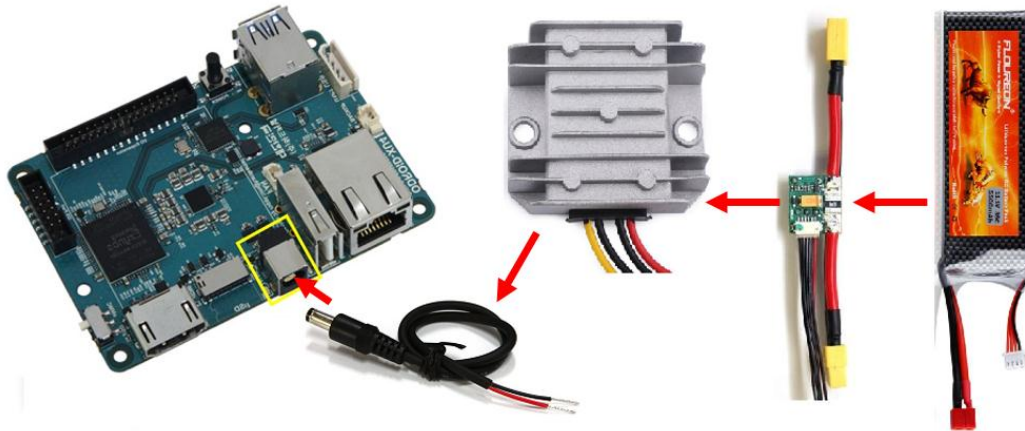


Figura 61: Energización del Odroid XU4

3.2.2.2. Diseño de la carcasa del Odroid XU4

Se realizó el diseño de una carcasa con el fin de proteger el ordenador de placa pequeña, el cual proporciona mayor estabilidad a la placa, de esta manera se evitarán daños por movimientos inesperados. Para ello, en el diseño se consideró los sujetadores que permiten agregar el Shifter Shield al Odroid XU4, por lo que se decidió emplear impresión 3D.

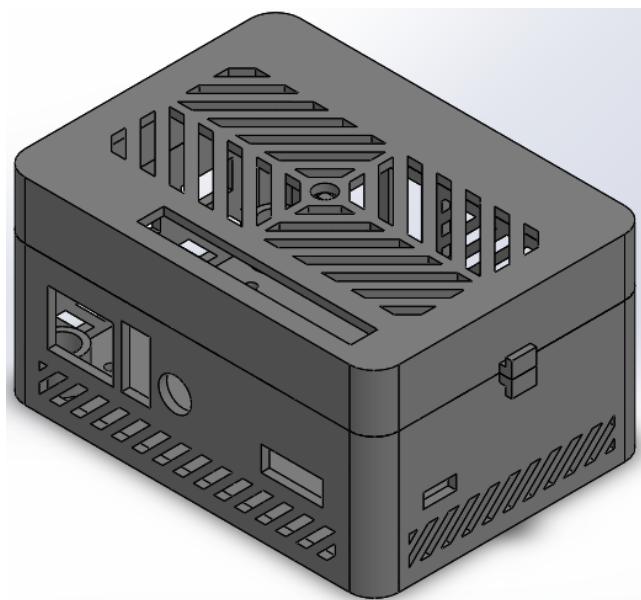


Figura 62: Diseño de la carcasa para el Odroid XU4 – Vista isométrica 1

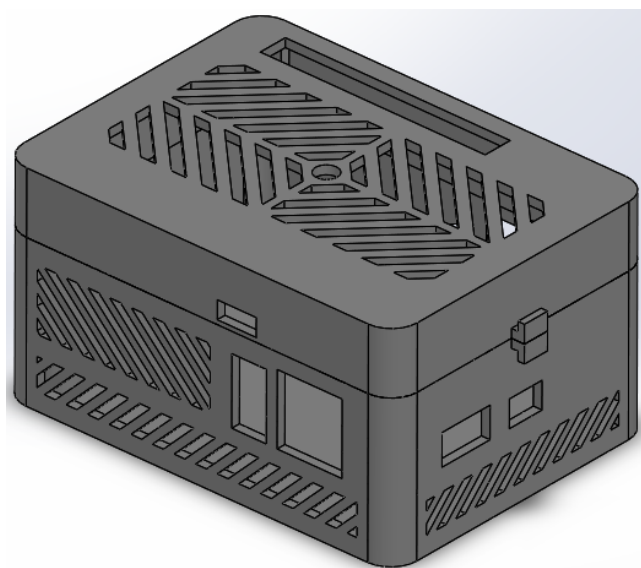


Figura 63: Diseño de la carcasa para el Odroid XU4 – Vista isométrica 2

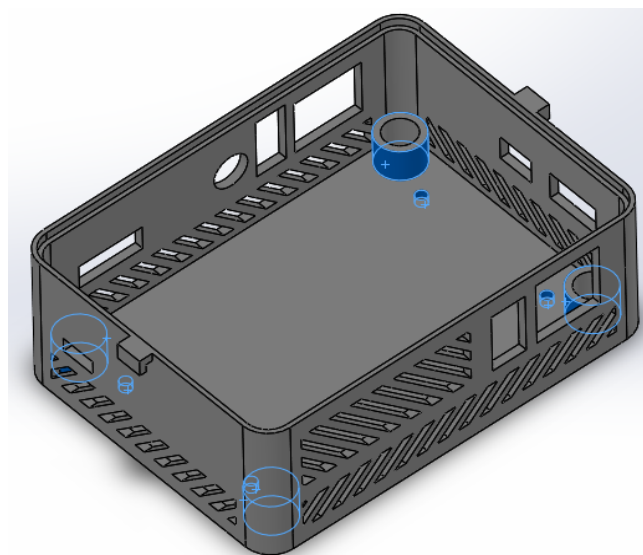


Figura 64: Puntos de sujeción del Odroid XU4

El requisito principal para la selección del material a usar en la impresión 3D es que este no genere un peso adicional significativo para el dron, de la misma manera, se consideró su costo y tiempo de impresión.

La mayoría de empresas que ofrecen el servicio de impresión 3D a precios asequibles dan las opciones de ABS, PLA y Nylon, es por eso, que a continuación, se presenta una tabla con las propiedades de ellos.

Cuadro 14

Propiedades de materiales para impresión 3D

	<i>Unidades</i>	<i>PLA</i>	<i>ABS</i>	<i>Nylon</i>
Densidad	g/cm ³	1.25	1.03-1.38	1.14
Módulo de elasticidad	GPa	3.5	1.7-2.8	3
Elongación a la rotura	%	6	3-75	20-200
Resistencia a la flexión	MPa	80	69-97	76
Resistencia a la tracción	MPa	50	33-110	73
Temperatura de transición vítrea	°C	60	100	49.4

Continúa



Biodegradable	-	Si	No	No
Reciclable	-	Si	Si	Si

Fuentes: (PHEMSA SL, 2017), (Plasticbages Industrial, S.L., 2017), (HXX, 2015)

Se tomó en cuenta que el material debe resistir temperaturas de hasta 50°C, ya que el Odroid XU4 puede alcanzar esta temperatura cuando se utiliza su procesador a su máxima capacidad.

Debido a que el PLA es uno de los materiales más utilizados en las impresiones 3D no solo por sus propiedades, sino también por ser amigable con el ambiente, además, de cumplir con el requisito mencionado anteriormente, ya que posee una temperatura de transición vítrea de 60°C, se eligió este material para la carcasa del Odroid XU4, aportando así un peso adicional de 53.5g.

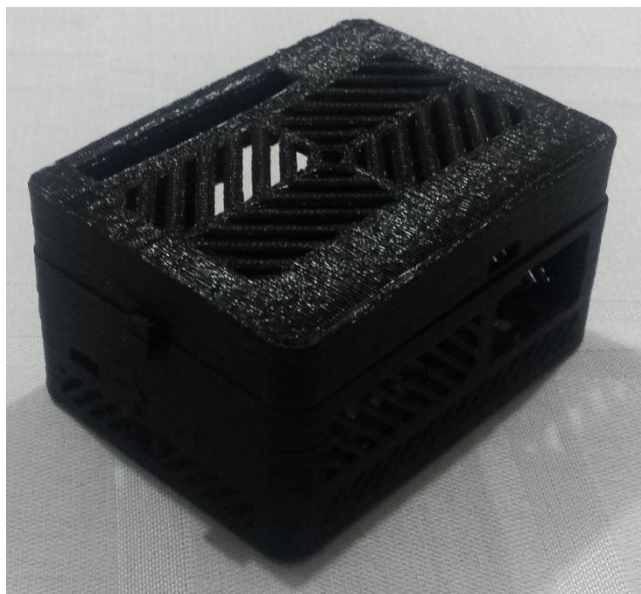


Figura 65: Carcasa del Odroid XU4

3.2.3. Diseño de la base total del sistema

La finalidad de esta base es que se pueda colocar en cualquier dron, ya sea mediante las uniones que se utilizaron o con una adaptación de acuerdo al dron que se desee usar. La base contará con todos los elementos seleccionados

anteriormente, ya que este es el sistema que realiza la estabilización de video para minimizar movimientos fantasma.

Para la realización del diseño de la base se tomó en consideración que:

- El dron seleccionado cuenta con una base para apoyar la batería, el cual es sujetado mediante 4 separadores, es por eso que se decidió utilizar estos mismos separadores en el diseño para que sujeten la base total.
- El controlador de la gimbal y su extensión del 3er eje cuentan en su placa con 4 orificios para realizar el acople de los mismos a la base mediante separadores.
- La batería se sujetó con cinta velcro. Debido al peso que poseen las baterías de litio de 11.1V 5500mAh, el cual es alrededor de los 400g, se la colocó cercano al centro de gravedad del dron.

Teniendo en cuenta la información mencionada anteriormente se procedió a realizar el diseño de la base, finalizando así con las siguientes ubicaciones de los elementos:

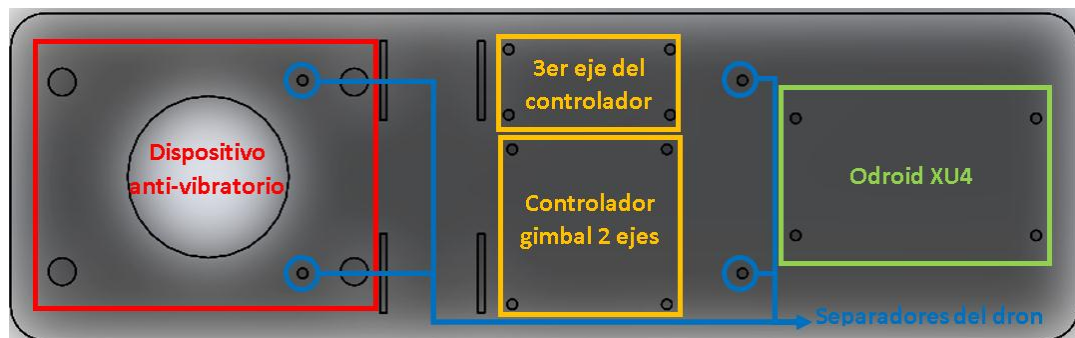


Figura 66: Ubicación de los componentes en la base - 1

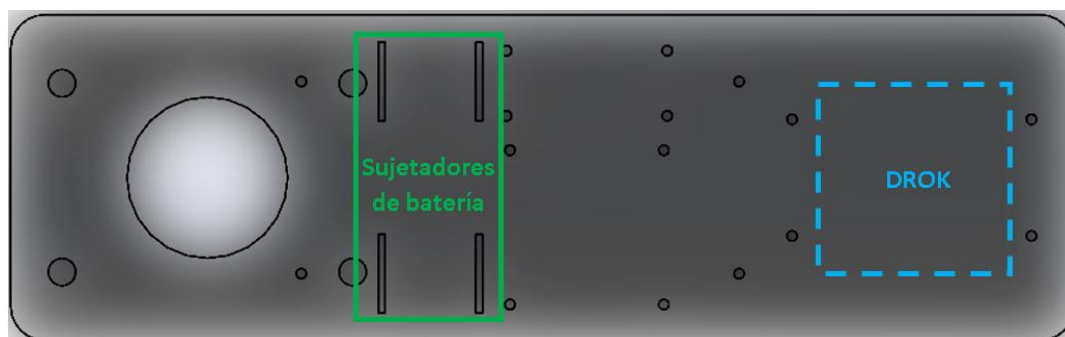


Figura 67: Ubicación de los componentes en la base - 2

Se sometió a un análisis estático para analizar el comportamiento de la base cuando es sometido a la carga de todos los componentes del sistema. Para ello se tiene que:

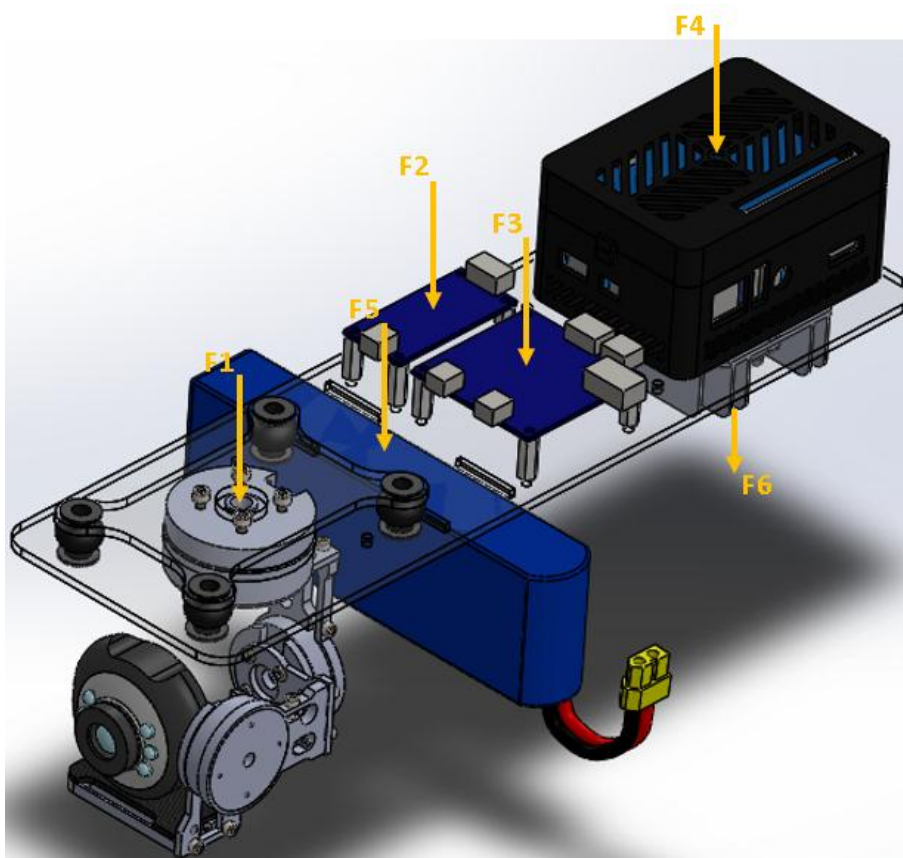


Figura 68: Fuerzas aplicadas en la base

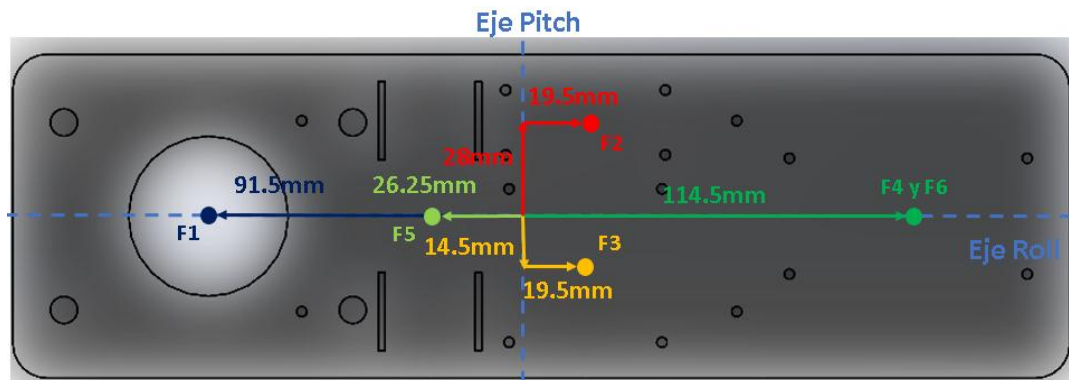


Figura 69: Ubicación de las fuerzas aplicadas en la base

$$F1 = \text{Peso ISP}$$

$$F2 = \text{Peso controlador 3er eje de gimbal}$$

$$F3 = \text{Peso controlador de gimbal}$$

$$F4 = \text{Peso odroid y carcasa}$$

$$F5 = \text{Peso batería y velcro}$$

$$F6 = \text{Peso DROK}$$

Donde:

$$F1 = 0.2754 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 2.7018 \text{ N}$$

$$F2 = 0.0167 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.1638 \text{ N}$$

$$F3 = 0.0244 \times 9.81 \frac{\text{m}}{\text{s}^2} = 0.2393 \text{ N}$$

$$F4 = 0.1596 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 1.5656 \text{ N}$$

$$F5 = 0.4137 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 4.0583 \text{ N}$$

$$F6 = 0.1043 \text{ kg} \times 9.81 \frac{\text{m}}{\text{s}^2} = 1.0231 \text{ N}$$

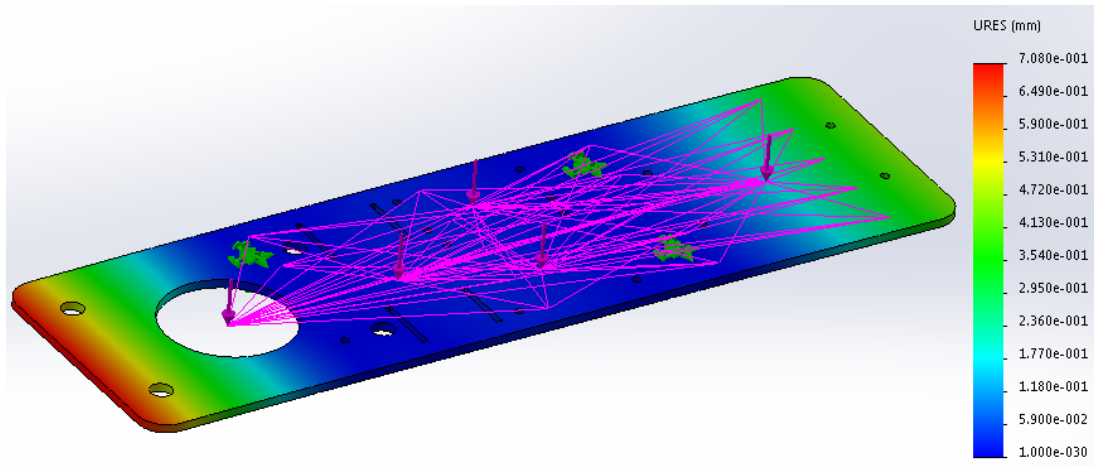


Figura 70: Desplazamiento en la placa del motor generado por la carga

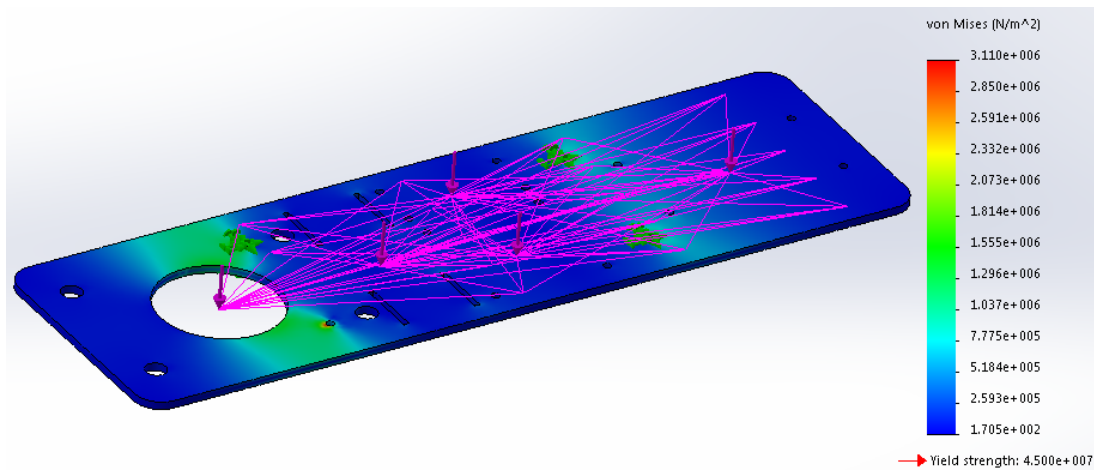


Figura 71: Tensión en la placa del motor generado por la carga

Como se puede observar, en la **Figura 70**, el desplazamiento máximo generado por las cargas es de $7.080e^{-1} \text{ mm}$, y la **Figura 71**, indica que la base no sufrirá un fallo elástico.

Debido al reducido espacio que se tiene para la distribución de los componentes en la base, esta no está equilibrada, por lo que se procedió a calcular cuan significativo es este.

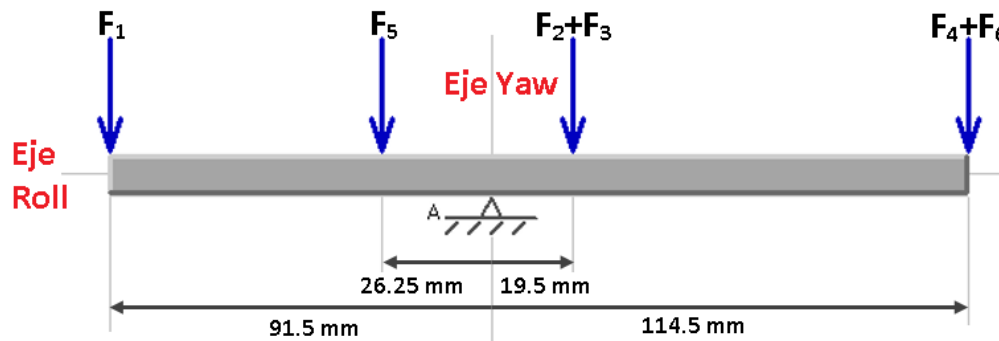


Figura 72: Diagrama de fuerzas de la base

Se tiene la siguiente sumatoria de momentos:

$$\sum M_A$$

$$\sum M_A = (F_2 + F_3) \times D_3 + (F_4 + F_6) \times D_4 - F_1 \times D_1 - F_5 \times D_2 \quad \text{Ec.0-3}$$

$$\begin{aligned} \sum M_A &= (0.1638 + 0.2393) \times 0.0195 + (1.5656 + 1.0231) \times 0.1145 - 2.7018 \\ &\quad \times 0.0915 - 4.0583 \times 0.02625 \end{aligned}$$

$$\sum M_A = 0.049467 \text{ N} \cdot \text{m}$$

$$\sum M_A = 49.467 \text{ N} \cdot \text{mm}$$

Esto indica que el dron posee un momento de inercia en sentido anti-horario respecto al eje Pitch, teniendo la cámara en el lado izquierdo, el cual no se equilibró debido a la limitación de peso máximo que soporta el dron, y también para evitar aumentar el riesgo de que se rompa la base en caso de sufrir algún

golpe. Sin embargo, en las pruebas realizadas se comprobó que este momento de inercia no afecta significativamente la maniobrabilidad del dron.

3.3. Conexiones del sistema de estabilización

En esta subsección se muestra las diferentes conexiones realizadas en el sistema de estabilización.

3.3.1. Diagrama de conexión del SimpleBGC

La **Figura 73** es el diagrama de conexión que se utilizó para la SimpleBGC, la extensión del 3er eje, los motores y la batería, este diagrama se encuentra en la página oficial de Basecam Electronics, pero se le modificó la conexión al 3er eje y se quitó la conexión al Autopilot, ya que eso se indicara posteriormente.

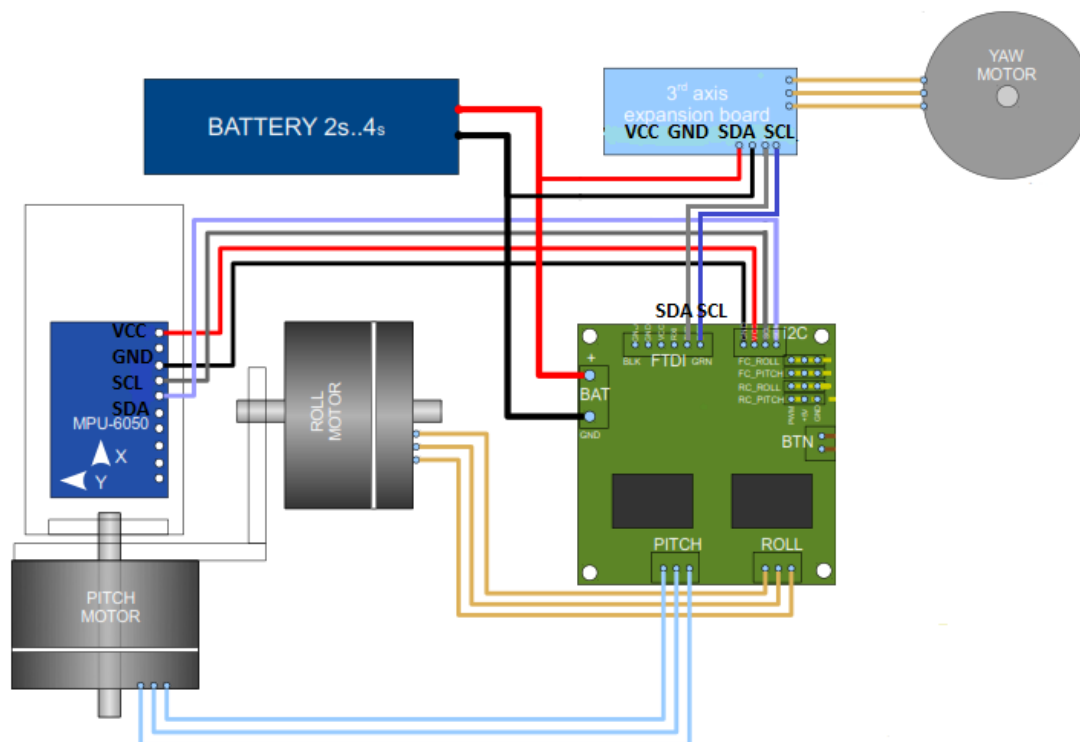


Figura 73: Diagrama de conexión de los motores al controlador

3.3.2. Conexión de la cámara a la SBC

Para realizar la captura de video, la cámara se conectó mediante un puerto USB a la SBC, ya que esta debe acceder a las imágenes para realizar el procesamiento de las mismas y posterior a eso realizar la estabilización de video. Debido a eso, el cable se lo paso a través de los motores, de esta manera se evitó adicionar un peso extra no deseado a los costados de la estructura, lo que hubiera dificultado la estabilización de la gimbal.

La cámara Webcam vino con el cable incluido, el cual estaba soldado a su placa. Debido a esto, para hacer que el sistema sea lo más modular posible, se cortó el cable. De esta manera, teniendo en cuenta la conexión de los cables del USB se puede conectar una gran variedad de cámaras, ya sea conectando directamente los cables o mediante un adaptador para puertos micro o mini USB.

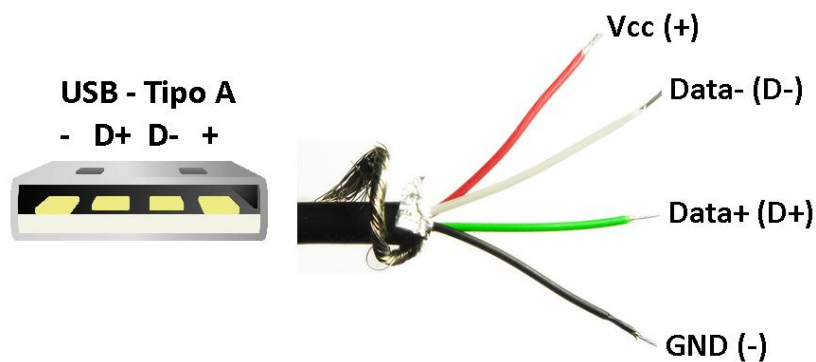


Figura 74: Conexión del cable USB

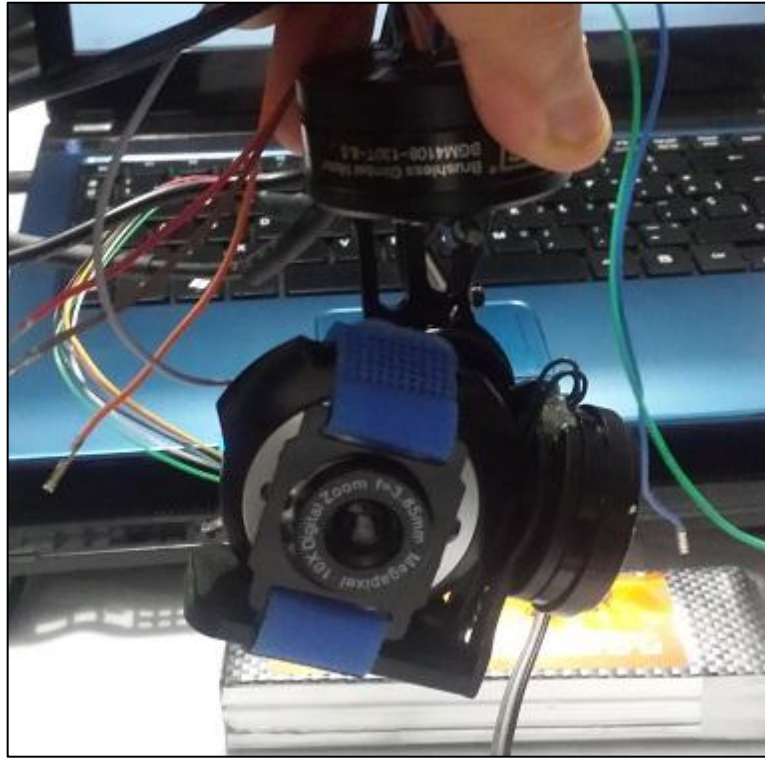


Figura 75: Ensamble de la gimbal

3.3.3. Conexiones para la envío y recepción señal de control

Se estableció el Canal 8 del receptor R9D para que controle el movimiento del eje Yaw del dron. La conexión para el envío y recepción de la señal de control del AT9 al Odroid XU4 se realizó entre el canal 2 del R9D, correspondiente al canal 8 del AT9, y el pin GPA0.2 del Odroid XU4.

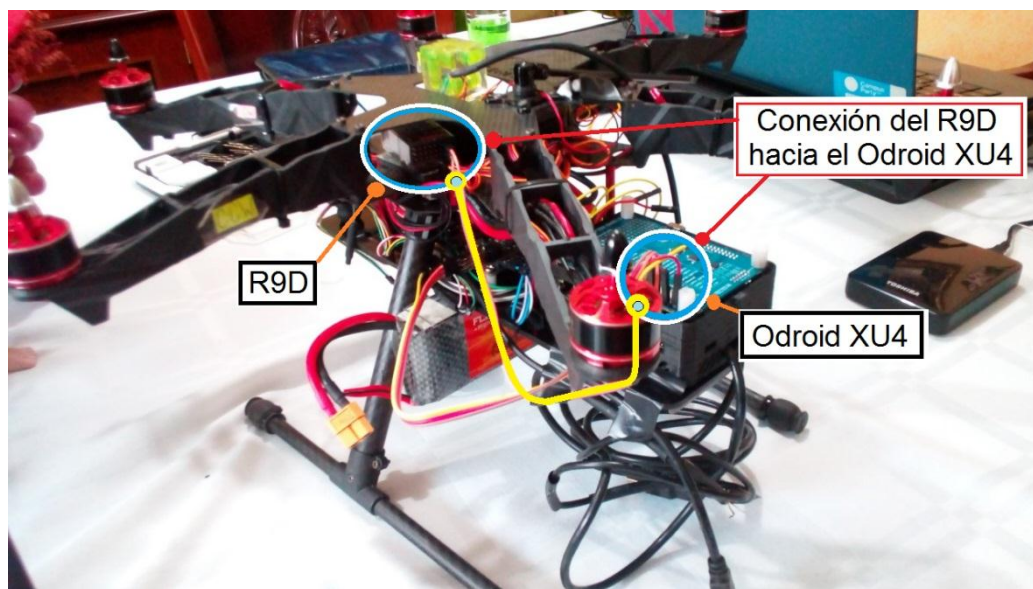


Figura 76: Conexión entre el R9D y el Odroid XU4.

La conexión para el envío y recepción de la señal de control del Odroid XU4 al SimpleBGC se realizó entre el pin GPA0.3 del Odroid XU4 y el pin RC_ROLL del SimpleBGC, junto con sus respectivos pines voltaje y tierra.

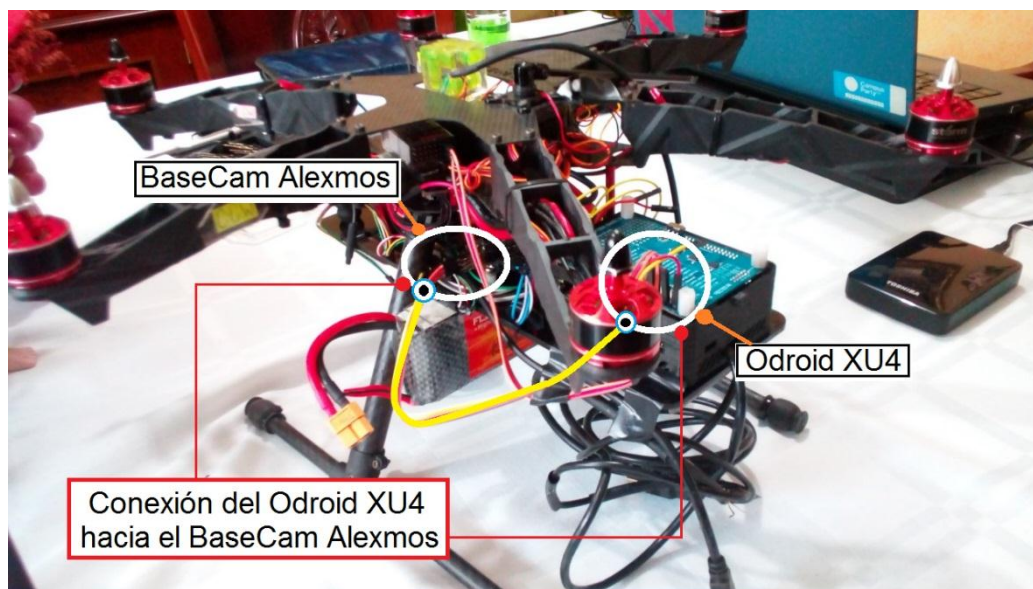


Figura 77: Conexión entre el Odroid XU4 y el SimpleBGC

3.3.4. Conexión del Odroid XU4 con el Pixhawk

El sistema operativo instalado en el Odroid XU4, viene con ROS Jade, el cual contiene librerías para la comunicación entre ambos dispositivos mediante la conexión USB (Odroid), y micro USB (Pixhawk), permitiendo la transmisión de datos por telemetría para su posterior almacenamiento y procesamiento.

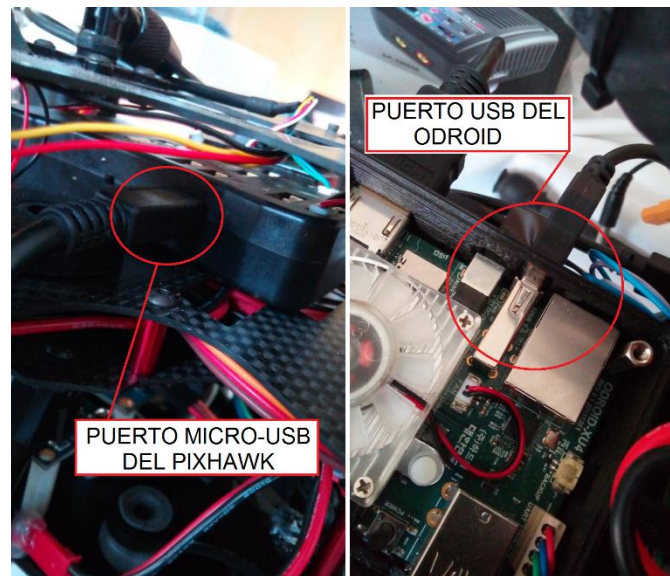


Figura 78: Puertos de comunicación entre Pixhawk y el Odroid XU4

3.3.5. Comunicación estación de tierra - Odroid XU4

La comunicación entre la laptop y el Odroid, se la estableció a través de un router, el cual es alimentado por una fuente de energía externa. Este se conectó por cable LAN con la estación de tierra, y se conectó con el Odroid mediante el puerto Wifi, el cual recibe los comandos y ejecuta los programas a través de la terminal por ssh.

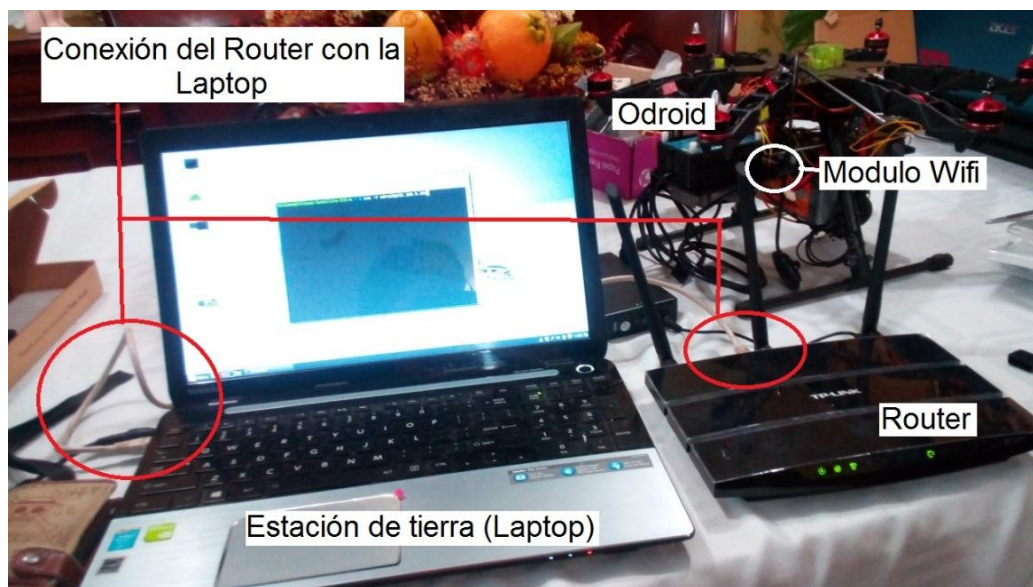


Figura 79: Comunicación Estación de Tierra-Odroid XU4 mediante un router y el módulo Wifi.

3.3.6. Conexión del R9D al SimpleBGC

Se obtuvo el ancho de pulso de la señal de control enviada por el radio control AT9 a través del receptor R9D. Se utilizó el canal 6 del receptor para enviar la señal PWM de control del RC hacia el eje Yaw de la gimbal.



Figura 80: Señal de control del RC hacia el eje Yaw de la gimbal

3.3.6.1. Ancho de pulso de la señal de control

Se midió el ancho de pulso de la señal de control enviada por el radio control AT9, teniendo como resultado $945.0 \mu\text{s}$ en el valor mínimo, el cual hace que la gimbal se mueva en sentido horario a máxima velocidad, y 2.056 ms en el valor máximo, el cual hace que la gimbal se mueva en sentido anti-horario a máxima velocidad, y 1500 ms para mantenerse quieto en la posición en la que se encuentre.



Figura 81: Ancho de pulso del valor mínimo de la señal de control



Figura 82: Ancho de pulso del valor máximo de la señal de control

3.4. Software del sistema de estabilización

En esta subsección se da a conocer el programa y la utilización del mismo para la calibración y configuración del controlador de la gimbal, también el programa y el procedimiento para la instalación del sistema operativo del Odroid XU4.

3.4.1. Controlador de la gimbal

El programa que se utilizó para la calibración del controlador fue el SimpleBGC GUI v2.3b4, el cual es la última versión compatible con el

controlador de gimbal que utilizo, este programa se encuentra disponible en la página oficial de Basecam Electronics. A continuación se indicarán los pasos que se siguió para la ejecución del mismo, así como, la calibración del sensor y el controlador.

3.4.1.1. Ejecución del programa SimpleBGC GUI

Previo a la instalación del programa, se verificó si el sistema operativo Windows reconoce o no el controlador de gimbal al conectarlo a la computadora, depende de la versión de Windows que se este usando el que se instale automáticamente los drivers.

Tanto en Windows 7, como en Windows 8 no reconoció el dispositivo, por lo que se procedió a descargar e instalar el driver CP210x USB to UART Bridge VCP, el cual se encuentra en la página oficial de Silicon Labs.

Después se procedió a ejecutar el archivo SimpleBGC.exe previamente descargado, el cual automáticamente detectó el puerto en el que se encuentra conectado el dispositivo, por lo que se procedió a seleccionarlo y a conectarlo.

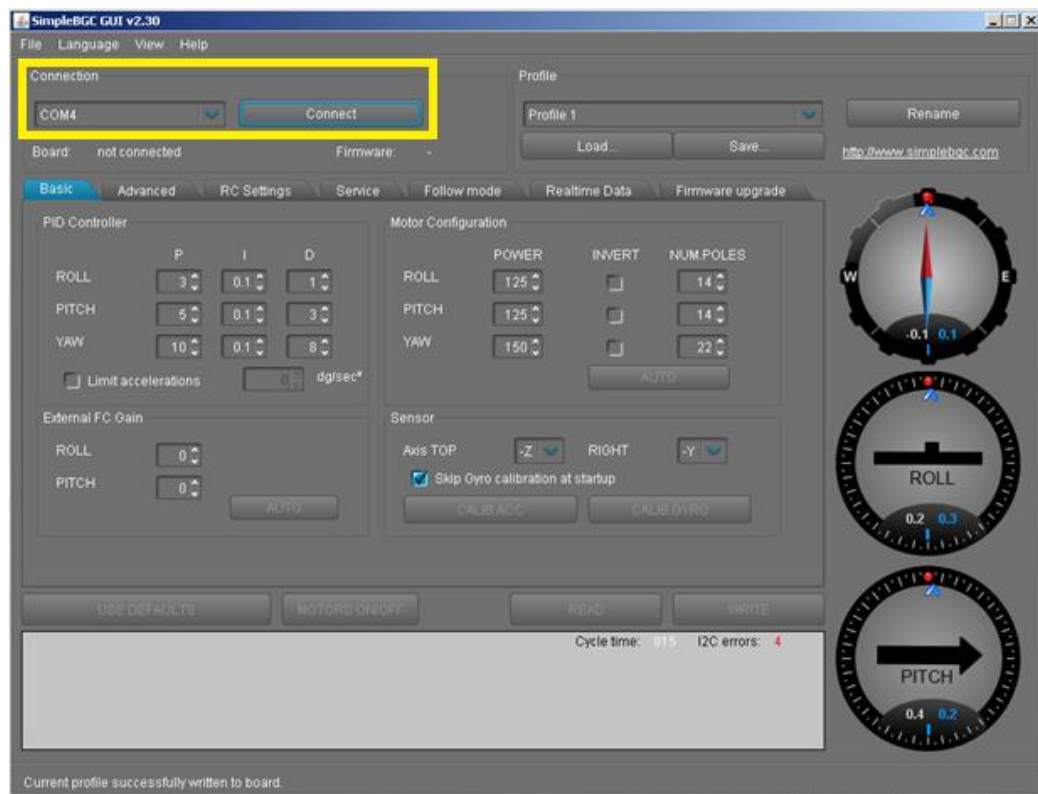


Figura 83: Interfaz del programa SipleBGC GUI v2

Debido a que es la primera vez que se realiza la configuración del controlador es necesario instalar el Firmware, el cual se lo encuentra en la misma página de Basecam Electronics donde se descargó el programa de SimpleBGC en formato .hex, ya que este debe ser compatible con la versión tanto del programa, como del dispositivo. Para la instalación del Firmware utilizamos la pestaña de Actualización de Firmware o Firmware update.

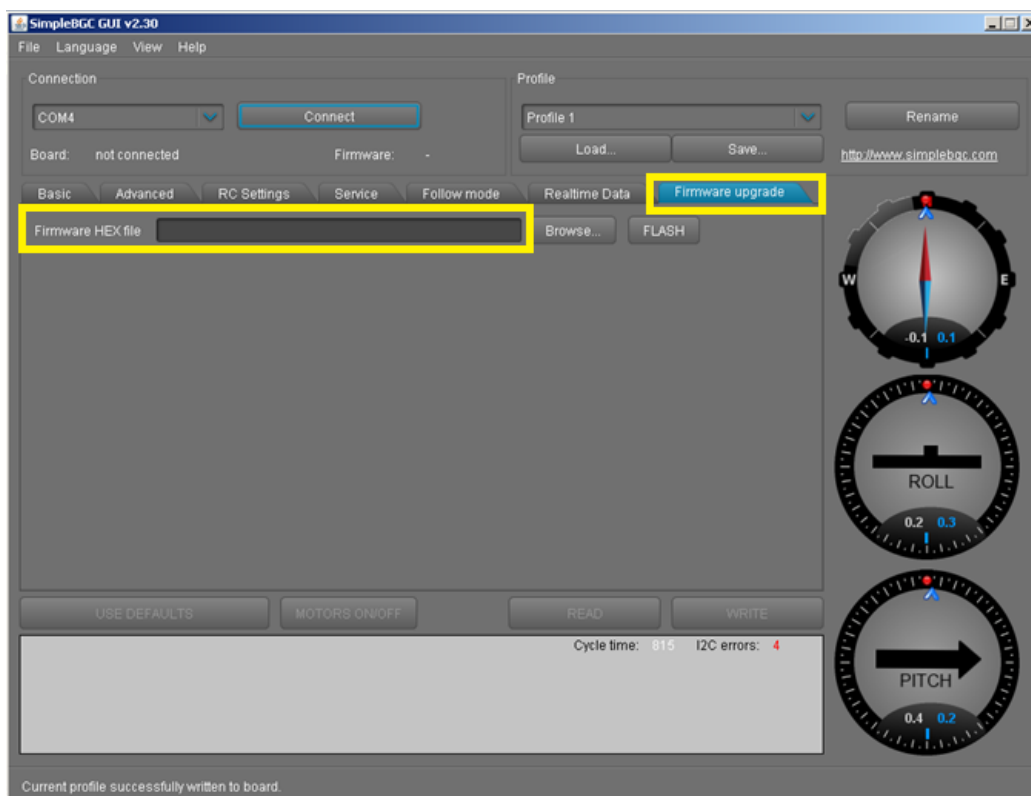


Figura 84: Instalacion del Firmware del controlador de gimbal

3.4.1.2. Calibración del sensor

El siguiente paso es la calibración tanto del acelerómetro, como del giroscopio del sensor. Para la calibración del acelerómetro se colocó el sensor en las posiciones que se indican en la **Figura 53**, presionando el boton CALIB.ACC en cada posición y esperando la indicación del led para realizar el cambio a la siguiente posición. El orden de las posiciones no afecta el resultado, ya que el programa detecta automáticamente.

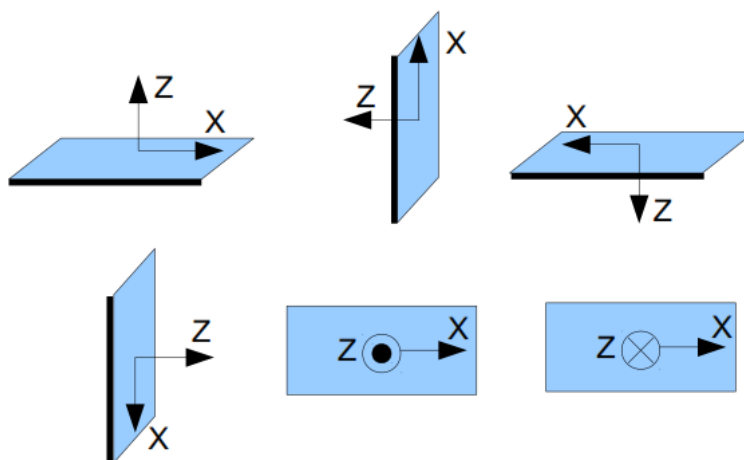


Figura 85: Posiciones del sensor para calibración

Fuente: (Basecam, 2013)

Después de realizar la calibración del acelerómetro, se configuró la posición del los ejes de la IMU, los cuales fueron seleccionados en la sección Sensor del programa, como se muestra en la **Figura 86**.

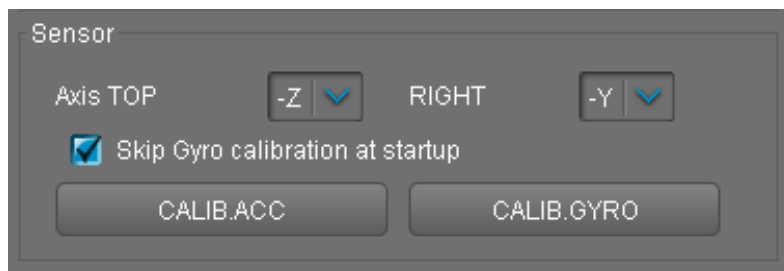


Figura 86: Posición de los ejes de la IMU

La posición de la IMU se verificó mediante los indicadores ubicados en el lado derecho de la pantalla. En estos indicadores tanto los puntos rojos, como las flechas azules deben estar en la posición que se indica en la **Figura 87**.



Figura 87: Indicadores de la posición de la IMU

Esta calibración se la puede realizar ya sea el sensor solo o colocado en la estructura, teniendo en cuenta de que se debe mantener inmóvil al momento de realizar la calibración. Por lo que es recomendable usar un cubo nivelador de 3 ejes, como se muestra en la **Figura 88**.

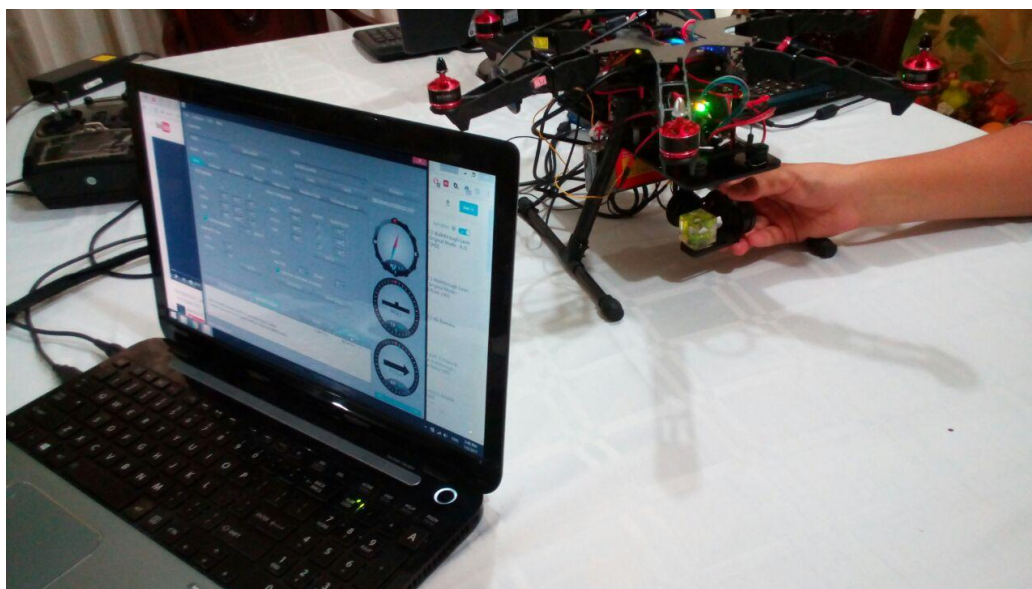


Figura 88: Utilización del cubo nivelador de 3 ejes

Para la calibración del giroscopio la posición de la cámara debe estar como se indica en la **Figura 89**. Si se desea que la cámara tenga una inclinación, el programa permite colocar ángulos de compensación.

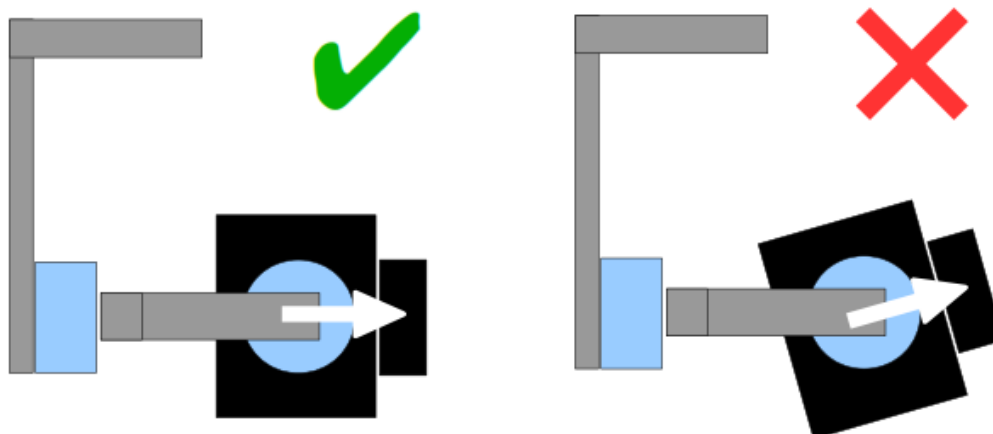


Figura 89: Posición de la cámara para calibración

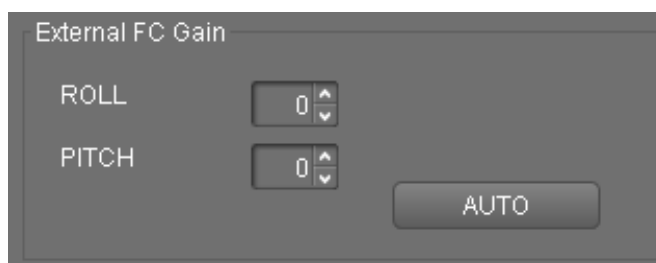


Figura 90: Ángulos de compensación para la cámara

Otra opción que posee el programa para colocar la cámara en una posición específica requerida es la calibración del giroscopio cada vez que se enciende el controlador. Para ello, el programa espera durante 4 segundos, tiempo en el que se posiciona la cámara, y se mantiene en esta posición durante 3 segundos, tiempo en el que el programa establece los parámetros iniciales. En caso de requerir o no esta opción se puede marcar o desmarcar la opción de calibración del giroscopio en el encendido.

3.4.1.2.1. Errores I2C de la IMU

Dependiendo de las dimensiones y de la robustez de la IMU esta puede presentar errores I2C, los cuales se pueden observar en la parte inferior de la pantalla principal del programa, como se muestra en la **Figura 91**.

```
Cycle time: 818 I2C errors: 0
```

Figura 91: Error I2C

Para evitar estos errores se procedió a colocar un núcleo de ferrita en el cable de la IMU, el cual actúa como filtro inductor de alta reactancia, cuya función es oponerse al paso y a la propagación de las corrientes de alta frecuencia que circula por los cables, evitando así que estas corrientes se expandan al exterior como señales ruidosas de radiofrecuencia o ruidos parásitos.



Figura 92: Núcleo de ferrita para en la IMU

3.4.1.3. Habilitación del 3er eje del controlador

El programa detecta automáticamente cuantos ejes posee la placa principal del controlador, al hacer uso de una extensión para el 3er eje (Yaw), este no se encuentra habilitado, debido a esto se habilitó este eje en la pestaña de configuraciones avanzadas.

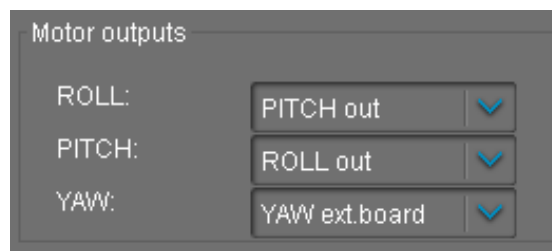


Figura 93: Habilitación del 3er eje del controlador

3.4.1.4. Calibración de los parámetros de control

Se procedió a realizar la calibración de los parámetros de control PID de cada eje, para lo cual se tomó en consideración las siguientes indicaciones:

- El parámetro P describe la potencia de la respuesta a perturbaciones externas, aumentar este valor significa obtener una respuesta más fuerte. Si este valor es demasiado alto, aparecerán oscilaciones en el eje, por lo que se recomienda aumentar el valor del parámetro D en 1 o 2 unidades y luego modificar el valor P nuevamente.
- El parámetro D reduce la velocidad de reacción. Este valor ayuda a eliminar las oscilaciones de baja frecuencia. Un valor demasiado alto puede causar oscilaciones de alta frecuencia.
- El parámetro I cambia la velocidad de reacción de la gimbal a los comandos RC entrantes y la velocidad a la cual regresa a su posición neutral. Valores bajos resultan en una reacción lenta y suave, valores altos aceleran el movimiento.

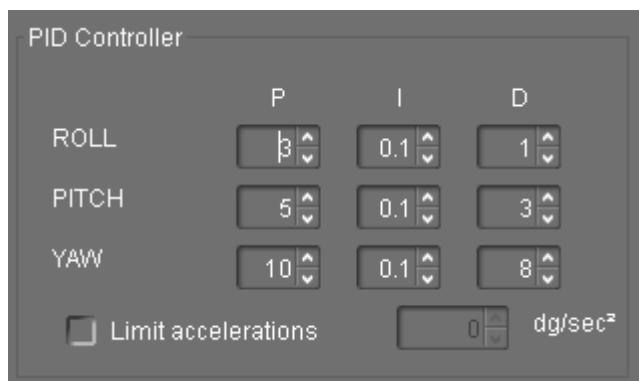


Figura 94: Calibración de parámetros PID del controlador

Se debe tener en cuenta que estos parámetros se encuentran afectados por las vibraciones que pueda tener la IMU, causando una mayor oscilación en cada eje.

3.4.1.5. Configuración de los motores

Se colocó la cantidad de energía que suministrará a cada motor, así como, el número de polos de cada motor, para ello se tomó en cuenta que los valores de voltaje suministrado a los motores varían de 0 – 255, siendo 255 el voltaje total de la batería, el cual se lo escoge de acuerdo a las características de cada motor. Un valor muy bajo no proveerá la energía suficiente para estabilizar la cámara o no se podrá usar en ambientes con mucho viento, y un valor muy elevado generará un desperdicio de energía, haciendo que estos se sobrecalienten.

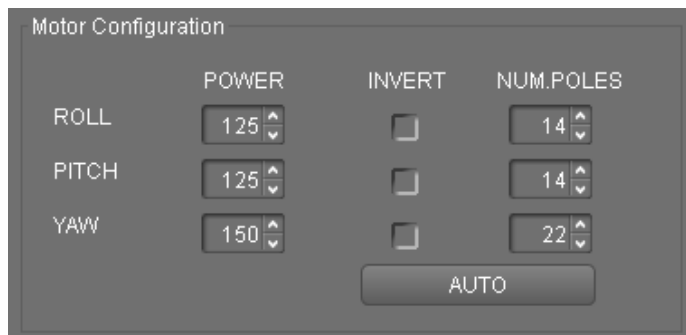


Figura 95: Calibración del voltaje suministrado a los motores

El programa permite enviar las señales de control a bajas y altas frecuencias. El uso de frecuencias bajas evita tener errores I2C, sin embargo, el controlador emite sonidos, lo contrario sucede con el uso de frecuencias altas, ya que evita estos sonidos, pero puede provocar errores I2C. Se eligió el control a altas frecuencias, ya que se adicionó el anillo de ferrita.

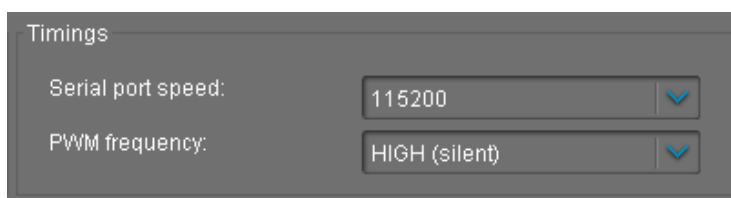


Figura 96: Frecuencia del PWM de los motores

3.4.1.6. Comprobación de funcionamiento

A pesar de que se puede comprobar visualmente el funcionamiento del control que realiza el controlador de gimbal, el programa posee la pestaña de Datos en tiempo real, en la cual se pueden observar las pequeñas vibraciones que existen en la gimbal, las cuales no son perceptibles al ojo humano, sin embargo, se encuentran presentes. De esta manera, se procedió a refinar los valores de los parámetros de control PID para obtener una mejor estabilización.

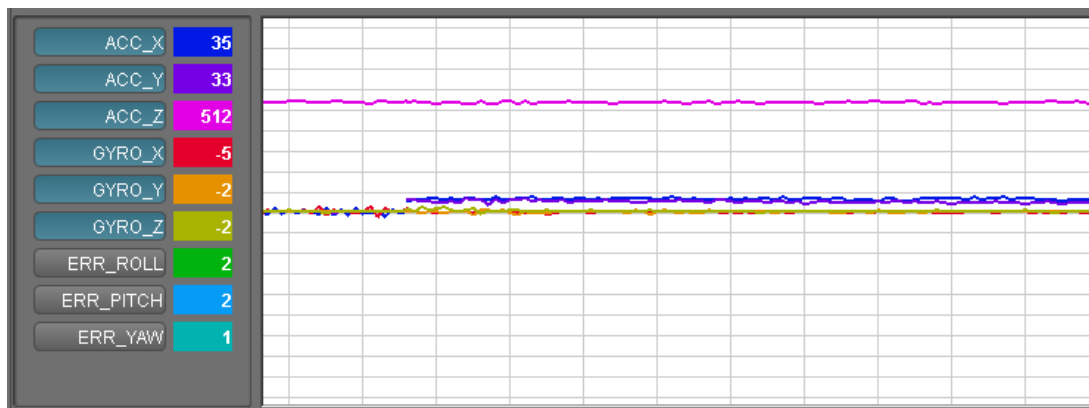


Figura 97: Datos en tiempo real de los acelerómetros y giroscopios

Tabla 2
Parámetros PID y motores para cámara ComQtech

Roll			
		P	Num.
	I	ower	Polos
	(12	14
.1	5		
Pitch			
		P	Num.
	I	ower	Polos
	(12	14
.1	5		
Yaw			
		P	Num.
	I	ower	Polos
	(15	22
0	.1	0	

Tabla 3
Parámetros PID y motores para cámara G-vision Inkfish A1

Roll			
		P	Num.
	I	ower	Polos
	0	13	14
0	.05	5	
Pitch			
		P	Num.
	I	ower	Polos

	0	13	
0	.06	5	14

3.4.2. Sistema Operativo del Odroid XU4

El sistema operativo de fábrica con el que viene la memoria eMMC de la empresa Hardkernel: ODROID, es Ubuntu 15.10 (20160114) (Hardkernel, ODROID WIKI, 2016), el cual no soporta a ROS Jade Turtle, versión que actualmente es el más actualizado y desarrollado por Clearpath Robotics (ROS.org, 2010), requerido para la programación y comunicación de los dispositivos del presente proyecto. Por este motivo, se realizó el cambio del sistema operativo al Odroid XU4, y la instalación del resto de programas, los cuales se indican a continuación.

3.4.2.1. Arranque del sistema operativo

Se descargó y descomprimió la “imagen de recuperación”, el cual es un archivo de arranque de sistema operativo para la Odroid XU4: Android-4.4.4-alpha-sd2emmc_installer-odroidxu3-20151006. (ODROID, 2015)

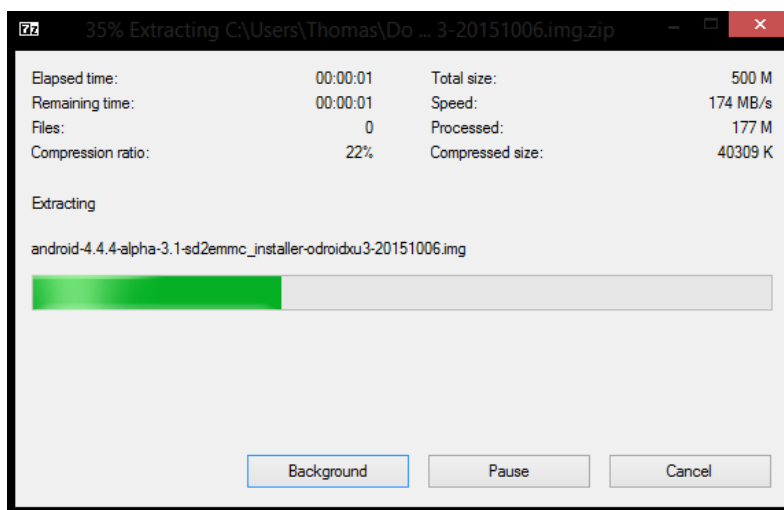


Figura 98: Extracción del archivo de arranque del sistema operativo

Se descargó e instaló el programa Win32DiskImager, el cual ayuda a instalar el archivo ejecutable. Se seleccionó la localidad y el archivo de arranque.

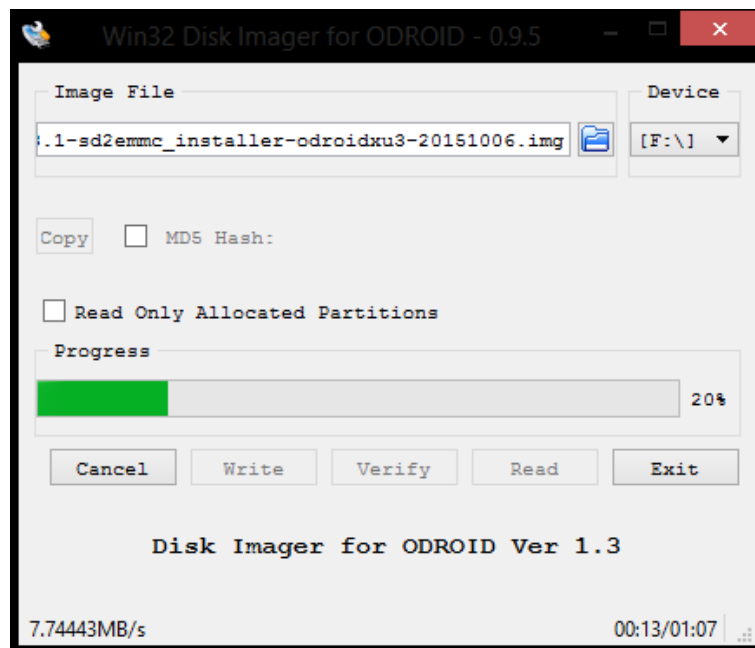


Figura 99: Instalación del archivo de arranque de sistema operativo

Finalizada la instalación, se inserta la memoria microSD y eMMC, dejando la opción de arranque en “ μ SD”.

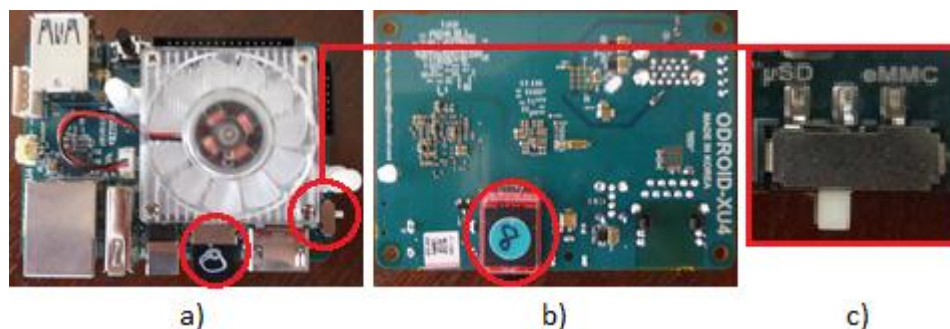


Figura 100: Tarjetas de Memoria del Odroid XU4

Parte Superior de la placa Odroid XU4: Ubicación del microSD y el switch de arranque, b) Parte Inferior de la placa Odroid XU4: Ubicación del eMMC, c) Posición de arranque de la placa en “µSD”

3.4.2.2. Instalación del arranque

Se instaló el arranque del sistema operativo a través de la computadora, mediante la conexión con el puerto USB-UART del Odroid XU4.



Figura 101: Conexión computador-Odroid XU4

Para ello, se instaló el programa de comunicación minicom en la computadora con el comando “sudo apt-get install minicom”, el cual se ejecuta con el comando “minicom -s”.

Se configuró el puerto serial, para lo cual, se seleccionó la opción “Serial port setup”.

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..             |
| Exit                         |
| Exit from Minicom           |
+-----+-----+
```


3.4.2.3. Instalación del sistema operativo

Se descargó el sistema operativo: “ubuntu-15.04-robotics-odroid-xu4-20150911” de la página de fórum de Odroid HardKernel (Hardkernel, 2015) (ODROID, 2016), el cual tiene instalado OpenCV 2.4.12.1, y ROS Jade (Bare Bones).

Se descomprime la descarga, se ejecuta el programa Win32DiskImager, y se elige al archivo del sistema operativo.

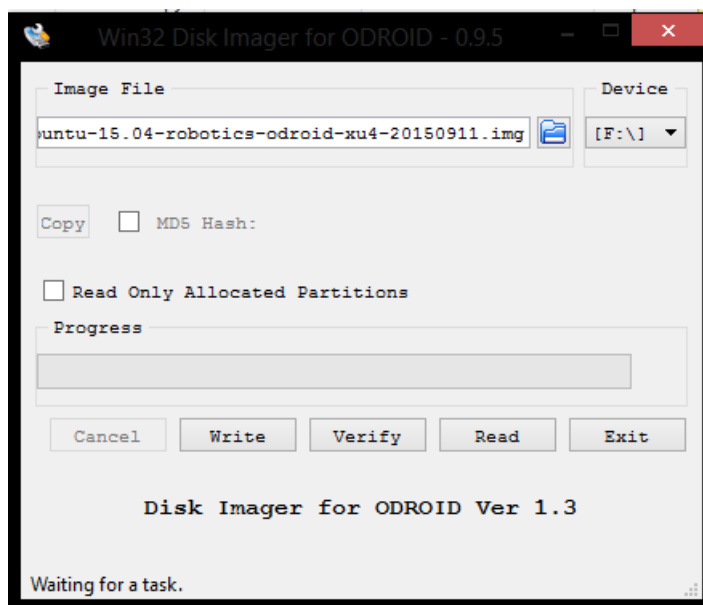


Figura 105: Instalación del sistema operativo para el Odroid XU4.

Se coloca la memoria eMMC al Odroid XU4, y se elige el arranque en la opción eMMC.

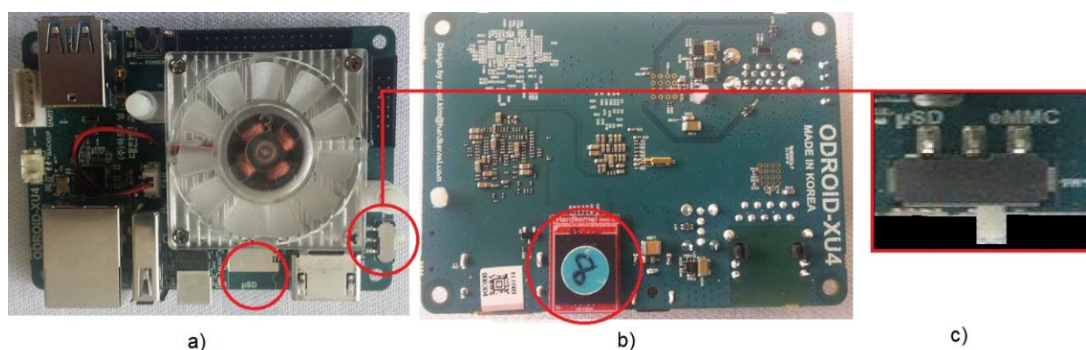


Figura 106: Posición de la eMMC en la placa Odroid XU4

Parte Superior de la placa Odroid XU4: Slot de microSD vacío y el switch de arranque, b) Parte Inferior de la placa Odroid XU4: Ubicación del eMMC, c) Posición de arranque de la placa en “eMMC”

Se energiza la placa, y se espera por varios minutos mientras corre la primera inicialización del sistema. Terminado esto, se desenergiza la placa para colocar los dispositivos (monitor al puerto HDMI, el teclado, el mouse y el adaptador Wifi-USB).

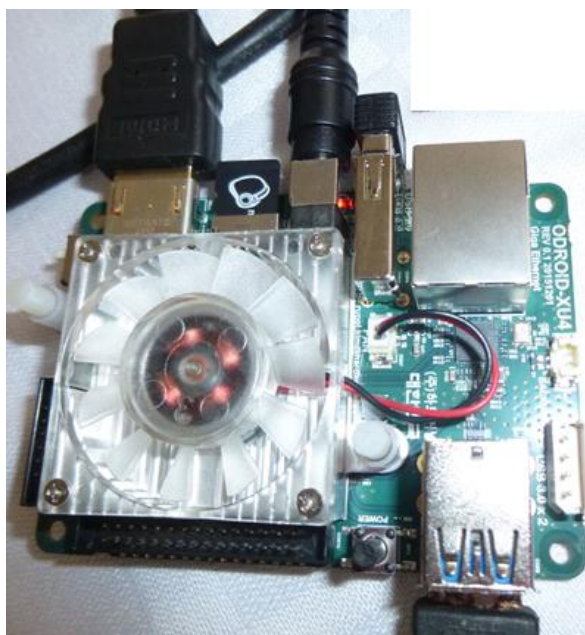


Figura 107: Conexión de los dispositivos al Odroid XU4.

3.5. Algoritmos del sistema de estabilización

En esta subsección, se da a conocer los algoritmos empleados para realizar la obtención de los datos del Pixhawk y los datos de la señal de control, enviada por el radio control AT9, utilizados para sacar la planta del dron. También los algoritmos utilizados para realizar la estimación de intensidad de movimiento y la estabilización de video.

3.5.1. Filtro de Kalman

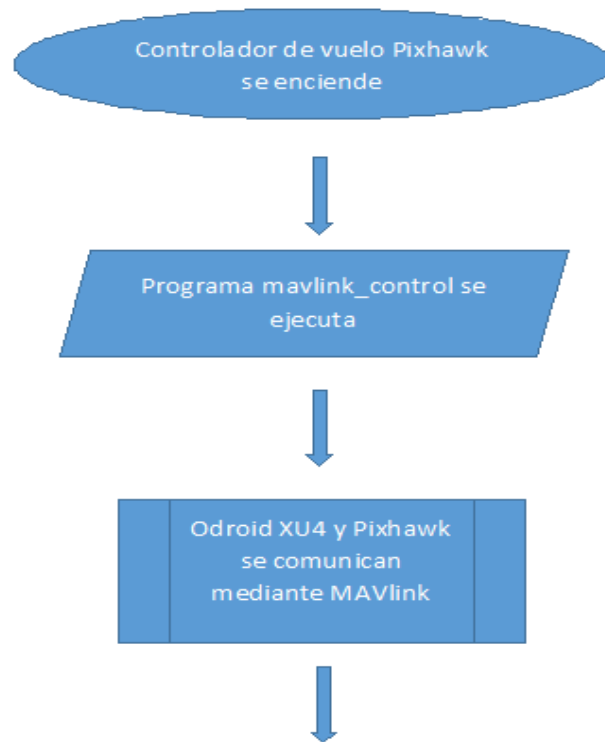
El filtro de Kalman es un método de suavización de movimiento para la estimación de la intención de movimiento. Es una combinación de un filtro pasa bajos de segundo orden, utilizando el último fotograma y la entrada de la acción de control para estimar una confiable intención de movimiento.

3.5.2. Planta del dron

La planta se calculó utilizando los datos de la señal de control, enviados por el radio control AT9, y su respuesta ante ellos, datos de la posición del dron. Para lo cual, se implementó los programas de detección de posición del dron y lectura de la señal de control, los cuales almacenan estos datos.

3.5.2.1. Obtención de datos de la señal de control

El código para la obtención y almacenamiento de los datos de la señal de control está basado en las librerías de MAVlink. Este debe tener acceso al canal que controla el movimiento en Yaw del dron. La lógica de programación se muestra a continuación:



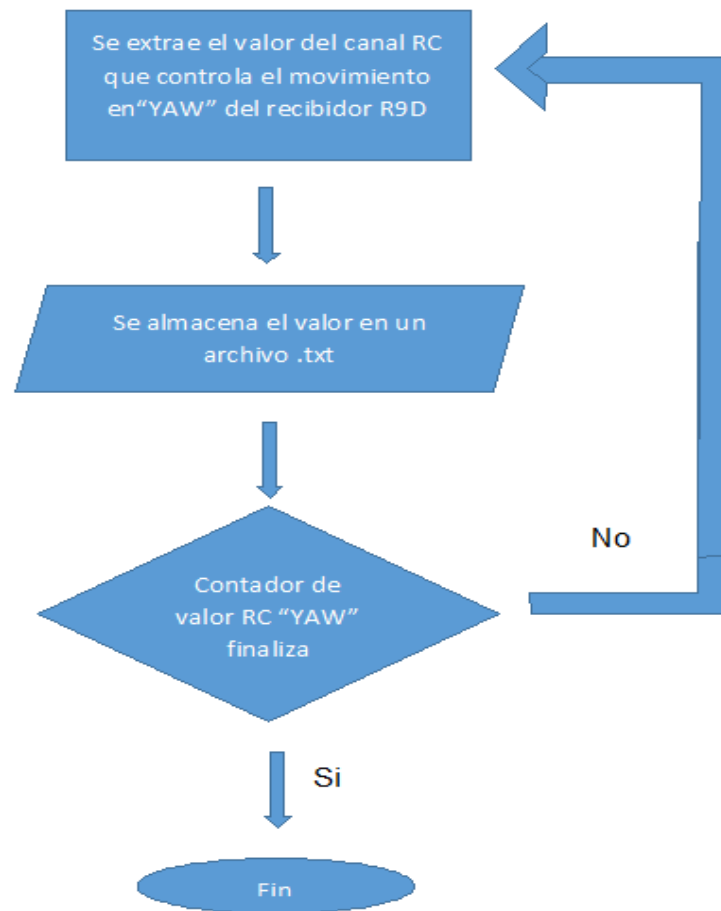
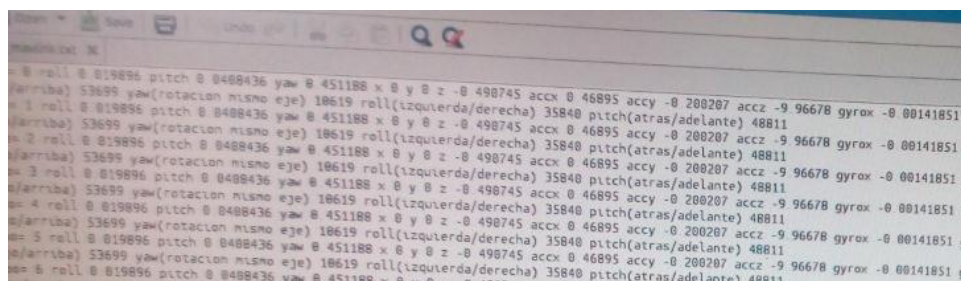


Figura 108: Diagrama de flujo para almacenar la señal de control del RC

El código lee la entrada en el pin GPA0.2 haciendo un salto de periodo por periodo leído, el valor ingresado es evaluado con múltiplos de 10, el cual es almacenado y comparado con el siguiente valor obtenido, para aumentar precisión de lectura de PWM. La salida de PWM es manejada mediante la función `SotfPwm` incluida en la librería `WiringPi`, que permite únicamente valores incrementales en múltiplos de $100\ \mu\text{s}$, inherente a la velocidad de procesamiento del Odroid (Drogon, 2012). Posee un valor mínimo de 1ms y máximo de 2ms de salida de PWM, con un total de 11 opciones de velocidad de giro en la gimbal.

Para la ejecución del programa, a través de la terminal se manda el comando: `"/mavlink_control -d /dev/ttyACM0 -b 921600"`. Donde "mavlink_control" es el programa ejecutable, "-d /dev/ttyACM0" es el puerto de conexión, y "-b 921600" es la velocidad de transferencia de datos.

Los datos son almacenados en un archivo .txt al finalizar el programa.



```

0 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811
1 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811
2 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811
3 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811
4 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811
5 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811
6 roll 0 019896 pitch 0 0400436 yaw 0 451188 x 0 y 0 z -0 490745 accx 0 46895 accy -0 200207 accz -9 96678 gyrox -0 00141851 gy
arriba) 53699 yaw(rotacion mismo eje) 10619 roll(izquierda/derecha) 35840 pitch(atras/adelante) 48811

```

Figura 109: Datos almacenados en el archivo .txt

3.5.2.2. Obtención de datos del Pixhawk

El código que se utilizó para obtener los datos de los sensores integrados en el Pixhawk hacia el Odroid es el "c_uart_interface_example" (LorenzMeier, GitHub, 2016), el cual genera los mensajes necesarios para la comunicación MAVLink integrado en el firmware del Pixhawk. (PixHawk, 2015)

Este código es una interface simple de MAVLink a UART que permite la comunicación entre Pixhawk y una computadora, que envía y recibe un mensaje MAVLink.

Procedimiento para implementación

Se conecta el adaptador Wifi al Odroid XU4, se ingresa a una red conocida y se descarga el programa "c_uart_interface_example" de la página GitHub, y finalmente se instala con el comando "make".

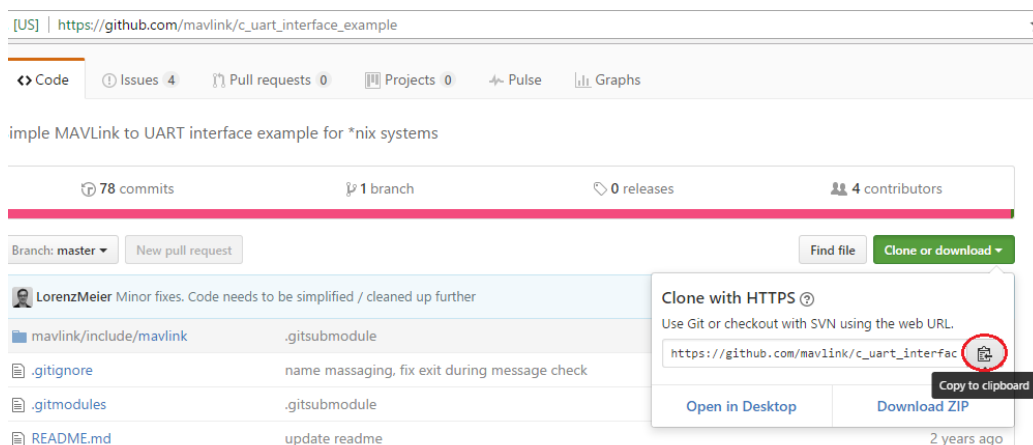
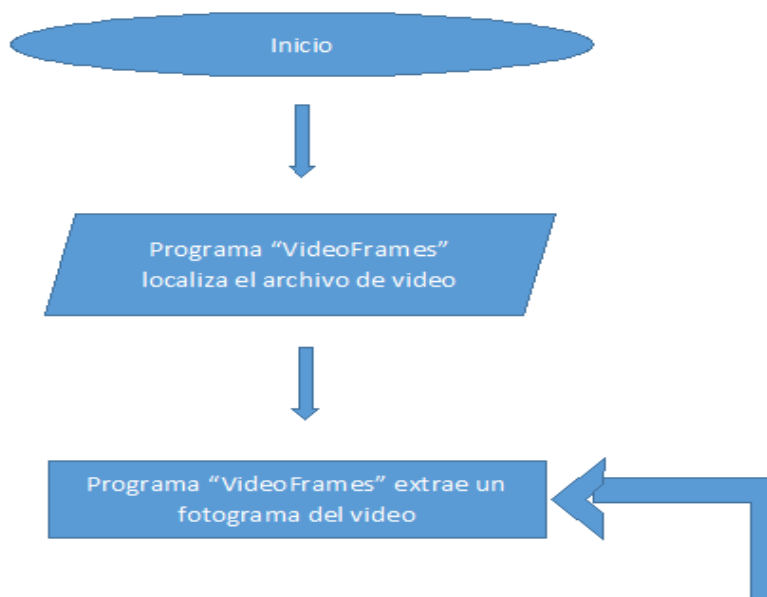


Figura 110: Descarga del programa `c_uart_interface_example`

El programa “`c_uart_interface_example`”, coloca la información en un archivo `.txt`, el cual almacena en 36 segundos, 364.865 vectores de datos, correspondientes a el lapso de tiempo en que se realizó la grabación de video.

Para dividir el video en varias imágenes, se requirió de un archivo de procesamiento desarrollado en MATLAB, de donde se obtuvieron en los 36 segundos de película, 465 fotogramas.



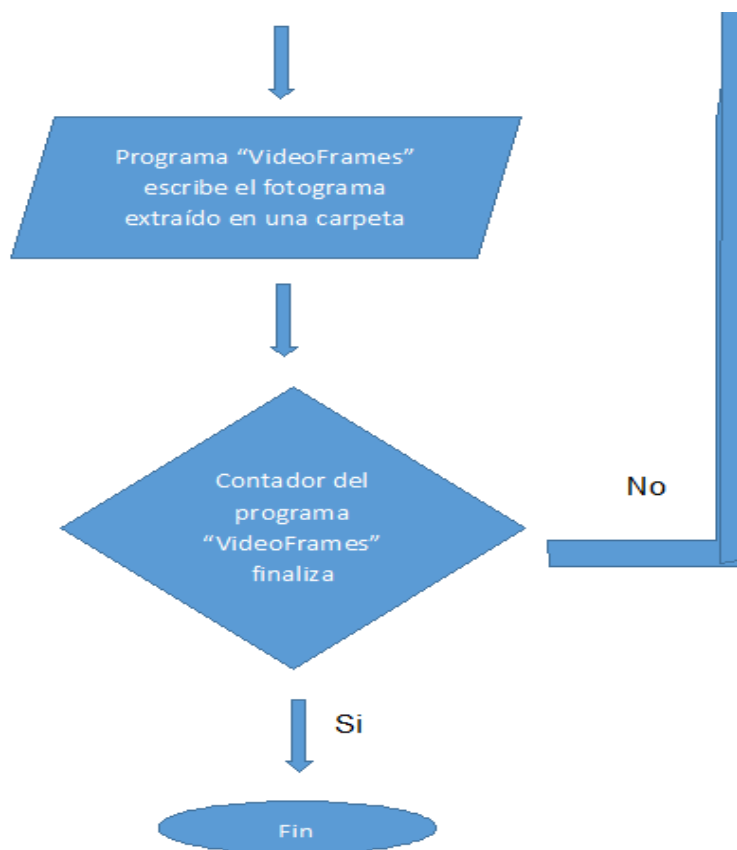


Figura 111: Pseudocódigo para la extracción de fotogramas

Fuente: Prof. Wilbert G. Aguilar, Ph.D.

Al observar que, para la estimación de la planta, se requieren de datos correspondientes entre entradas y salidas, y se tenía un excedente de datos de entrada, se tuvo que disminuir el mismo cambiando el tiempo de la tasa de muestreo y descartando tramos de los datos excedentes en el archivo Excel.

Cálculo de correspondencias entre entrada y salida:

Conociendo que, para 36 segundos, se tienen 364.865 vectores de datos y 465 fotogramas, para el cálculo de la planta se debe tomar un vector de datos:

$$\frac{364865 \text{ muestras}}{465 \text{ fotogramas}}$$

Un dato cada 784 muestras, correspondientes a un fotograma

Descartamos los tramos de la siguiente manera en los datos ubicados en el archivo en Excel:

Utilizamos el siguiente código realizado en VBA, una vez abierto el archivo respectivo.

The screenshot displays the Microsoft Excel interface with a VBA macro editor window open. The macro editor shows the following VBA code:

```

Sub DeleteRowa()
    Dim A As Integer
    Dim B As Integer
    Dim i As Integer
    A = 2
    B = 270
    Rows(A & ":" & B).Delete
    For i = 0 To 382
        A = A + 1
        B = B + 1
        Rows(A & ":" & B).Delete
    Next i
End Sub

```

The background Excel spreadsheet shows a table with the following data:

	A	B	C	D
1	tiempo=	0 roll		0.0031
2	tiempo=	270 roll		0.0034
3	tiempo=	540 roll		0.0034
4	tiempo=	810 roll		0.0038
5	tiempo=	1080 roll		0.0038
6	tiempo=	1350 roll		0.0038
7	tiempo=	1620 roll		0.0038
8	tiempo=	1890 roll		0.0038
9	tiempo=	2160 roll		0.0038
10	tiempo=	2430 roll		0.0040
11	tiempo=	2700 roll		0.0040
12	tiempo=	2970 roll		0.0040
13	tiempo=	3240 roll		0.0040
14	tiempo=	3510 roll		0.004
15	tiempo=	3780 roll		0.004
16	tiempo=	4050 roll		0.004
17	tiempo=	4320 roll		0.0041
18	tiempo=	4590 roll		0.004
19	tiempo=	4860 roll		0.0041
20	tiempo=	5130 roll		0.0042
21	tiempo=	5400 roll		0.0042
22	tiempo=	5670 roll		0.0042
23	tiempo=	5940 roll		0.0042

Figura 112: Selección de muestras correspondientes a los fotogramas.

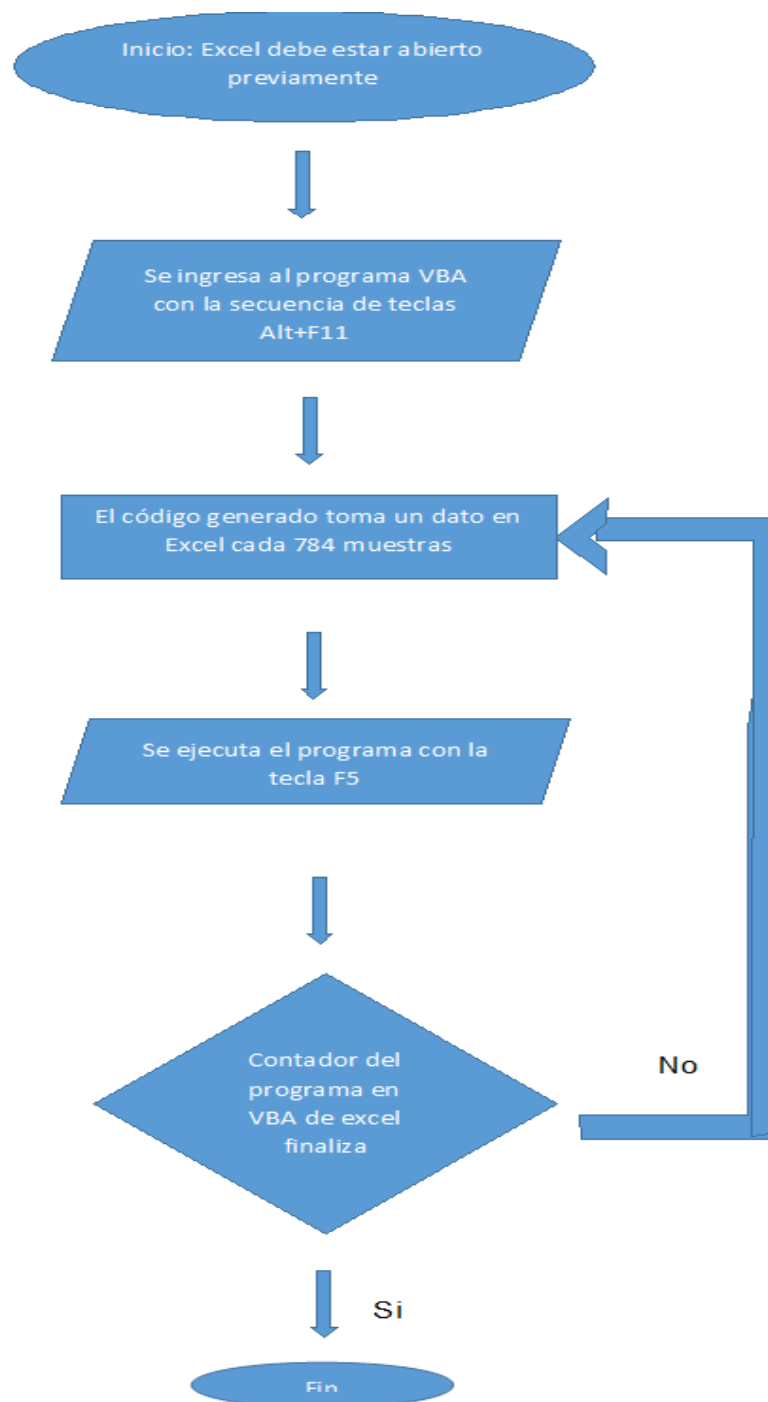


Figura 113: Pseudocódigo en VBA para eliminación de tramos y selección de muestreos correspondientes a los fotogramas.

Una vez obtenidas las muestras, se llega a tener como resultado 465 vectores de datos de entrada para los 465 fotogramas de salida.

Para procesar las imágenes en datos que puedan calcularse para la estimación de la planta, se utilizó el programa desarrollado en MATLAB para transformar los fotogramas en un parámetro de movimiento de traslación en el eje x, en base a la transformación "affine".





Figura 114: Pseudocódigo para la transformación de fotogramas en un parámetro de movimiento de traslación en el eje x.

Fuente: Prof. Wilbert G. Aguilar, Ph.D.

El programa genera una variable llamada “tx”, la cual es una matriz 1x465.

Del archivo en Excel con los datos extraídos del Pixhawk, se copia la columna que contiene los valores de entrada de la acción de control en “Yaw” hacia MATLAB.

Para el cálculo del tiempo de muestreo se realizó lo siguiente: conociendo que hay un cambio de cantidad de fotogramas proporcionados por la cámara para cuando existe movimiento o cuando se encuentra quieto, en 30 segundos se tomaron 434 vectores de datos del Pixhawk, por ende:

$$\frac{30 \text{ segundos}}{434 \text{ muestreos}} = 0.0691244239 \text{ segundos por muestra} \quad \text{Ec.0-5}$$

Una muestra es tomada cada 0.0691244239 segundos, y para cuando la cámara está quieta, se tiene 114 fotogramas en 6 segundos.

$$\frac{6 \text{ segundos}}{114 \text{ fotogramas}} = 0.0526315789 \text{ segundos por muestra} \quad \text{Ec.0-6}$$

Para los 114 fotogramas restantes, el tiempo de muestreo es de 0.0526315789 segundos.

$$\begin{aligned} & \text{Un dato cada } 0.0526315789 \text{ muestras quietas} * 114 \text{ fotogramas} + \\ & \text{Un dato cada } 0.0691244239 \text{ muestras en movimiento} * 434 \text{ fotogramas} = \\ & \qquad \qquad \qquad 36 \text{ segundos} \qquad \qquad \qquad \text{Ec.0-7} \end{aligned}$$

El tiempo de muestreo que se utilizará es de 0.0691244239 segundos en movimiento, debido a que el dron estará en constante movimiento debido a las acciones del control RC, y los efectos de viento.

Con las variables ingresadas, se ejecuta la función “ident” en MATLAB, el cual permite realizar la estimación de la planta.

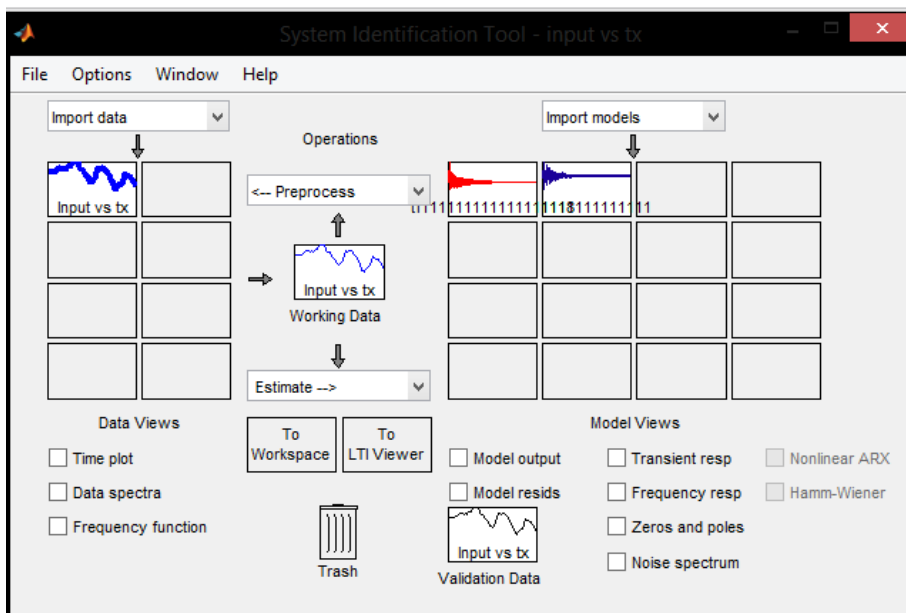


Figura 115: Estimación de la planta.

Se elige la estimación que más se acerque a la curva generada de la relación entre la acción de control en “Yaw”, y el parámetro de movimiento de traslación en el eje x, en este caso una función de transferencia de dos polos y un cero.

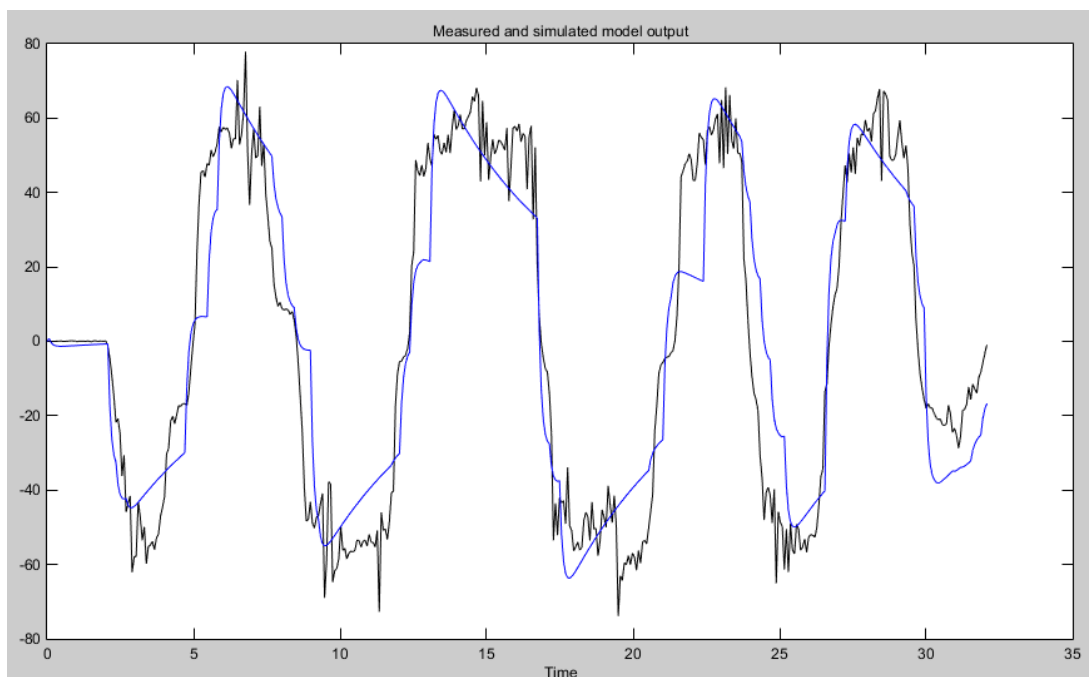


Figura 116: Estimación de la planta basada en la acción de control en “Yaw”, y el parámetro de movimiento de traslación en el eje x. Curva Azul: Modelo de salida simulada. Curva Negra: Modelo medido.

Una vez estimada la función de transferencia más próxima a la planta generada, se procede a modelar los datos a espacios de estado, ingresando los parámetros K_p , T_{p1} , T_{p2} , y T_z , correspondientes a la ecuación:

$$G(s) = K_p * \frac{1+T_z*s}{(1+2*Zeta*Tp1*s+(Tp1*s)^2)*(1+Tp2*s)} \quad \text{Ec.0-8}$$

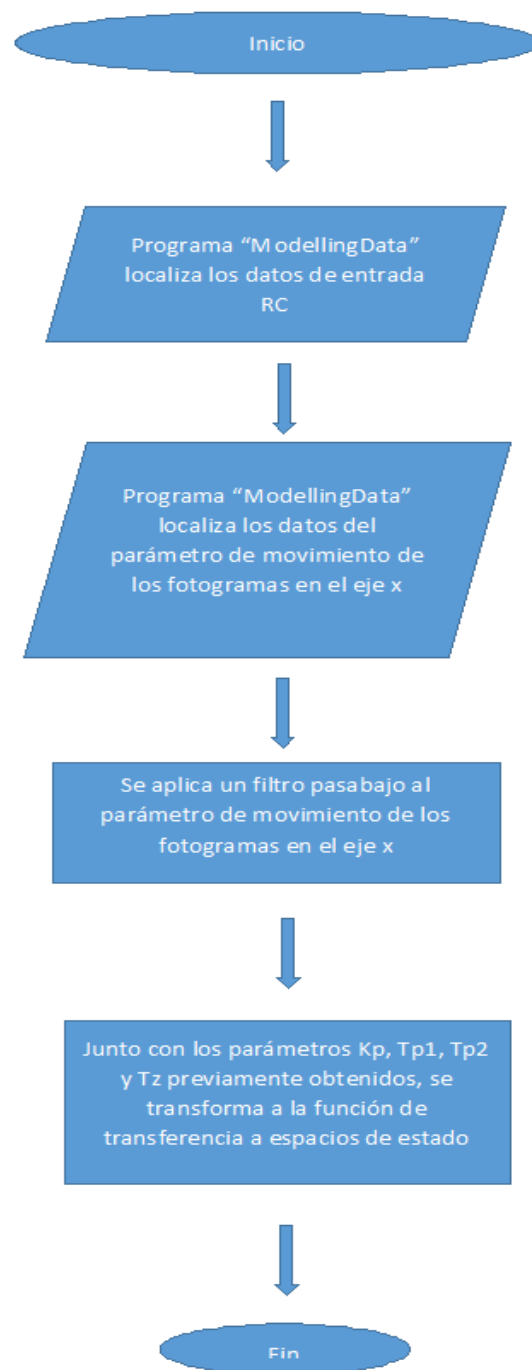


Figura 117: Pseudocódigo para la transformación de la función de transferencia a espacios de estado.

Fuente: Prof. Wilbert G. Aguilar, Ph.D.

Después de poder hallar los espacios de estado, se procede a usar el filtro de Kalman para poder estimar una función que se aproxime mucho mejor a la planta original, y de la cual obtenemos los parámetros de ruido de covarianza del proceso (Q), ruido de covarianza medida (R) y la relación de la variable de entrada RC con la salida del posicionamiento del programa de estabilización mediante el filtro de Kalman aplicado a cada fotograma.

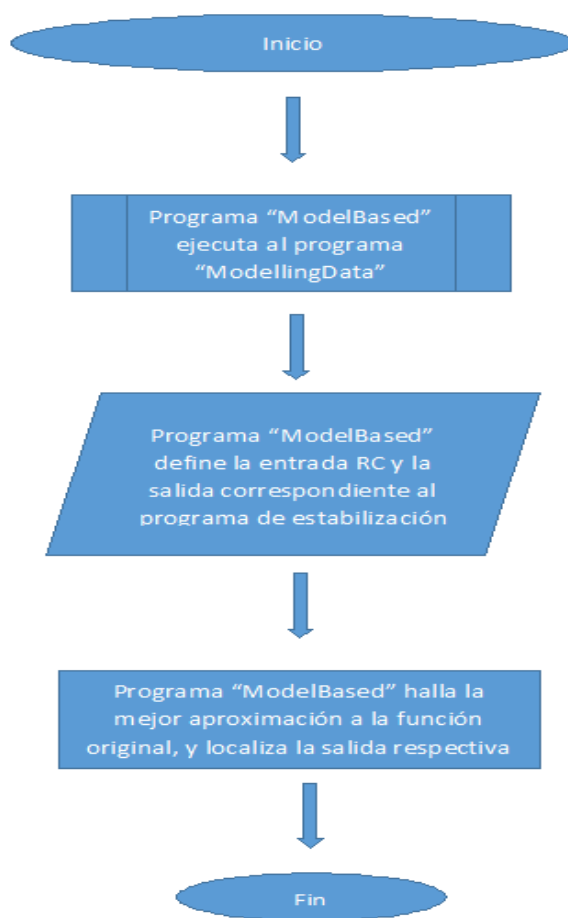


Figura 118: Pseudocódigo para la obtención de la función basada en el filtro

Fuente: Prof. Wilbert G. Aguilar, Ph.D.

Una vez realizada esta operación, se obtiene el siguiente resultado, el cual se implementa en el programa de estabilización de video mediante OpenCV.

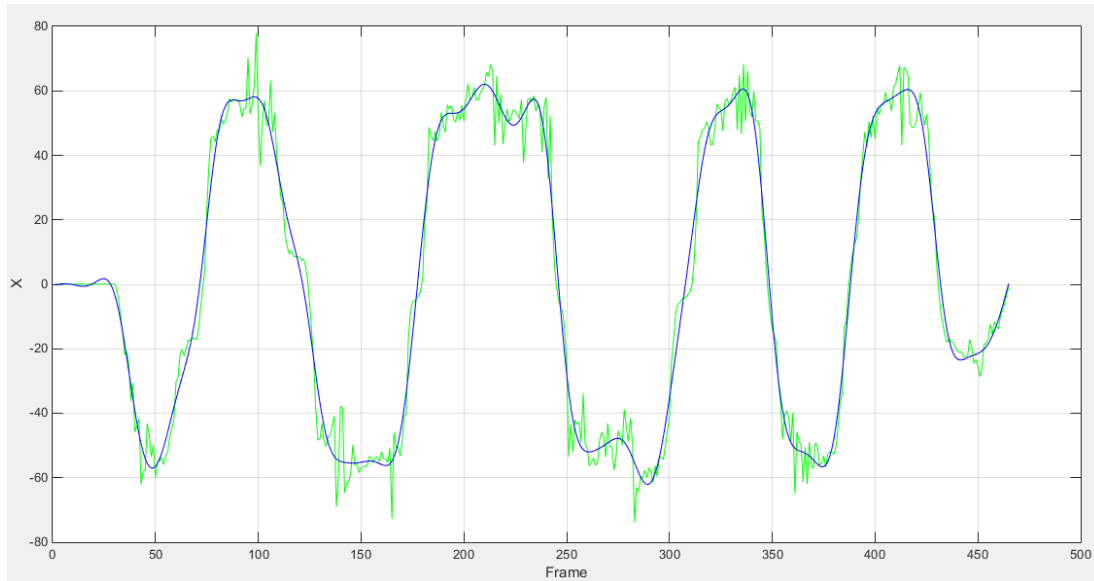


Figura 119: Estimación de la planta basada en el filtro de Kalman a partir del espacio de estado previamente definido.

Curva Azul: Modelo con filtro de Kalman. Curva Verde: Modelo medido.

3.5.3. Algoritmo de estabilización de video

Las imágenes son tomadas directamente de la cámara, y almacenados en el programa como matrices, seguidamente se definen los parámetros de SURF como detector de puntos de interés y BRIEF como descriptor. Pese a que SURF tiene mayor costo computacional que FAST, se lo seleccionó debido a que sus puntos son más fiables. Eso se evidencia durante el proceso de búsqueda de correspondencia en el algoritmo utilizando los dos métodos.

Respecto a los descriptores de puntos de interés se seleccionó BRIEF en lugar de SURF debido a que su coste computacional es considerablemente inferior, con una robustez de los descriptores comparable.

Pese a que se cuenta con un estabilizador mecánico (gimbal) para la cámara a bordo del vehículo, se ha desarrollado un algoritmo de estabilización de video basado en el algoritmo de estabilización de video en tiempo real publicado en (Aguilar & Angulo, 2015)

Este algoritmo se basa en la obtención de la transformación geométrica que representa el movimiento de traslación y rotación desde el fotograma previo hacia el actual, acumulando las variaciones en los parámetros de movimiento entre dos fotogramas consecutivos a lo largo de todo el video. Con base en ello se puede estimar la trayectoria de la cámara con base en las imágenes capturadas. A continuación, se lleva a cabo un proceso de suavizado de la trayectoria que permite estimar el movimiento deseado por el usuario, y compensar únicamente el movimiento indeseado. Como resultado se obtiene un video estabilizado que mantiene los movimientos más significativos de la cámara.

Se requirió de un algoritmo de búsqueda de correspondencias robusto, el cual emplea una prueba de radio que elige puntos de interés obtenidos por el detector que no sean muy próximos unos a otros y detecte los mejores emparejamientos, para facilitar el trabajo del detector de puntos de interés y los resultados no sean redundantes (Abid, 2013). Esta búsqueda de correspondencias simétrica compara las correspondencias actuales con las previas para asegurarse que los mismos sean correctos (ISIS, 2016). Debido a que la estimación de la transformación geométrica depende de la fiabilidad de un correcto emparejamiento de puntos de interés, se utiliza RANSAC, que es ampliamente utilizado como desestimador de falsas correspondencias. RANSAC considera falsas correspondencias a aquellos puntos que estén alejados del modelo. (Choi et al., 2009)

A partir de los puntos de interés emparejados, los parámetros de movimiento pueden ser estimados entre el fotograma actual y el anterior, donde

la variación entre estos puede ser matemáticamente expresada por la transformación geométrica que relaciona los puntos de interés de un fotograma con sus correspondencias en la siguiente.

$$I_t = H_t^{-1} * I_{sp}$$

Donde I_{sp} , I_t son el set de puntos de interés de la imagen de referencia y la imagen no compensada, respectivamente. Y H_t es la matriz de transformación geométrica.

Utilizamos el modelo de traslación, que refiere al movimiento de la imagen cuando el movimiento del dispositivo de captura es de traslación, paralelo al plano de la imagen. Su modelo es el siguiente:

$$H_t = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$H_t = \begin{bmatrix} s * \cos(\emptyset) & -s * \sin(\emptyset) & t_x \\ s * \sin(\emptyset) & s * \cos(\emptyset) & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Donde t_x y t_y son traslaciones, y \emptyset es el ángulo de rotación en “roll”.

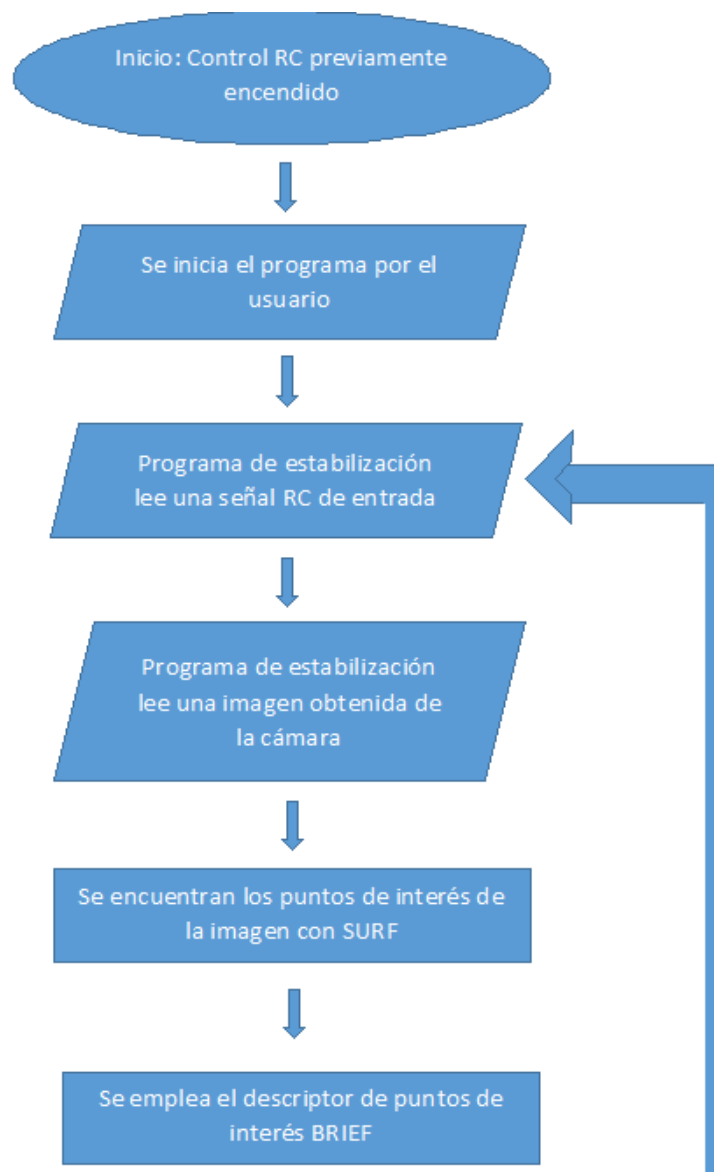
Usamos este modelo, debido a que de su matriz extraemos los valores de las variables t_x , t_y , y \emptyset , las cuales procesamos utilizando el filtro de Kalman previamente implementado, donde comparamos los valores actuales y los anteriores, estimando una confiable intención de movimiento, para posteriormente utilizarlo en la transformación de modelo “Affine”.

El modelo “Affine” estima cuatro parámetros, dos desplazamientos en el plano paralelo a la imagen, como en el modelo de traslación, rotación en “roll”, y el escalado s , que es proporcional al movimiento en la orientación del eje “roll”. (Aguilar & Angulo, 2014a)

Finalmente, se almacena la imagen sin procesar, y la imagen procesada para poder emplearlos en el método de comprobación de resultados ITF.

3.5.4. Implementación del algoritmo

Para acceder a las funciones de la librería de WiringPi, se enlazaron las librerías C y C++ para tener acceso a las GPIO del Shifter Shield, de las cuales se definieron un rango de valores de PWM de salida, a partir de las señales medidas entre el control RC y el R9D en un osciloscopio, y el empleo mediante programación de la función “softPwm” que WiringPi proporciona en el programa de estabilización de video mediante OpenCV.



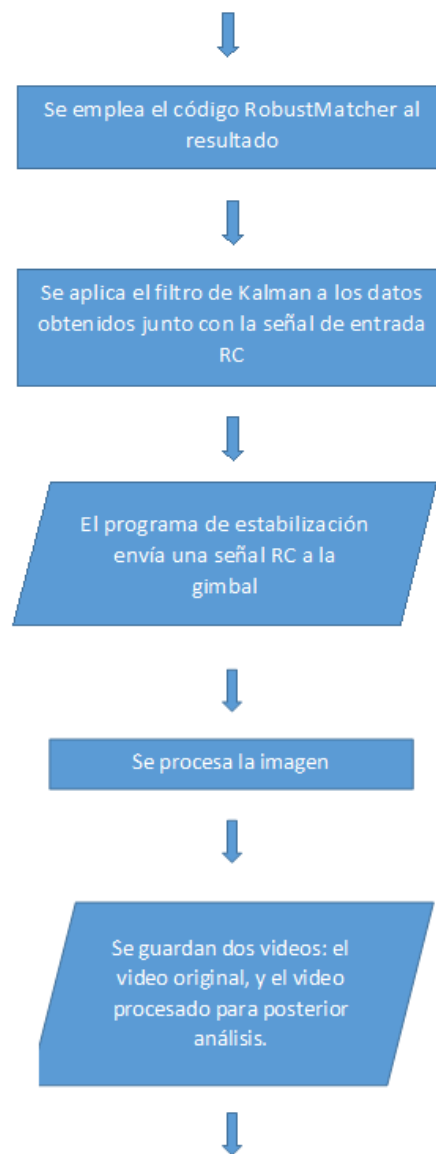




Figura 120: Pseudocódigo del programa de estabilización de video mediante OpenCV.

3.6. Resumen

- El capítulo se centró en: la selección de un Autopilot y una estación de control para el mini RUAV, los componentes adecuados para el sistema de estabilización, el diseño de la base total donde se colocarán los mismos y la implementación de los diferentes algoritmos para determinar la planta del dron, la estimación de intención de movimiento y la estabilización de video.
- Se establecieron los requerimientos que debe poseer el Autopilot, para proveer un vuelo estable y una buena maniobrabilidad del mini RUAV, y de la estación de control, para que permita la adquisición de sus datos.
- Se definieron los requerimientos mínimos de cada componente del sistema de estabilización, de acuerdo a las limitaciones del mini RAUV, disponibilidad del mercado, y precios accesibles.
- Para definir el diseño del sistema de estabilización se consideró que este no afecte la maniobrabilidad y la estabilidad del mini RUAV.

- Se realizó las conexiones del sistema de estabilización mecánico sin que este afecte su funcionamiento ni el de los otros componentes del sistema de estabilización.
- Se establecieron los parámetros de control de controlador de la gimbal con base en las capacidades de cada motor.
- Empleamos el filtro de Kalman, SURF y BRIEF para el sistema de estabilización por software.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

Este capítulo presenta los resultados de las pruebas realizadas al sistema de estabilización. Se evalúan las dimensiones más importantes de la estructura, tiempo de funcionamiento del sistema, estaciones de control, y el algoritmo de estabilización, para verificar el método de evaluación ITF del video resultante del procesamiento.

4.1. Pruebas mecánicas

4.1.1. Peso máximo soportado por el dron

La carga máxima que soporta un dron varía dependiendo de la velocidad del viento y de la capacidad y voltaje de la batería. Las pruebas realizadas para establecer una carga máxima promedio, se las realizo cerca de los hangares de la Universidad de las Fuerzas Armadas – ESPE, con el fin de minimizar las perturbaciones causadas por el viento. Por lo que, se obtuvieron los siguientes datos:

Tabla 4
Carga máxima soportada por el dron

Peso [Kg]	Altitud [cm]
1.5	<6
1.4	<20
1.3	<70
1.1	>150



Figura 121: Pruebas de carga del dron

Se comprobó que el dron soporta una carga promedio máxima de 1.325Kg, sin embargo, no es recomendable llegar a este límite, ya que el tiempo de vuelo del dron disminuye considerablemente, debido a que, los motores consumen una mayor cantidad de energía. Además, no se posee un vuelo estable, lo que podría ocasionar colisiones.

4.1.2. Prueba dimensional de las placas base

En esta prueba se toma en cuenta las piezas que se mandaron a elaborar en acrílico, ya que las dimensiones de estas piezas, así como, las ubicaciones de los agujeros que estas poseen son cruciales, debido al espacio reducido que se tiene para la colocación de los dispositivos.

4.1.2.1. Placa base del motor

La placa base del motor sirve de unión entre la gimbal y la placa base total. La cual se encuentra sujeta mediante tornillos al motor del eje Yaw, y mediante amortiguadores a la placa base total.

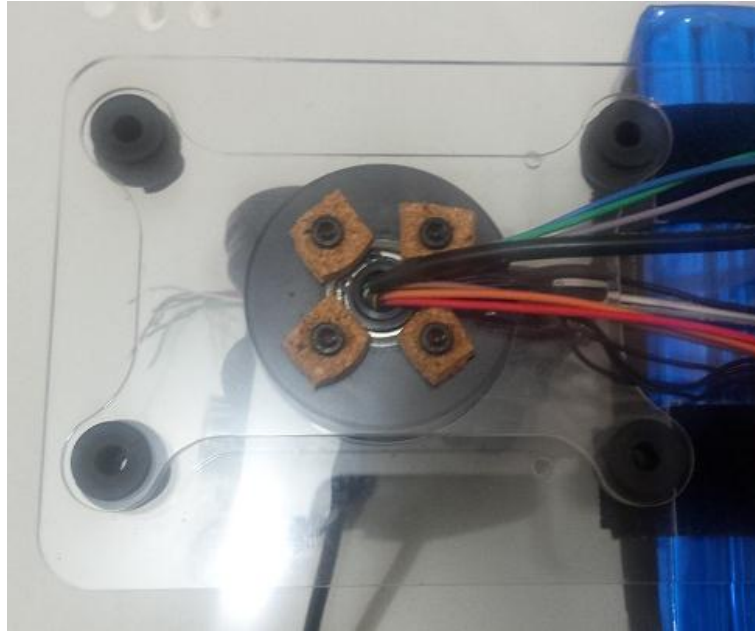


Figura 122: Unión de la gimbal con la placa base total

Cabe mencionar que los agujeros, que se encuentran en esta base deben estar en concordancia con los agujeros tanto del motor, como de la placa base, ya que por este medio estarán sujetos estos elementos. Debido a que el corte en acrílico posee una tolerancia de $\pm 0.2\text{mm}$, las dimensiones colocadas en el diseño CAD de esta placa no se vieron afectadas, por lo que se aceptan estos resultados como positivos.

4.1.2.2. Placa base total

La placa base total sirve de soporte para los dispositivos utilizados en el sistema, también permite unir este sistema al dron mediante separadores con sus respectivos tornillos.

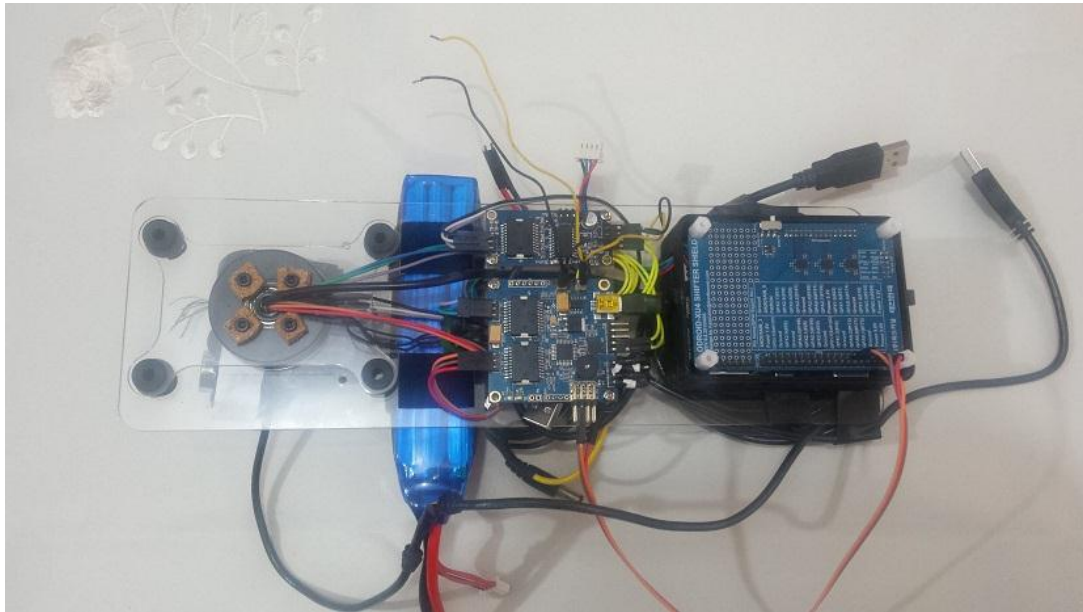


Figura 123: Disposición de los dispositivos en la placa base total



Figura 124: Unión del sistema de estabilización de video al dron

Cabe mencionar que los agujeros, que se encuentran en esta base total deben estar en concordancia con los agujeros de los dispositivos del sistema, ya que por este medio estarán sujetos estos elementos. Debido a que el corte

en acrílico posee una tolerancia de $\pm 0.2\text{mm}$, las dimensiones colocadas en el diseño CAD de esta placa no se vieron afectadas, por lo que se aceptan estos resultados como positivos.

4.1.3. Prueba de compresión de los amortiguadores

Se comprobó que los amortiguadores soportan una carga mayor a la del peso de la gimbal, para ello se sometió a la carga, la cual posee un peso de 409.1g.



Figura 125: Peso de la batería



Figura 126: Deformación de los amortiguadores cuando se somete a una carga

Como se puede observar en las imágenes anteriores, cuando se somete a una carga de 409.1g, la deformación que se produce en los amortiguadores es casi imperceptible, además, soporta vibraciones sin sufrir una deformación permanente.

El software que se utiliza para el control del autopilot y su correcta instalación es variada y dependiendo del uso que se le dé se lo toma en cuenta para su aplicación y es de lo cual se enuncia a continuación.

4.2. Comparación de software

4.2.1. Mission Planner

Al utilizar el programa Mission Planner, comprobamos que posee una interfaz más amigable en cuanto a configuración y calibración del Autopilot. Además, realiza una mejor calibración de los sensores internos del Pixhawk, lo que permite realizar un vuelo más estable.



Figura 127: Calibración de los sensores del Pixhawk a través de Mission Planner

Se comprobó que Mission Planner a través de Dr. Ming permite extraer sus datos, sin embargo, no permite enlazar con códigos de programación en C, entre otros. Por lo que se dificultó la extracción de estos datos, los cuales son necesarios para el cálculo de la planta del dron.

4.2.2. QGroundControl

Al utilizar el programa QGroundControl, comprobamos que este no posee una interfaz tan amigable como la de Mission Planner. La configuración y calibración de los sensores del Autopilot no es tan robusta, ya que se comprobó que no se posee un vuelo estable sin la utilización de sensores adicionales.



Figura 128: Calibración de los sensores del Pixhawk

a través de QGroundControl

A pesar de esto, se comprobó que QGroundControl permite acceder a las variables mediante el programa de `c_uart_interface_example` desarrollado por Lorenz Meiyer, el cual permite una programación en código, con una salida de datos en archivos `.txt`, sin embargo, se posee un tiempo de retardo de envío de datos de 3 segundos.

```

mavlink_msg_heartbeat [-/c_uart_interface_example/mavlink/include/mavlink/v1.0/command] - pluma
odroid@odroid:~/c_uart_interface_example$ cd c_uart_interface_example
odroid@odroid:~/c_uart_interface_example$ ./mavlink_control -d /dev/ttyACM0 -b 921600
OPEN PORT
Connected to /dev/ttyACM0 with 921600 baud, 8 data bits, no parity, 1 stop bit (
8N1)

START READ THREAD

CHECK FOR MESSAGES
Found

GOT VEHICLE SYSTEM ID: 1
GOT AUTOPILOT COMPONENT ID: 1

```

Figura 129: Programa `c_uart_interface_example`

Tomando en cuenta todos los dispositivos conectados en el sistema y la disponibilidad de energía, se realizó la siguiente consideración.

4.3. Tiempo de funcionamiento del sistema

Para determinar el tiempo de funcionamiento del sistema, es necesario conocer los parámetros que definen a la batería que se utilizó, la cual era de 11.1V 5500mAh 30C.

- Capacidad (mAh)

Es la cantidad de energía que puede soportar la batería. Es una forma indicar cuánta carga se puede colocar en la batería durante 1 hora, momento en el que la batería se descargará por completo.

- Velocidad de descarga (C)

Es la velocidad con la que una batería puede descargarse con seguridad. Una batería con una capacidad de descarga de XC, se descarga a una velocidad X veces mayor que la capacidad del paquete.

La batería indica que soporta una carga de 5500 mA durante 1 hora, y que puede ser descargada 30 veces más veloz, es decir, soporta una carga de 165000 mA durante 2 minutos. El tiempo de duración se lo calculó dividiendo la capacidad de la batería (mAh) para la carga utilizada (mA).

El valor de la corriente para los motores del dron se calculó utilizando el tiempo promedio de vuelo y la capacidad de la batería, el valor de las corrientes de los diferentes componentes electrónicos se obtuvo a través de las páginas web respectivas de estos dispositivos, por último, la corriente de los motores de la gimbal se obtuvo mediante mediciones realizadas durante sus pruebas de funcionamiento. Sin embargo, estas mediciones no contemplan al sistema trabajando con la carga máxima soportada tanto del dron, como de los motores de la gimbal, y sin perturbaciones externas.

Cuadro 15

Cargas del sistema

Carga	mA
Motores del dron	21600
Pixhawk	400
Controlador de gimbal	350
Odroid XU4	4000
Motor G-Yaw	2000

Continúa



Motor G-Pitch	1000
Motor G-Roll	1000
Total	30350

Tabla 5

Calculo del tiempo de funcionamiento del sistema

Capacidad (mAh)	Carga utilizada (mA)	Tiempo de funcionamiento (min)
5500	30350	10.8731466

La batería utilizada de 11.1V 5500mAh 30C proporcionó un tiempo de vuelo entre 10-11 minutos, los cuales variaron, debido a que también depende del modo de vuelo, la velocidad del viento y el peso total del dron.

Para analizar el correcto funcionamiento del sistema de estabilización, se realizaron varias pruebas de video y se analizaron con el método ITF que se detalla en el siguiente tema.

4.4. Pruebas del Algoritmo

A continuación, se presentan fotogramas de los videos resultantes de las pruebas del sistema de estabilización que minimiza los movimientos fantasma.



Figura 130: V6 frame 6 y 7.

Arriba: Video original. Abajo: Video estabilizado



Figura 131: V6 frame 7 y 8.

Arriba: Video original. Abajo: Video estabilizado



Figura 132: V3 frame 11 y 12.

Arriba: Video original. Abajo: Video estabilizado

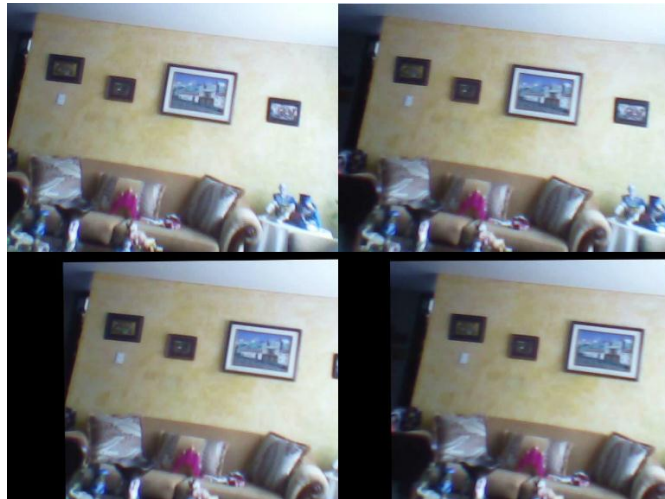


Figura 133: V6 frame 47 y 48.

Arriba: Video original. Abajo: Video estabilizado



Figura 134: V6 frame 48 y 49.

Arriba: Video original. Abajo: Video estabilizado



Figura 135: V4 frame 54 y 55.

Arriba: Video original. Abajo: Video estabilizado



Figura 136: V6 frame 81 y 82.

Arriba: Video original. Abajo: Video estabilizado



Figura 137: V6 frame 82 y 83.

Arriba: Video original. Abajo: Video estabilizado

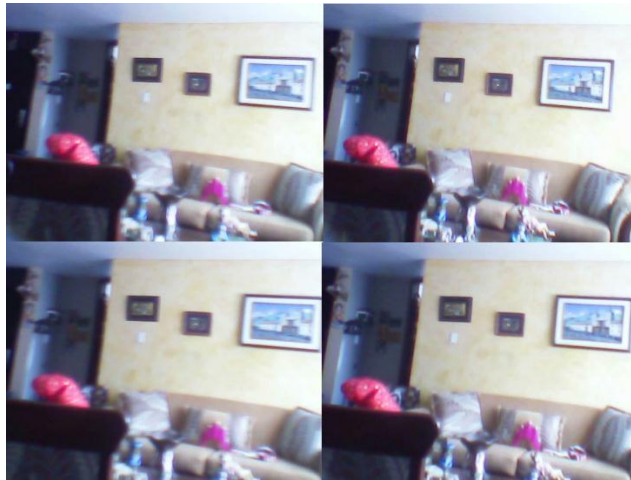


Figura 138: V5 frame 84 y 85.

Arriba: Video original. Abajo: Video estabilizado



Figura 139: V6 frame 103 y 104.

Arriba: Video original. Abajo: Video estabilizado



Figura 140: V6 frame 104 y 105.

Arriba: Video original. Abajo: Video estabilizado



Figura 141: V2 frame 3 y 4.

Arriba: Video original. Abajo: Video estabilizado



Figura 142: Video de cámara estática sin estabilizar, frame 119 y 120.

Arriba: Video original. Abajo: Video estabilizado



Figura 143: Video de cámara con gimbal sin estabilizar, frame 133 y 134.

Arriba: Video original. Abajo: Video estabilizado



Figura 144: Video offline del algoritmo de estabilización, frame 133 y 134.

Arriba: Video original. Abajo: Video estabilizado

Para esta prueba se realizaron varios vuelos, de los cuales se extrajeron los fotogramas con mejores resultados, los cuales evidencian el funcionamiento del sistema y han sido procesados mediante la métrica de evaluación que determina el desempeño del algoritmo, que es el ITF. Para obtener el ITF, primero se requiere obtener el RMSE y posterior a este, el PSNR.

4.4.1. Obtención del RMSE

El RMSE evalúa la confiabilidad del movimiento estimado y el movimiento observado. Un RMSE bajo significa que el movimiento estimado es similar a la intención de movimiento real.

$$RMSE(k) = \left(\frac{1}{M*N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||Frame_k(i,j) - Frame_{k-1}(i,j)||^2 \right)^{\frac{1}{2}}$$

Tabla 6

RMSE por fotograma

<i>Nombre de Muestra</i>	<i>Original</i>	<i>Procesado</i>
V6 frame 6 y 7	74.82	52.83
V6 frame 7 y 8	53.83	27.06
V3 frame 11 y 12	24.56	18.61

Continúa



V6 frame 47 y 48	44.06	35.75
V6 frame 48 y 49	36.96	26.96
V4 frame 54 y 55	1024.26	372.71
V6 frame 81 y 82	45.16	42.58
V6 frame 82 y 83	40.66	27.00
V5 frame 84 y 85	1192.46	202.44
V6 frame 103 y 104	40.95	35.33
V6 frame 104 y 105	32.91	27.63
V2 frame 3 y 4	316.37	325.09
Video de cámara estática sin estabilizar, frame 119 y 120	1730.62	655.95
Video de cámara con gimbal sin estabilizar, frame 133 y 134	1452.57	700.83
Video offline del algoritmo de estabilización, frame 1222 y 1223	167.19	113.99

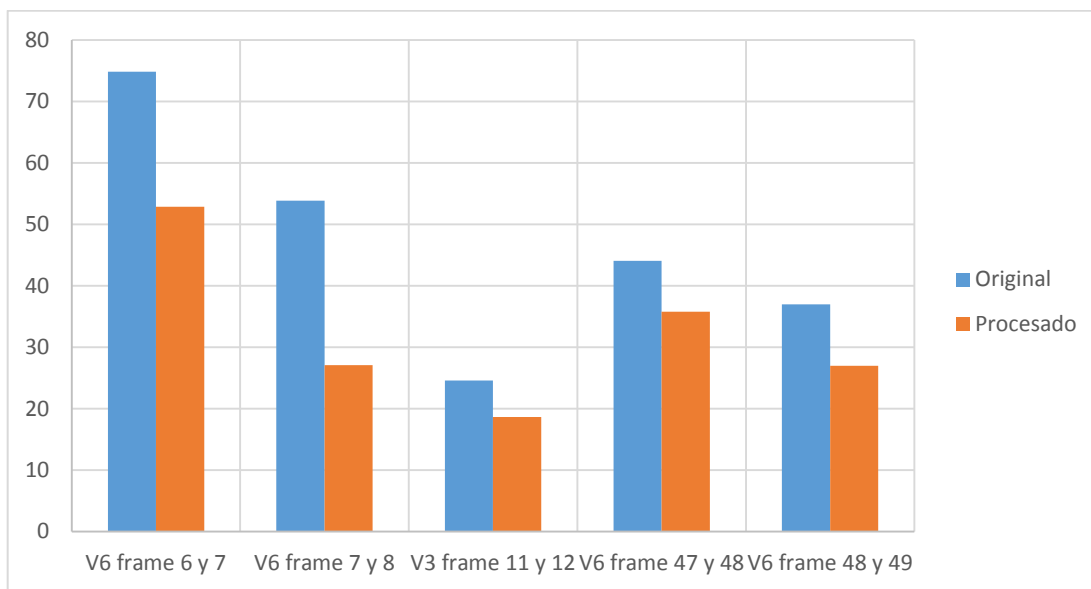


Figura 145: Comparación del RMSE de los fotogramas originales y los fotogramas procesados.

Tabla 7

RMSE total y Porcentaje de mejora

	<i>RMSE sin estabilización</i>	<i>RMSE con estabilización</i>	<i>Porcentaje de mejora (%)</i>
Video1	57.20	76.85	-25.56
Video2	239.77	299.86	-20.04
Video3	453.27	702.36	-35.46
Video4	453.96	612.39	-25.86
Video5	514.43	618.20	-16.78
Video6	638.20	970.22	-34.22
Cámara estática	723.66	871.79	-16.99

Continúa



Cámara			
con	1002.36	1290.37	-22.31
gimbal			
Prueba	159.65	154.29	3.35
Offline			

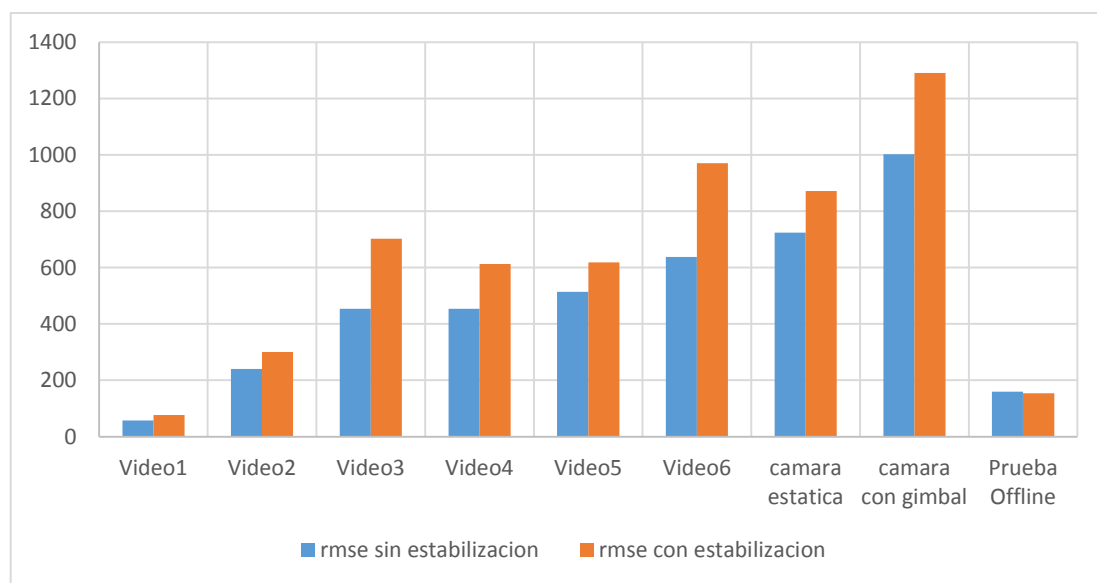


Figura 146: Comparación del RMSE total de los videos originales y los videos procesados.

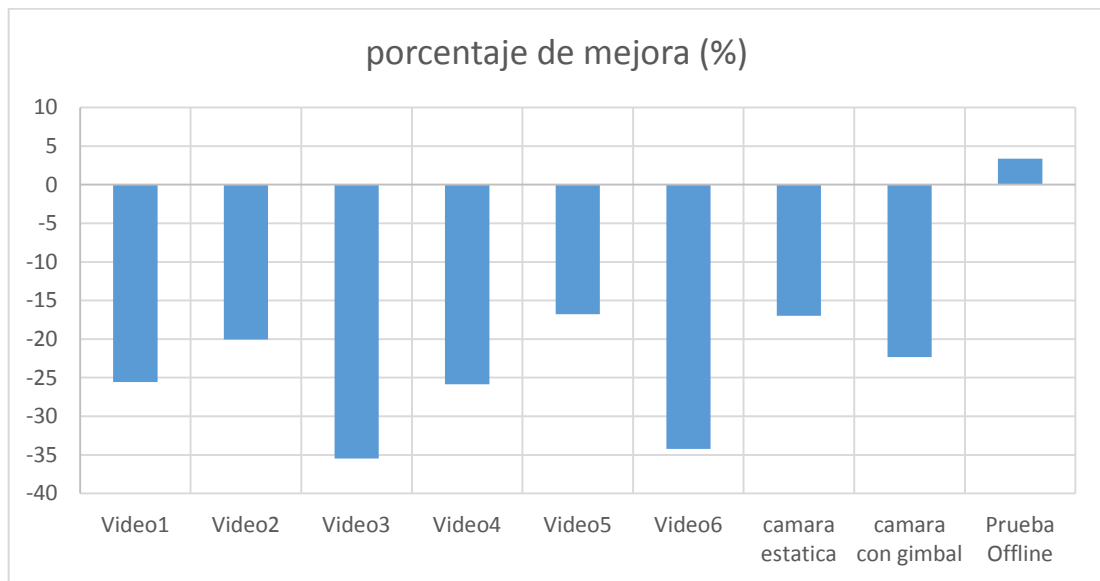


Figura 147: Porcentaje de mejora del uso del algoritmo de estabilización en base al RMSE entre los videos originales y los videos procesados.

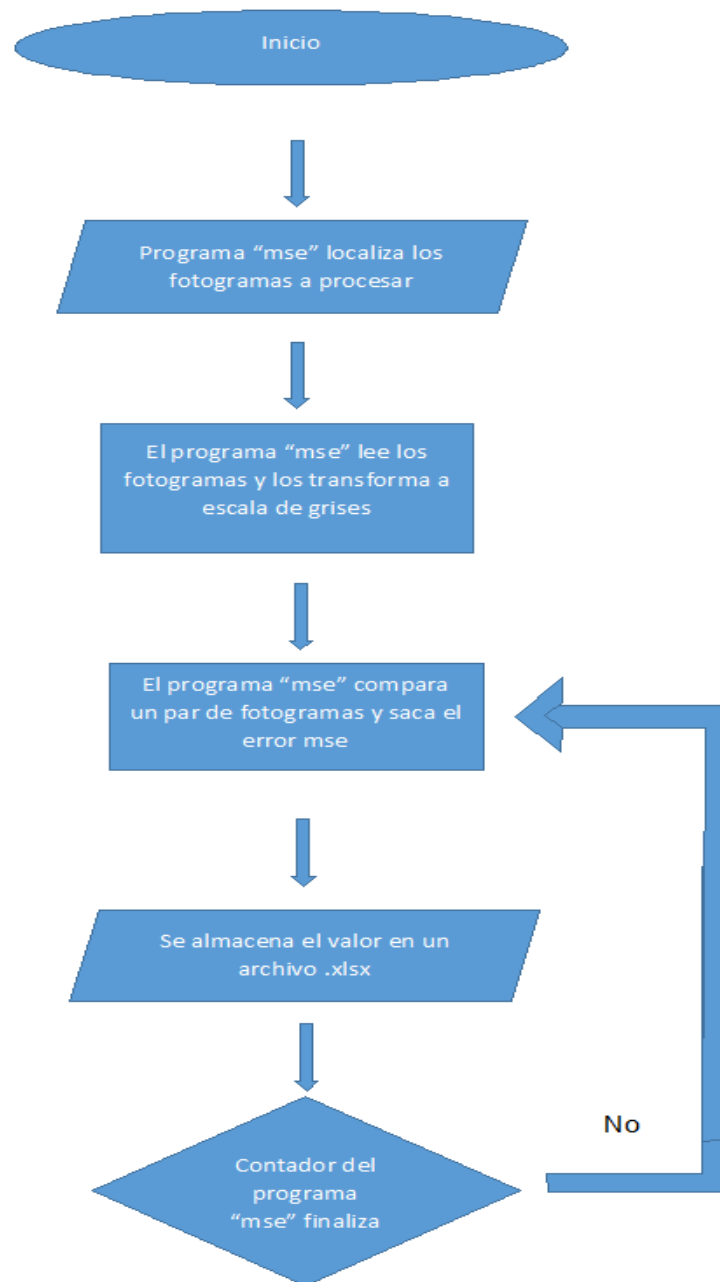


Figura 148: Pseudocódigo del programa "mse", que compara fotogramas en escala de grises y saca su error medio cuadrático.

4.4.2. Obtención del PSNR

El PSNR es un término ingenieril para la proporción entre la energía máxima posible de una señal y el poder de ruido que afecta la fidelidad de su representación.

$$PSNR(k) = 10 \log_{10} \frac{I_{p_{MAX}}}{MSE(k)}$$

Tabla 8

PSNR por fotograma

<i>Nombre de Muestra</i>	<i>Original</i>	<i>Procesado</i>
V6 frame 6 y 7	10.73	13.58
V6 frame 7 y 8	13.40	18.84
V3 frame 11 y 12	20.22	22.72
V6 frame 47 y 48	15.14	16.91
V6 frame 48 y 49	16.56	19.22
V4 frame 54 y 55	14.99	19.16
V6 frame 81 y 82	14.95	15.49
V6 frame 82 y 83	15.77	19.54
V5 frame 84 y 85	14.18	21.79
V6 frame 103 y 104	15.86	17.15
V6 frame 104 y 105	17.76	19.17
V2 frame 3 y 4	19.89	19.81
Video de cámara		
estática sin estabilizar,	12.72	16.64
frame 119 y 120		
Video de cámara con		
gimbal sin estabilizar,	13.43	16.65
frame 133 y 134		
Video offline del	22.86	24.5

Continúa



**algoritmo de
estabilización, frame
1222 y 1223**

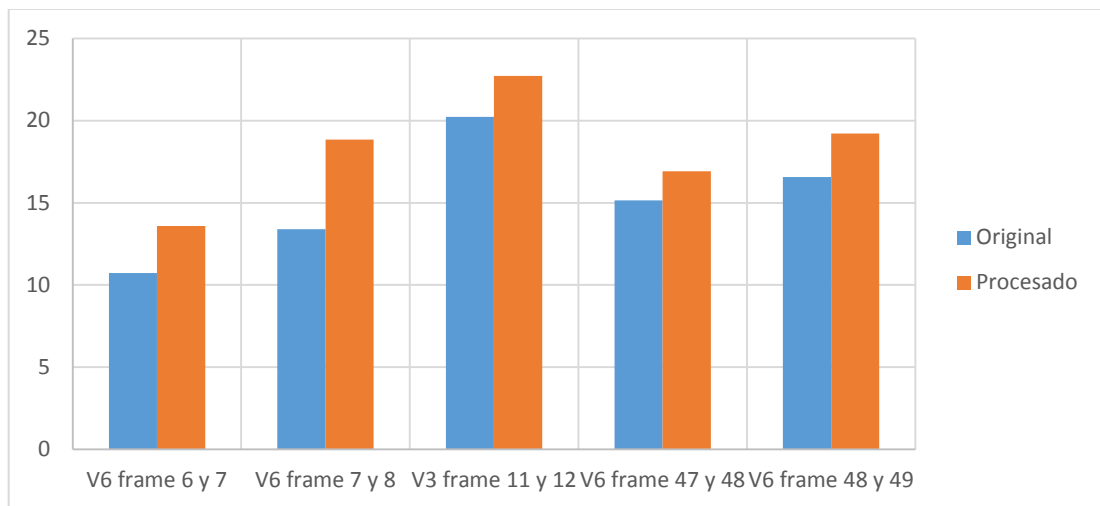


Figura 149: Comparación del PSNR de los fotogramas originales y los fotogramas procesados.

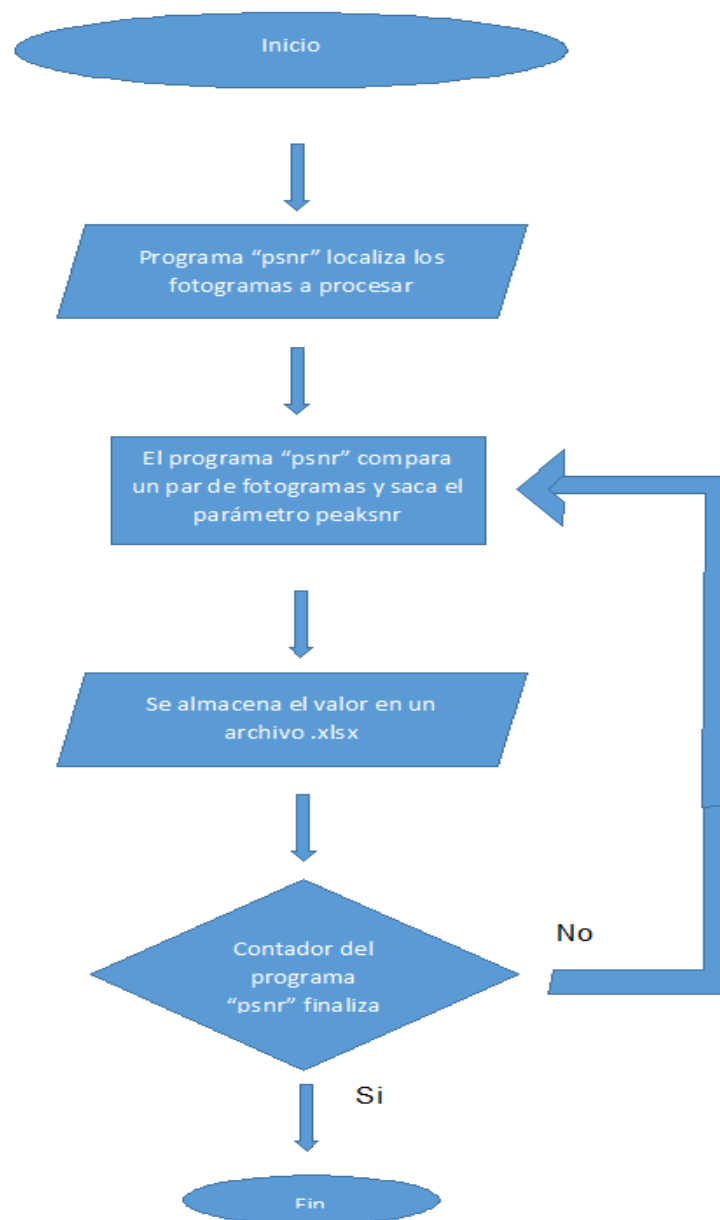


Figura 150: Pseudocódigo del programa "psnr", que compara fotogramas en proporción entre la energía máxima posible de una señal y el poder de ruido que afecta la fidelidad de su representación.

4.4.3. Obtención del ITF

El ITF es un método comúnmente usado para medir la efectividad y rendimiento de la estabilización de video, cuya fórmula es la siguiente:

$$ITF = \frac{1}{Nf - 1} * \sum_{k=1}^{Nf-1} PSNR(k)$$

Y de la cual se obtienen los valores de la siguiente tabla:

Tabla 9

ITF y Porcentaje de mejora

	<i>ITF sin estabilización</i>	<i>ITF con estabilización</i>	<i>Porcentaje de mejora (%)</i>
Video1	27.44	26.21	-4.69
Video2	21.05	19.88	-5.84
Video3	19.2	17.43	-10.10
Video4	21.05	19.81	-6.24
Video5	19.40	18.62	-4.23
Video6	18.10	16.81	-7.67
Cámara estática	17.29	16.68	-4.82
Cámara con gimbal	16.34	15.41	-6.04
Prueba Offline	23.66	24.22	2.30

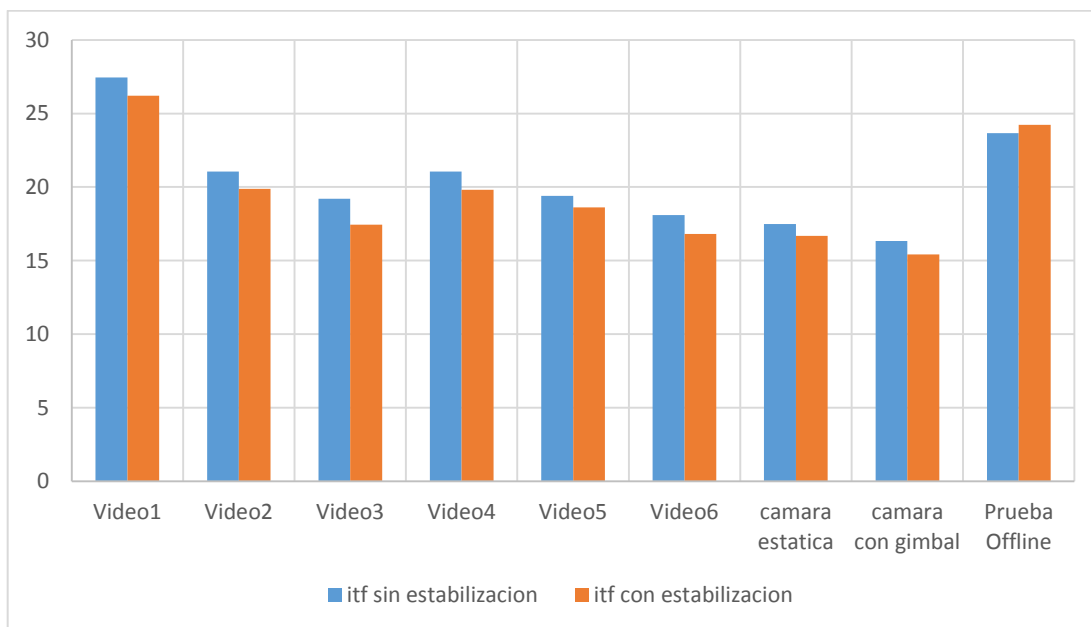


Figura 151: Comparación del ITF total de los videos originales y los videos procesados.

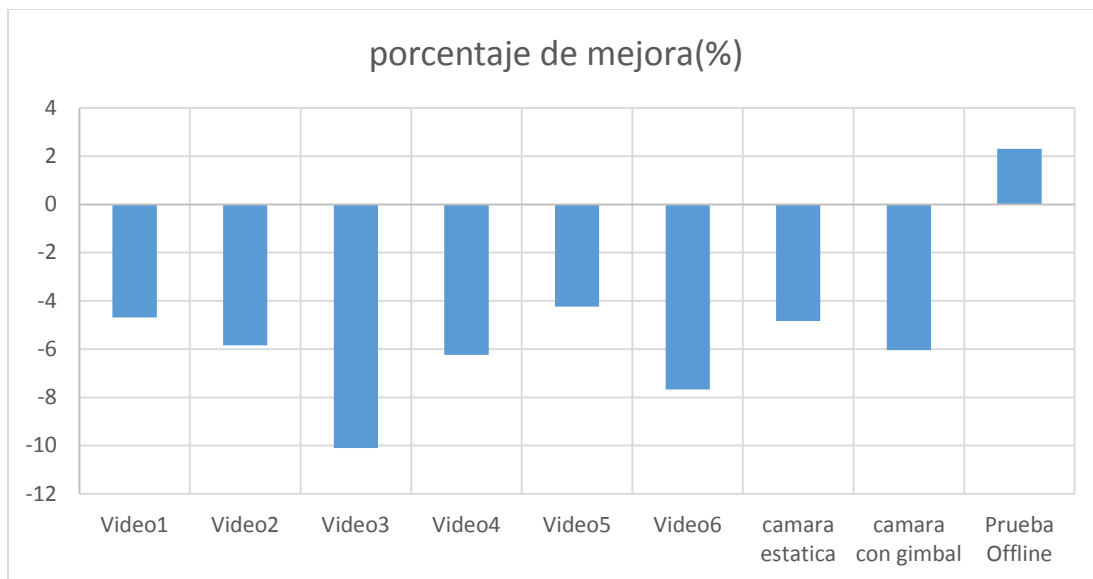


Figura 152: Mejora del uso del algoritmo de estabilización en base al ITF entre los videos originales y los videos procesados.

4.5. Resumen

- En el presente capítulo se muestran los resultados de las pruebas realizadas a los componentes del sistema y al sistema como tal.
- Se realizaron pruebas para comprobar la carga máxima soportada por el dron, las cuales demostraron que efectivamente este pertenece a la categoría de mini RUAV ya que soporta un peso máximo de 1.3kg.
- Se comprobó que las dimensiones de las placas utilizadas en el sistema de estabilización son iguales a las dimensiones de los diseños realizados en el software de simulación, esto se debe a que el material utilizado es acrílico y las cortadoras laser poseen una tolerancia de $\pm 0.2\text{mm}$.
- Se realizó una prueba simple pero efectiva, para comprobar el funcionamiento de los amortiguadores utilizados en la gimbal, los cuales a pesar de estar sometidos a una carga mayor a la carga que deben soportar no presentaron deformaciones.
- Durante las pruebas realizadas a las estaciones de control, se comprobó que Mission Planner realiza una configuración y calibración al Autopilot superior a la estación de control QGroundControl, ya que el dron presentaba mejor estabilidad durante el vuelo.
- El tiempo de funcionamiento fue estimado a partir de todos los componentes involucrados en el sistema, además de la energía requerida para cada dispositivo, lo que da a entender que en el corto tiempo de vuelo que tiene, se pudieron realizar vuelos para las pruebas correspondientes.
- Se empleó el método ITF al video resultante, el cual nos asegura una efectividad y rendimiento de la estabilización de nuestros videos resultantes, obteniendo el mejor resultado en la prueba offline.

CAPÍTULO 5

ANÁLISIS FINANCIERO Y ECONÓMICO

5.1. Análisis financiero

En el análisis financiero se toman en cuenta dos aspectos importantes, los cuales son:

5.1.1. Costos directos

Son aquellos costos que tienen relación directa con la producción. Estos costos se subdividen en partes de obras bien definidas, homogéneas y coherentes, llamadas fases, subdividida en cinco componentes fundamentales: Mano de obra directa, Materiales, Equipos, Combustibles, Subcontratistas. (Macchia, 2005).

5.1.1.1. Costos de materiales directos

En los cuadros a continuación se detallan todos los elementos que fueron adquiridos para la implementación del sistema de estabilización MkChaser.

En estos costos se encuentran:

- Materiales mecánicos.
- Materiales eléctricos y electrónicos.
- Materiales para estructura.
- Accesorios eléctricos y electrónicos.
- Accesorios para estructura.
- Costos de accesorios.
- Costos de materiales directos totales.

Tabla 10

Costos de materiales mecánicos

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unitario [\$]</i>	<i>Costo total [\$]</i>
Amortiguadores	4	20.40	20.40
Total:			20.40

Tabla 11

Costos de materiales eléctricos y electrónicos

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unitario [\$]</i>	<i>Costo total [\$]</i>
Odroid-XU4	1	98.39	98.39
AlexMos Basecam	1	80	80
DROK&reg DC-DC Buck Converter 12V/24V	1	35.45	35.45
Módulo Wifi 4 de Odroid	1	37.39	37.39
XU4 Shifter Shield	1	41.39	41.39
Módulo 8GB eMMC 5.0 XU3/XU4 Linux	1	45.39	45.39
Módulo USB-UART	1	34.39	34.39
Módulo USB GPS	1	48.39	48.39
Motor sin escobillas 2804-210Kv	2	33.91	67.82
Motor sin escobillas BGM4108-130t	1	87.10	87.10
Kit Pixhawk PX4 2.4.6	1	191.44	191.44
Total:			767.15

Tabla 12

Costos de materiales para estructura

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unitario [\$]</i>	<i>Costo total [\$]</i>
Gimbal	1	21.86	21.86
Soporte Motor	1	8	8
Soporte Base	1	15	15
Case Odroid XU4	1	14.35	14.35
Total:			59.21

Tabla 13

Accesorios eléctricos y electrónicos

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unitario [\$]</i>	<i>Costo total [\$]</i>
Cables			15
Espagueti térmico	2	2.25	4.50
Espadines	1	0.95	0.95
Flux	1	6	6
Estaño	1	3	3
Total:			29.45

Tabla 14

Accesorios para estructura

<i>Descripción</i>	<i>Cantidad</i>	<i>Precio Unitario [\$]</i>	<i>Costo total [\$]</i>
Tornillo	16	0.15	2.40
Pernos	4	0.15	0.60
Separadores	8	0.20	1.60
Taipe	1	3.00	3.00
Cinta doble faz	1	5.00	5.00
Total:			12.60

Continúa



Tabla 15

Costos de accesorios

<i>Descripción</i>	<i>Costo total [\$]</i>
Accesorios eléctricos y electrónicos	29.45
Accesorios para estructura	12.60
Total:	42.05

Los costos directos totales que intervienen en el tema desarrollado se presentan en la siguiente tabla.

Tabla 16

Costos directos totales de materiales

<i>Descripción</i>	<i>Costo total [\$]</i>
Elementos mecánicos	20.40
Elementos eléctricos y electrónicos	767.15
Elementos para la estructura	59.21
Costos accesorios	42.05
Total:	888.81

5.1.1.2. Costos de mano de obra directa

Son los costos que corresponden a las personas que intervinieron en la implementación del proyecto. En este caso constan la mano de obra para las piezas adquiridas, la programación y adaptación del sistema al STORM Drone V6, en la siguiente tabla se detallan los costos.

Tabla 17

Mano de obra directa

<i>Encargado</i>	<i>Actividad</i>	<i>Días</i>	<i>Nro de Horas</i>
Tesistas	Medición de las piezas	1	2
	Generación de planos	1	2
Director del proyecto	Revisión y aprobación	1	5
Tesistas	Compra de material	5	35
	Programación	60	210
	Investigación	40	140
Total:		108	394

Por lo tanto, en aproximadamente 3 meses 18 días de trabajo, se emplearon 394 horas para desarrollar el sistema de estabilización para el STORM Drone V6. Es necesario conocer el costo de la hora de trabajo de las personas que intervinieron. El costo depende del salario que poseen; un salario básico es de 366 dólares, lo que significa que trabajando 8 horas diarias y los 20 días laborables del mes, el costo por hora de trabajo es de \$ 2.28.

El costo por hora del directo del proyecto se lo calcula de la misma manera, el salario es de 2400 dólares, por lo tanto, el costo por hora es de \$ 15. El costo total de mano de obra directa se presenta en la siguiente tabla.

Tabla 18

Costo de mano de obra directa

<i>Descripción</i>	<i>Nro de Horas</i>	<i>Costo por hora [\$]</i>	<i>Costo total [\$]</i>
Tesistas	391	2.28	891.48
Director del proyecto	5	15	75
Total:			966.48

5.1.2. Costos totales directos

En la siguiente tabla se presenta los costos totales directos para el proyecto.

Tabla 19

Costos totales directos

<i>Descripción</i>	<i>Costo [\$]</i>
Materiales directos	888.81
Mano de obra directa	966.48
Total:	1855.29

5.2. Costos indirectos

Los costos indirectos son aquellos que no intervienen directamente con la implementación y el estudio necesario para el mismo. Pero son indispensables para lograr la elaboración del proyecto, es por esto que se los toma en cuenta para el estudio financiero (Macchia, 2005).

5.2.1. Costos totales indirectos

En este caso se consideran factores como: la electricidad, el agua y el internet. En la siguiente tabla se encuentra detallado el costo de los mismos.

Tabla 20

Costos totales indirectos

<i>Descripción</i>	<i>Días</i>	<i>Horas</i>	<i>Costo por hora [\$]</i>	<i>Costo total [\$]</i>
Electricidad	107	394	0.0824	32.4656
Internet	107	394	0.055	21.67
Agua Potable	107	394	0.1	39.4
Total:			0.2374	93.5356

5.3. Costos totales

El costo total del proyecto es la suma de los costos directos e indirectos, en la siguiente tabla se presenta el costo total del proyecto.

Tabla 21

Costo total del proyecto

<i>Descripción</i>	<i>Costo [\$]</i>
Costos directos	1855.29
Costos indirectos	93.5356
Total:	1948.8256

RESUMEN

Este sistema de estabilización propuesto en la actualidad no tiene comparativa en el mercado, debido a que su implementación es relativamente nueva debido a su desarrollo en el campo de investigación destinado al CICTE.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

En el presente capítulo se presentan las conclusiones y recomendaciones que se han obtenido a partir del desarrollo del sistema de estabilización.

6.1. Conclusiones

Con la elaboración del presente proyecto se han obtenido las siguientes conclusiones:

- Se tomaron en cuenta las dimensiones de los diferentes componentes mecánicos y electrónicos para tener una distribución de fuerzas que generen el menor momento de inercia en el espacio reducido que se posee, la base del motor del eje “Yaw” es el principal elemento que permite la unión entre la gimbal y la base total del sistema mediante amortiguadores, ya que de esta manera se evita el efecto “jitter” en el video resultante obtenido.
- Se seleccionó el Autopilot Pixhawk, debido a que es de código abierto y permite el acceso a las variables medidas por los sensores mediante MAVLink, además permite la utilización de una gran variedad de estaciones de control.
- Se elaboró el sistema de estabilización de video que involucra en ingreso de la señal RC y el fotograma obtenido por la cámara, los cuales son enviados al código de programación que procesa los datos mediante OpenCV y WiringPi, donde la señal de control del filtro de Kalman obtenido de la imagen procesada, dirige a la gimbal minimizando el efecto de los movimientos fantasma.
- Se calculó los errores de covarianza de los fotogramas y la función de transferencia en espacios de estado obtenida mediante el filtro de Kalman a

partir de los fotogramas procesados en MATLAB, cuyas variables se utilizaron en el código del programa para la estabilización del video.

- Se comprobó que los componentes electrónicos y mecánicos se acoplaron adecuadamente en las ubicaciones definidas para cada uno sobre la base de la estructura y se realizó el cálculo inercial respectivo, el cual dio como resultado un momento de inercia de $49.467 N \cdot mm$ en sentido anti-horario tomando en cuenta la posición de la cámara ubicada al lado la izquierda.
- El sistema se evaluó comparando el video sin procesar y el video procesado, obteniendo como resultado las siguientes mejoras: 9.718% en RMSE, 16.706 % en PSNR, y 20.618% en ITF, lo que demuestra que este sistema de estabilización de video es funcional.

6.2. Recomendaciones

- Para la ubicación de los diferentes dispositivos en la estructura se debe considerar la extensión de los cables ya que pueden generar un peso adicional al sistema, pueden impedir el movimiento de la gimbal o desconectarse al ser estos muy cortos.
- Para el uso de la estación de QGroundControl con Pixhawk se debe poseer sensores adicionales para que ayuden a realizar un vuelo más estable.
- Se recomienda que para la lectura de la señal RC que involucra la programación del sistema de estabilización, se posea un dispositivo dedicado la lectura del PWM, el cual envíe el dato leído en tiempo real al Odroid XU4 para que el proceso de estabilización sea mucho más eficaz.
- Se recomienda utilizar la versión 2014b de MATLAB o posteriores a esta para obtener los datos utilizados en este proyecto, requerido para la estabilización, si se realizara el cambio de gimbal en el sistema.

BIBLIOGRAFÍA

- Aguilar, W. G., & Angulo, C. (2014a). Optimization of Robust Video Stabilization based on Motion Intention for Micro Aerial Vehicles. *Systems Signals and Devices (SSD), 2014 International Multi-Conference on*, (February 2014), Accepted. <http://doi.org/10.1109/SSD.2014.6808863>
- Aguilar, W. G., & Angulo, C. (2014b). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 2014(1), 46. <http://doi.org/10.1186/1687-5281-2014-46>
- Aguilar, W. G., & Angulo, C. (2015). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, (JANUARY 2015), 1–19. <http://doi.org/10.1007/s11063-015-9439-0>
- Alee, N., Rahman, M., & Ahmad, R. B. (2011). Performance comparison of Single Board Computer: A case study of kernel on ARM architecture. *ICCSE 2011 - 6th International Conference on Computer Science and Education, Final Program and Proceedings*, (Iccse), 521–524. <http://doi.org/10.1109/ICCSE.2011.6028693>
- Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3), 283–310. <http://doi.org/10.1007/BF00158167>
- Basecam. (2013). Software User Manual.
- Bay, H., Tuytelaars, T., Van Gool, L., Leonardis, A., Bischof, H., & Pinz, A. (2006). SURF: Speeded Up Robust Features. *Computer Vision – ECCV 2006*, 3951, 404–417. http://doi.org/10.1007/11744023_32
- Bradski, G., & Kaehler, A. (2008). *Projection and 3D Vision. Learning OpenCV*. <http://doi.org/10.1109/MRA.2009.933612>

- Burt, P. J., & Adelson, E. H. (1983a). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4), 217–236. <http://doi.org/10.1145/245.247>
- Burt, P. J., & Adelson, E. H. (1983b). The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, 31(4), 532–540. <http://doi.org/10.1109/TCOM.1983.1095851>
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF : Binary Robust Independent Elementary Features. *European Conference on Computer Vision (ECCV)*, 778–792. http://doi.org/10.1007/978-3-642-15561-1_56
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698. <http://doi.org/10.1109/TPAMI.1986.4767851>
- Chao, H., Cao, Y., & Chen, Y. (2010). Autopilots for Small Unmanned Aerial Vehicles : A Survey, 8, 36–44. <http://doi.org/10.1007/s12555-010-0105-z>
- Chao, H., Luo, Y., Di, L., & Chen, Y. (2009). Fractional Order Flight Control of a Small Fixed-Wing UAV: Controller Design and Simulation Study, 1–8.
- Choi, S., Kim, T., & Yu, W. (2009). Performance Evaluation of RANSAC Family. *Proceedings of the British Machine Vision Conference 2009*, (2), 81.1-81.12. <http://doi.org/10.5244/C.23.81>
- Cook, K. L. B. (2007). The silent force multiplier: The history and role of UAVs in warfare. *IEEE Aerospace Conference Proceedings*. <http://doi.org/10.1109/AERO.2007.352737>
- D. Terzopoulos A. Witkin, M. K., & D Terzopoulos, A. W. (1988). Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial*, 36(91), 91–123. Retrieved from papers3://publication/uuid/A006D127-4F85-41E3-9176-9FAAA3410F5A

- Debevec, P. E., & Malik, J. (1997). Recovering High Dynamic Range Radiance Maps from Photographs. *University of California at Berkley*, (August), 1–10. <http://doi.org/10.1145/258734.258884>
- Drahanský, M., Orság, F., & Hanáček, P. (2010). Accelerometer Based Digital Video Stabilization for General Security Surveillance Systems. *International Journal of Security and ...*, 4(1), 1–10. Retrieved from http://www.sersc.org/journals/IJSIA/vol4_no1_2010/1.pdf
- Fang, Y. Y., & Chen, X. J. (2011). Design and simulation of UART serial communication module based on VHDL. *2011 3rd International Workshop on Intelligent Systems and Applications, ISA 2011 - Proceedings*, 1. <http://doi.org/10.1109/ISA.2011.5873448>
- Faugeras, O. D. (1992). What can be seen in three dimensions with an uncalibrated stereo rig? *Eccv*, 588, 563–578. http://doi.org/10.1007/3-540-55426-2_61
- Federal Aviation Administration. (2009). *Automated Flight Control. Advanced Avionics Handbook*.
- García, A., & Sanz del Río, A. (2016). Soporte de sistema operativo para ahorro de energía en plataformas móviles con procesadores multicore asimétricos, 94. Retrieved from <http://eprints.ucm.es/38726/1/Memoria.pdf>
- González, A. R. (2015). Metodología de la Investigación Científica. *Facultad de Estudios Ambientales Y Rurales de La Pontificia Universidad Javeriana*, 22(7), 1527–1531. <http://doi.org/10.1016/j.conbuildmat.2007.03.027>
- Hartley, R. I. (1992). Estimation of Relative Camera Positions for Uncalibrated Cameras. *Computer Vision-ECCV'92*, (1), 579–587. http://doi.org/10.1007/3-540-55426-2_62
- Hartley, R. I. (1994). Projective reconstruction and invariants from multiple

- images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10), 1036–1041. <http://doi.org/10.1109/34.329005>
- Hilkert, J. M. (2008). Inertially stabilized platform technology: Concepts and principles. *IEEE Control Systems Magazine*, 28(1), 26–46. <http://doi.org/10.1109/MCS.2007.910256>
- Hulens, D., Verbeke, J., & Goedemé, T. (2016). Choosing the best embedded processing platform for on-board UAV image processing. *Communications in Computer and Information Science*, 598, 455–472. http://doi.org/10.1007/978-3-319-29971-6_24
- IdePlas. (2012). *Ficha técnica lamina PMMA*.
- InvenSense. (2013). *MPU-6000 and MPU-6050 Product Specification* (Vol. 1).
- Ivković, J., Veljović, A., Ranđelović, B., & Veljović, V. (2016). ODR0ID-XU4 as a desktop PC and microcontroller development boards alternative. *Technics and Informatics in Education*, 439–444.
- Karpenko, A., Jacobs, D., Baek, J., & Levoy, M. (2011). Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. *Stanford Tech Report CTSR*, 1–7. Retrieved from http://171.67.77.70/papers/stabilization/karpenko_gyro.pdf
- Kass, M., Witkin, a., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4), 321–331. <http://doi.org/10.1007/BF00133570>
- Kendoul, F. (2011). Path-following algorithms and experiments for an autonomous surface vehicle. *Journal of Field Robotics*, 7(PART 1), 81–86. <http://doi.org/10.1002/rob>
- Lindeberg, T. (1990). Scale-Space for Discrete Signals. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, 12(3), 234–254.
<http://doi.org/10.1109/34.49051>

Lonvick, C. (2006). *No Title*.

Madaan, P. (2013). Brushless DC Motors – Part I : Construction. *EDN Network*, 1, 7.

Mair, E., Hager, G. D., Burschka, D., Suppa, M., & Hirzinger, G. (2010). Adaptive and generic corner detection based on the accelerated segment test. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6312 LNCS(PART 2), 183–196. http://doi.org/10.1007/978-3-642-15552-9_14

Martínez, J., & Portillo, R. (2013). *SISTEMA MECATRÓNICO PARA LA ATENUACIÓN DE MOVIMIENTO Y CONTROL DE ADQUISICIÓN DE IMÁGENES*. Instituto Politécnico Nacional.

Mendez, W., Rodriguez, Y., Brady, L., & Tosunoglu, S. (2010). Design of A Three–Axis Rotary Platform. *Florida Conference on Recent Advances in Robotics–FCRAR*, 20–21.

Molloy, D. (2016). *Exploring Raspberry Pi*.

Motor Consultants, M. (2015). *White Paper How to Acquire the Best Application Advantage*.

Nalwa, V. S., & Binford, T. O. (1986). On Detecting Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), 699–714.
<http://doi.org/10.1109/TPAMI.1986.4767852>

Negus, C., & Caen, F. (2008). <http://elib.ntt.edu.vn/>. Indianapolis: Wiley Publishing, Inc. Retrieved from <http://elib.ntt.edu.vn/>

Novatel. (2014). IMU Errors and Their Effects. Report APN-064 (Rev A), 1–6.

- Ovr, H., & Forssén, P.-E. (2015). Gyroscope-based Video Stabilisation With Auto-Calibration, 2090–2097.
- Pierre Vigouroux Cavolina, D. (2010). *Implementación De Unidad De Mediciones Inerciales (Imu) Para Robótica Utilizando Filtro De Kalman*. Universidad Simón Bolívar.
- Preckler Clemente, C. (2012). *Caracterización De Una Unidad De medida Inercial Para La Obtencion De Entornos De Simulacion De Un Sistema De Navegacion Inercial De Bajo Coste Mediante Allan Variance*. Universidad Autónoma de Barcelona.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... Mg, A. (2009). ROS: an open-source Robot Operating System. *Icra*, 3(Figure 1), 5. <http://doi.org/http://www.willowgarage.com/papers/ros-open-source-robot-operating-system>
- Rosenfeld, A., & Pfaltz, J. L. (1966). Sequential Operations in Digital Picture Processing. *Journal of the ACM*, 13(4), 471–494. <http://doi.org/10.1145/321356.321357>
- Ruble, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Proceedings of the IEEE International Conference on Computer Vision*, 2564–2571. <http://doi.org/10.1109/ICCV.2011.6126544>
- Solera Ramírez, Á. (2003). El filtro de kalman. *Nota Técnica, Departamento de Investigaciones Económicas, Banco Central de Costa Rica*.
- Szeliski, R. (2010). Computer Vision : Algorithms and Applications. *Computer*, 5, 832. <http://doi.org/10.1007/978-1-84882-935-0>
- Tarot Aviation Technology, C. (2015). Tarot T4-3d.

- Terzopoulos, D., & Fleischer, K. (1988). Deformable models. *Visual Computer*, 4(6), 306–331. <http://doi.org/10.1007/BF01908877>
- Terzopoulos, D., Witkin, A., & Kass, M. (1987). Symmetry-Seeking Models and 3D Object Reconstruction. *International Journal of Computer Vision*, 221, 211–221. <http://doi.org/10.1007/BF00127821>
- Wasenm, O. (2015). 3D Computer Vision & Augmented Reality.
- Witkin, A. (1984). Scale-space filtering: A new approach to multi-scale description. *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 9, 150–153. <http://doi.org/10.1109/ICASSP.1984.1172729>
- Witkin, A., Terzopoulos, D., & Kass, M. (1987). Signal matching through scale space. *International Journal of Computer Vision*, 1(2), 133–144. <http://doi.org/10.1007/BF00123162>
- Yu, J., Wu, K., Zong, G., & Kong, X. (2016). A Comparative Study on Motion Characteristics of Three Two-Degree-of-Freedom Pointing Mechanisms. *Journal of Mechanisms and Robotics*, 8(2), 21027. <http://doi.org/10.1115/1.4032403>
- Zhao, J., & Yangwei, Y. (2011). Brushless DC Motor Fundamentals Application Note. *MPS, The Future of Analog IC Technology*, (July 2011), 7–8.

ANEXOS