

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

PROGRAMA DE SISTEMAS AUTÓNOMOS DE
NAVEGACIÓN AÉREA Y SUBMARINA

“Control de posición de un
cuadri-rotor mediante
visión monocular”

TESIS

Que presenta

Ing. José Isaac Flores Delgado

Para obtener el grado de
Maestro en Ciencias

EN

SISTEMAS AUTÓNOMOS DE NAVEGACIÓN
AÉREA Y SUBMARINA

Director de Tesis:

Dr. Rogelio Lozano Leal

Ciudad de México

Febrero 2017

AGRADECIMIENTOS

Agradezco a la vida y a Dios por permitirme concluir con satisfacción este peldaño en mi carrera, a mi familia, la cual a estado presente en cada paso de mi trayectoria. En especial a mi madre Claudia Delgado Aguilar quien ha hecho muchos sacrificios para brindarme la oportunidad de tener una carrera universitaria.

Agradezco también al amor de mi vida Laura Guadalupe, quien a estado presente a lo largo de este gran sueño que ha sido la maestría.

Extiendo el agradecimiento a mis asesores de tesis el Dr. Rogelio Lozano, el Dr. Iván González, Dr. Diego Alberto Mercado y el Dr. Hugo Romero, quienes han estado al pendiente para la realización del presente trabajo, así como a todos y cada uno de los amigos que componen la gran familia del Laboratorio UMI-LAFMIA, de quienes he aprendido mucho. En especial a Josue Cariño Escobar, Orlando García Pérez, Donovan Flores Meza, Yukio Rosales Luengas quienes me han apoyado durante algunas etapas del proyecto

Agradezco también a todas aquellas personas que han formado parte de mi vida, y sobre todo alas que han contribuido con mi desarrollo profesional y personal.

Por ultimo pero no menos importante agradezco profundamente el apoyo y la oportunidad que el CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA (CONACYT) me ha proporcionado para poder concluir con mis estudios de Maestría.

RESUMEN

Actualmente el posicionamiento de los vehículos aéreos no tripulados (UAV) llamados drones, es realizado con GPS con el propósito de explorar algunos ambientes, implementar seguimiento de trayectorias, repartición de pizzas, brigadas de búsqueda, detección de defectos estructurales, amenazas o de identidad.

Para llevar a cabo estos ejemplos es necesario conocer la posición del drone mientras esta en operación, por que si se conoce la posición, es posible tener control de sus movimientos. Por lo que las aplicaciones son enormes.

Pero que es lo ¿Que sucede cuando no es posible utilizar GPS? ¿Como el drone puede obtener su posición?

Una manera de hacerlo es respecto a un objeto. Por lo que en esta tesis se estará hablando de como un drone puede ser posicionado respecto a un rostro humano.

En el futuro cuando se refiera a un Cuadri-rotor s estará refiriendo a un vehículo aéreo no tripulado.

En el presente trabajo, el posicionamiento esta basado en un algoritmo de detección. El cual puede detectar el rostro humano, para si poder realizar un seguimiento del mismo. Es decir el cuadri-rotor es capaz de modificar su posición para ubicarse exactamente enfrente del rostro humano.

VI

Para ello el cuadri-rotor fue dotado de la capacidad visual por una cámara de tipo ojo de pez, cuya función es la detección del rostro y proveerle al auto piloto de los parámetros necesarios de control, con el fin de llevar el posicionamiento del cuadri-rotor, respecto al rostro humano.

Ya que el procesamiento de imágenes es una tarea altamente demandante para las computadoras, y como uno de los objetivos principales realzar un procesamiento embarcado en el dron, se realizo la selección de una computadora embebida cuyo nombre es Odroid XU4.

Este reto fue conseguido utilizando una ley de control de tipo PD, cuyos resultados experimentales, son observados en el desempeño del posicionamiento del cuadri-rotor respecto a una cara. Trayendo consigo un nuevo resultado dentro de los trabajos con vehículos aéreos no tripulados.

ABSTRACT

Actually unmaned aerial vehicles (UAV) also named drones are positioned with GPS in order to explore some environments, go through trajectories, delivering pizza, brigades search, threats detection or inclusive identity recognition.

To do all this examples we need to know position during whole work, beacuse if we have drone's position, we can get control of movements. So that we get a high range of applications.

But What happen when GPS is denied? How can drone gets its position?

One way to do it, is regarding an object. In this thesis we will be talked about how drone can be situated respect one human face.

In future when we talk about Quadcopter we refer to Unmaned aerial vehicle.

Positioning in present work is based on a detection algorithm. This algorithm can detect a human face, so that follow it. In other words quadcopter is able to modify its position in order to situate it in front of human face.

For that quadcopter have been constructed with a fish eye camera, which function is face detection and provide control parameters to implement in an auto pilot, in order to achieve face position control.

VIII

Since image processing is a really hard task to computers, and one of main objectives to carry out processing within unmaned aerial vehicle, we selected a embedded computer which name is Odroid XU4.

Task was completed using PD control law, which experimental results, it will be noticed in behavior of quadcopter position respect with a human face. Bring us a new results within unmaned aerial vehicles works.

ÍNDICE GENERAL

AGRADECIMIENTOS	III
RESUMEN	V
ABSTRACT	VII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XIII
ÍNDICE DE FIGURAS	XV
1 INTRODUCCIÓN	1
1.1 OBJETIVOS	3
1.2 JUSTIFICACIÓN	3
2 PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA SOLUCIÓN	5
3 MODELADO MATEMÁTICO DEL CUADRI-ROTOR	7
3.1 Diagrama de Cuerpo Libre de un Cuadri-rotor	8
3.1.1 Modelo matemático mediante método Euler-Lagrange	9
3.1.2 Modelo matemático mediante método Newton-Euler	10

4	PROTOTIPO EXPERIMENTAL	13
4.1	Sensores y Dispositivos	13
4.2	Estructura Original y Rediseño	18
4.3	Configuración de los Motores Plataforma Experimental	22
4.4	Plataforma experimental final	24
5	ALGORITMOS DE VISIÓN ARTTIFICIAL	27
5.1	Detección de Rostros	28
5.2	Seguimiento de Rostros	29
6	COMUNICACIÓN	33
6.1	Parámetros de Control	34
6.2	Protocolo Envío	35
7	ESTRATEGIA DE CONTROL	37
7.1	Control de Orientación	38
7.1.1	Desarrollo Estrategia de Control	40
7.2	Control PD para Seguimiento Facial	43
7.2.1	Desarrollo Estrategia de Control	44
8	RESULTADOS EXPERIMENTALES	47
8.1	Resultados Computadora Embebida (Odroid XU4)	48
8.1.1	Calibración de Distancia a Objetivo	48
8.1.2	Calibración del Desplazamiento Lateral del Objetivo	50
8.1.3	Comportamiento de posiciones del cuadri-rotor	51
8.1.4	Comportamiento de velocidades del cuadri-rotor	55
8.2	Resultados Pixhawk Compatible (Autopiloto)	57

<i>ÍNDICE GENERAL</i>	XI
8.2.1 Datos Recibidos	57
9 CONCLUSIONES	65
10 TRABAJO FUTURO	67
A ANEXO TRABAJO ADICIONAL REALIZADO	69
A.1 Clasificador Aumentado	69
A.1.1 Detección de 3 Flancos de un Rostro	71
A.2 Datos Procesados Computadora Embebida (Odroid XU4)	73
A.2.1 Comportamiento de Posiciones y Velocidades Procesadas	73
A.3 Datos Recibidos Autopiloto (Pixhawk)	77
A.3.1 Comportamiento de Posiciones y Velocidades Recibidas	77
B ANEXO DE PSEUDOCÓDIGOS	81
BIBLIOGRAFÍA	85

ÍNDICE DE TABLAS

6.1	Parámetros de Error en Posición del Cuadri-rotor	34
6.2	Parámetros de Error en Posición del Cuadri-rotor	34
6.3	Datos Enviados	36

ÍNDICE DE FIGURAS

3.1	Diagrama de Cuerpo Libre	8
4.1	Computadora embebida Odroid XU4	14
4.2	Esquema Computadora embebida Odroid XU4	15
4.3	Distorsión Cámara Ojo de Pez	16
4.4	Cámara Ojo de Pez Seleccionada	16
4.5	Cámara Ojo de Pez Seleccionada	17
4.6	Autopiloto Pixhawk	18
4.7	Estructura comercial.	19
4.8	Base estructura comercial	20
4.9	Brazo Estructura	21
4.10	Fijación Cámara	22
4.11	Tipos de Configuraciones	23
4.12	Tipos de giros para las hélices	23
4.13	Versión Previa Vista Frontal	24
4.14	Versión Previa Vista Lateral	25
4.15	Versión 3 Final	26
5.1	Filtro de Kalman Velocidades	30

5.2	Filtro de Kalman Ecuaciones	31
7.1	Ángulos de Rotación	39
7.2	Dinámica Completa	40
8.1	Calibración Cámara	48
8.2	Calibración Cámara Distancia a Persona	49
8.3	Calibración Cámara Pixel por cm para Distancia Lateral	51
8.4	Posiciones Originales	52
8.5	Filtro de Kalman Posiciones	53
8.6	Error Posicionamiento Respecto al Centro de La Cámara	54
8.7	Error Posicionamiento Respecto al Rostro	54
8.8	Filtro de Kalman Velocidades	55
8.9	Error Velocidad Respecto al Centroide de La Cámara	56
8.10	Error Velocidad Respecto al Rostro	57
8.11	Error de Posición y Velocidad en el Z	58
8.12	Error de Posición y Velocidad en el Y	58
8.13	Error de Posición y Velocidad en el X	59
8.14	Error Filtrado de Posición y Velocidad en el Z	60
8.15	Error Filtrado de Posición y Velocidad en el Y	60
8.16	Error Filtrado de Posición y Velocidad en el X	61
8.17	Señales de Control	62
8.18	PWM en Motores	62
8.19	Versión 3 Final	63
A.1	Detecciones y su identificación	70
A.2	Relación de Perfiles Detectados	71

A.3	Perfiles Detectados Decodificados	72
A.4	Posiciones Originales	73
A.5	Posiciones Filtro de Kalman en Ejes Y y Z	74
A.6	Velocidad Filtro Kalman en Ejes Y y Z	74
A.7	Posiciones Ajustadas al Centro en Ejes Y y Z	75
A.8	Errores de Posición en 3 Ejes	76
A.9	Errores de Velocidad en 3 Ejes	76
A.10	Errores de Posición y Velocidad en Eje Z	77
A.11	Errores de Posición y Velocidad en Eje Y	78
A.12	Errores de Posición y Velocidad en Eje X	78
A.13	Errores de Velocidad en 3 Ejes	79

CAPÍTULO 1

INTRODUCCIÓN

Actualmente las aplicaciones de los vehículos aéreos no tripulados comúnmente llamados drones dentro de la vida cotidiana son cada vez mas comunes, ya que las tareas que pueden realizar son versátiles. Una de las razones principales para su uso, es debido a que estos vehículos pueden ingresar a lugares en los cuales la interacción humana fracasa. Estas tareas pueden ser principalmente aplicaciones civiles o militares. Algunos ejemplos de estas son: la transportación de materiales, la detección, reconocimiento y búsqueda de personas, lugares o inclusive fallas estructurales en edificios o materiales.

Pero para llevar a cabo todas y cada una de estos ejemplos mencionados, es preciso tomar en cuenta mas de una variable, entendiendo como variable a todo aquel factor que afecta el desempeño del vehículo aéreo. Dichas variables comienzan con la tarea a realizar e influyen directamente en consideraciones como son el rango de operación en altura o presión, en el empleo de GPS. Cuyo efecto principal recae en el peso del vehículo. El cual esta directamente relacionado con los sensores que se encontraran en el vehículo.

En general los procesos de visión artificial son empleados para realizar tareas cuyo

objetivo principal es la detección, el reconocimiento y de forma general la búsqueda.

Para ello el vehículo aéreo es dotado de un sensor visual que de ahora en adelante llamaremos cámara, la cual añade al vehículo un sentido similar al de la vista humana. Para así transformar todo aquello que es enfocado en información, la cual es procesada para determinar la presencia del objeto de interés esta presente en el escenario captado.

Algunos trabajos relevantes que emplean la visión artificial en vehículos no tripulados son: [8], [10], [11], [13], [14], [16], [17] [18], [19], [20].

Los trabajos anteriores tienen como principal herramienta la visión de tipo monocular, es decir emplean solo una cámara para el procesamiento de imágenes. Es importante decir que esta configuración en comparación con la visión estereoscópica, representa la mitad de procesamiento, lo cual brinda un aumento en la velocidad de ejecución de los algoritmos. Aunque por el contrario se pierde la perspectiva que la segunda cámara pudiera aportar.

Para mayor información sobre trabajos con configuración estereoscópica se puede consultar [12].

Ya que el presente trabajo estará enfocado a la detección y seguimiento de un rostro humano como objetivo principal, se ha trabajado con algoritmos de visión artificiales, los cuales en conjunción con un control han de realizar la tarea antes mencionada.

1.1 OBJETIVOS

En el presente trabajo de tesis tiene como principales objetivos:

1. Implementar un control para el seguimiento de un rostro utilizando visión monocular empleando un helicóptero de 4 rotores equipado con una cámara ojo de pez.
2. Detección y seguimiento de un rostro humano.
3. Procesamiento Imagen dentro de una computadora embebida.

1.2 JUSTIFICACIÓN

El presente trabajo implementa e innova algunos principios manejados en los siguientes trabajos [14] donde se realiza control mediante visión monocular y [10], donde el objetivo fundamental es el control de un dron mediante el rostro humano.

Como parte de la innovación los algoritmos son implementados dentro de una Odroid XU4, la cual es una computadora embebida minuciosamente seleccionada, por ser capaz de lidiar con los cálculos computacionales tan demandantes que demandan los algoritmos de visión. Lo cual difiere completamente del trabajo realizado en [14] y [10].

Por otra parte a diferencia de los trabajos antes mencionados y de los cuales se han tomado ideas y principios, el presente trabajo se ha realizado en una plataforma libre que en comparación con estos; pues la que poseen es cerrada, la cual no puede ser dotada de sensor alguno o en general modificarla, ya que el desempeño del vehículo se ve comprometida. Es por eso que al dotarlo de una arquitectura libre es posible la manipulación del software y hardware. Así se puede observar que una plataforma libre trae muchos más beneficios que una cerrada.

Por su parte dotar al vehículo aéreo con una computadora embebida, no solo hace que los cálculos puedan llevarse a cabo a bordo en el vehículo, sino que además el flujo de datos es más directo en comparación con la transmisión inalámbrica, como es el caso de [10] y [14].

La resultante de lo anterior puede ser visto directamente cuando la ejecución de la tarea programada se lleva a cabo de manera satisfactoria, como se presentará más adelante en el capítulo número 8 cuya temática es la de exponer al lector los principales resultados generados durante la ejecución del seguimiento facial.

Pero para una mayor comprensión acerca del presente trabajo los siguientes capítulos tendrán el cometido de brindar los tópicos necesarios, para a comprensión del funcionamiento del presente proyecto.

CAPÍTULO 2

PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA SOLUCIÓN

El problema fundamental del proyecto radica en que al ser una plataforma completamente diferente al Parrot de AR Drone 2.0, el manejo tanto de sensores y datos, no puede manipularse de la misma manera que en Parrot. Lo cual no solo requiere de un diseño capaz de igualar las características de hardware que presenta este vehículo; sino también las de software.

Además considerando que la intención del presente trabajo es innovar los trabajos que le anteceden, como es caso del trabajo presentado [14], el cual fue realizado en Francia, y [10], los cuales requieren de una estación en tierra para su operación debido a que la plataforma es de código cerrado y es restringida la modificación del software, siendo este el motivo principal para buscar la eliminación de la estación en tierra como principal medio de computo para los algoritmos de visión, pero opcional

6CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA SOLUCIÓN

para supervisión.

El por que de la eliminación de la estación en tierra es debido a que el tiempo de transmisión de datos en estas aplicaciones aéreas es mas lenta que la cableada, pudiendo generar no solo un retardo en la señales, sino también mayor perdida de información, lo cual para efectos de control, entorpecen el funcionamiento del mismo, así como también de la tarea a realizar. Además de lo expresado anteriormente.

Dado que el Parrot es un vehículo aéreo cuya aplicación principal es de hobby, los controladores que están inmersos en el son de código cerrado, lo cual implica que no se puede llegar a programar al vehículo, limitado su función a un emisor de medidas, las cuales son provistas por su programación de fabrica.

Para solucionar esto, después de una ardua investigación se llevo a la conclusión que para poder resolver el problema de la estación en tierra la mejor opción es la implementación de una computadora embebida Odroid XU4, la cual es capaz de realizar procesos de visión artificial, debido a sus características, las cuales pueden ser encontradas en [\[1\]](#).

CAPÍTULO 3

MODELADO MATEMÁTICO DEL CUADRI-ROTOR

El presente capítulo es incluido ya que presenta una gran importancia, cuando de control se habla, ya que para llevar a cabo una tarea en un vehículo aéreo, es necesario saber como es el comportamiento del vehículo, es decir se debe encontrar una representación matemática, con la cual es posible conocer la interacción del medio en el cual estará expuesto el vehículo, con el fin de poder contrarrestar todo aquel comportamiento que interfiera en la realización de la tarea seleccionada.

Algunos ejemplos directos sobre tareas que requieren un control en base al modelo obtenido son: seguimiento de trayectorias trazadas, levantamiento de objetos, trabajos de cooperación (formaciones), la orientación y posición, siendo esta última en la cual se enfocara el presente trabajo de tesis.

Así pues para realizar el control de posición del cuadri-rotor, primeramente es necesario conocer el modelo matemático que emula el comportamiento del cuadri-rotor, con el firme propósito de emplear dicho modelo para así gobernar el movimiento del

vehículo. Modelo que será empleado en el capítulo 7 dedicado a la estrategia de control PD.

3.1 Diagrama de Cuerpo Libre de un Cuadri-rotor

De manera general un cuadri-rotor puede ser modelado de 2 metodologías diferentes ya sea desde la perspectiva de Euler- Lagrange o mediante la propuesta por Newton-Euler, ya que de esto dependerán las consideraciones físicas que tendrán relevancia.

Pero sin importar que criterio sea empleado el diagrama que contiene las relaciones físicas necesarias para un modelo aceptable puede ser encontrado en 3.1. Diagrama obtenido de [7] y que será empleado para las metodologías que a continuación se presentan.

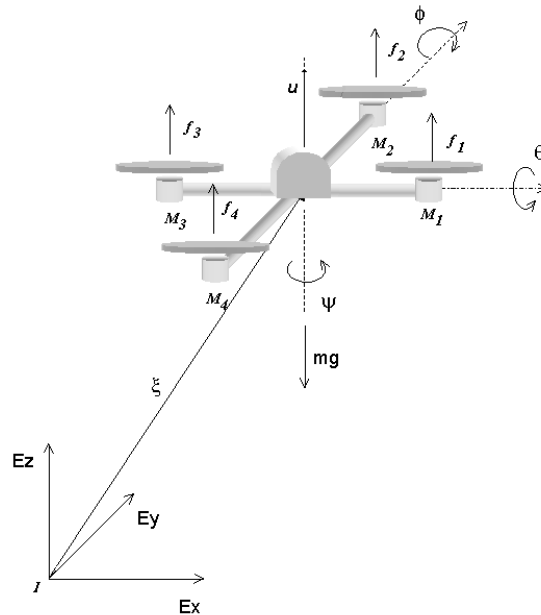


Figura 3.1: Diagrama de Cuerpo Libre

3.1.1 Modelo matemático mediante método Euler-Lagrange

En la presente subsección se muestra el modelo completo de un cuadri-rotor, el cual es obtenido a partir de la ecuaciones de Euler- Lagrange 3.1 para modelado de sistemas.

$$F = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} \quad (3.1)$$

Donde L representada por la suma de las energías cinéticas y potenciales, como puede verse en 3.2.

$$L = T_{tras} + T_{rot} - U \quad (3.2)$$

Con T_{tras} , T_{rot} y U definidos por 3.3.

$$\begin{bmatrix} T_{tras} \\ T_{rot} \\ U \end{bmatrix} = \begin{bmatrix} \frac{m}{2} \dot{\xi}^T \dot{\xi} \\ \frac{1}{2} \dot{\eta}^T J \dot{\eta} \\ mgz \end{bmatrix} \quad (3.3)$$

Dichas ecuaciones resultantes de 3.1, mas algunas otras consideraciones como son las fuerzas de coriolisis presentes en un cuerpo que posee movimiento y realizando un reacomodo podemos llegar a la ecuación 3.4.

$$\begin{bmatrix} F_1 \\ \tau_\eta \end{bmatrix} = \begin{bmatrix} m\ddot{\xi} & mg \\ J\ddot{\eta} & C(\eta, \dot{\eta})\dot{\eta} \end{bmatrix} \quad (3.4)$$

Donde:

1. F_1 es el vector de Fuerzas.
2. τ_η es el vector de Torques.
3. m es la masa de vehículo.
4. g es la constante de gravedad.
5. $\ddot{\xi}$ es el vector de aceleraciones traslacionales.
6. $\ddot{\xi}$ es el vector de aceleraciones rotacionales.
7. $C(\eta, \dot{\eta})\dot{\eta}$ son las fuerzas de coriolis del vehículo.

3.1.2 Modelo matemático mediante método Newton-Euler

El modelo siguiente obtenido a partir de las ecuaciones de Newton-Euler, parten del modelo de cuerpo libre, del cual se obtienen las fuerzas que interactúan en dicho modelo. Considerando dichas fuerza es posible llegar al modelo mostrado en 3.5, y que además puede ser encontrado en [6], el cual nuevamente surge a partir del diagrama de cuerpo libre mostrado en 3.1.

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} u(\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)) \\ u(\cos(\phi)\sin(\theta)\sin(\psi) - \cos(\psi)\sin(\phi)) \\ u\cos(\phi)\cos(\theta) - mg \\ \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\phi \end{bmatrix} \quad (3.5)$$

Donde:

1. ψ es el ángulo en yaw.
2. θ es el ángulo en pitch.
3. ϕ es el ángulo en roll.
4. $\ddot{\psi}$ es la aceleración en yaw.
5. $\ddot{\theta}$ es la aceleración en pitch.
6. $\ddot{\phi}$ es la aceleración en roll.
7. $\tilde{\tau}_\psi$ es el torques del eje yaw.
8. $\tilde{\tau}_\theta$ es el torques del eje pitch.
9. $\tilde{\tau}_\phi$ es el torques del eje roll.
10. m es la masa de vehículo.
11. g es la constante de gravedad.

12. u es la entrada del sistema.

No obstante el anterior modelo puede tener una variante mucho mas reducida como la que se presenta en [7], el cual surge de manera diferente debido a que la matriz de rotación que es aplicada al vector de fuerza es distinta, esto a causa de el orden en que se rota al rededor de los ángulos roll, pitch y yaw.

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} -u\sin(\theta) \\ u\cos(\theta)\sin(\phi) \\ u\cos(\theta)\cos(\phi) - mg \\ \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tau_\phi \end{bmatrix} \quad (3.6)$$

Para mayor información acerca de la obtención de dicho modelo se recomienda, como literatura alternativa [7], en el cual es posible encontrar tanto temas de modelado como de control en vehículos aéreos.

CAPÍTULO 4

PROTOTIPO EXPERIMENTAL

La presente sección esta enfocada al prototipo físico, el cual mediante la unión de diversos materiales comerciales se ha logrado la construcción de un cuadri-rotor capaz de realizar tareas referentes con visión por computador.

Para demostrar la importancia de dicho diseño en las siguientes secciones se abordaran los cambios que se tuvieron que implementar para el correcto funcionamiento. Así como también los sensores principales para el funcionamiento del mismo.

4.1 Sensores y Dispositivos

Con anterioridad ya se ha mencionado la propuesta de solución al problema que comprende el presente trabajo de tesis, así como la ventaja que se encuentra al usar una computadora embebida como lo es 4.1. Pero en la esta presente sección se profundizará más en las razones del por que la implementación de esta computadora embebida,

así como también de los sensores y dispositivos que han sido seleccionados para este proyecto.

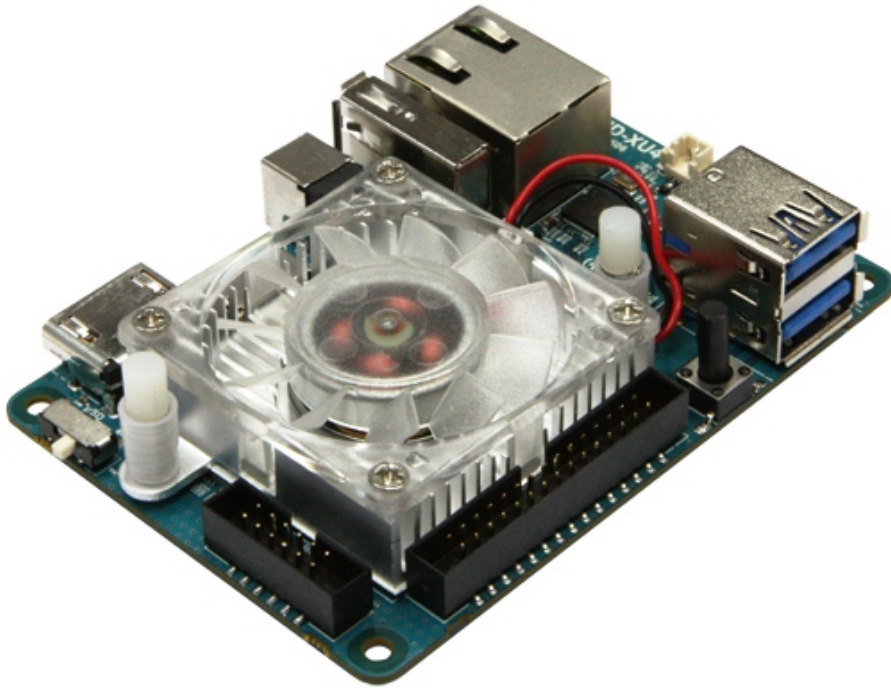


Figura 4.1: Computadora embebida Odroid XU4

Una de las características importantes de la computadora embebida Odroid XU4, es que compatible con varios sistemas operativos, pero principalmente con Linux un sistema operativo que posee la cualidad de ser de código abierto, dotando al que lo usa de una gran libertad de programación. Esta libertad permite trabajar con la librería de OpenCV, que a su vez es compatible con lenguajes de programación como C y Python, siendo este último el lenguaje a utilizar y el cual está integrado a Linux.

Otra característica destacable es que esta computadora posee bastantes periféricos como puede observarse en 4.2, pero principalmente se puede comunicar de manera serial,

lo cual mas adelante sera de gran utilidad. Además de tener 3 puertos USB de gran velocidad de transmisión de datos.

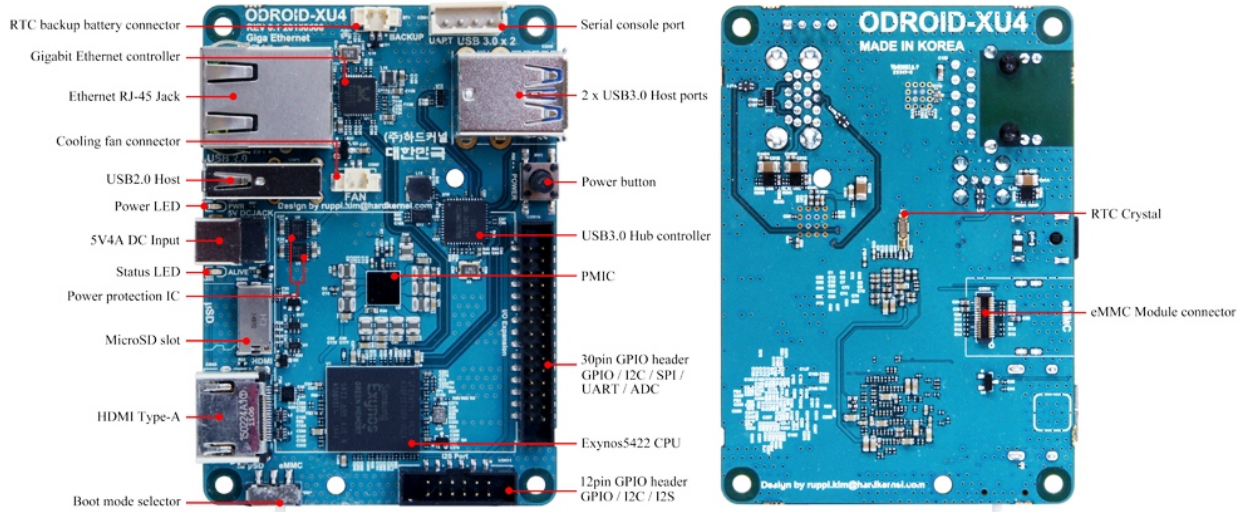


Figura 4.2: Esquema Computadora embebida Odroid XU4

Respecto al sensores de visión artificial, ya que el prototipo que se tiene, al ser genérico, carece de algún sensor de visión por lo que en base a una extenuante investigación, se concluyo que el uso de cámaras con efecto ojo de pez, son una excelente opción ya que este tipo de cámaras permite tener un ángulo de visión mucho mas amplio de un escenario, para si tener un mayor índice de detecciones, aun cuando se presenta como efecto secundario una pequeña distorsión como se aprecia en 4.3.

Dada la ventaja que proporcionan las cámaras con ojo de pez se ha seleccionado una con conexión USB como se puede observar en 4.4, la cual posee las cualidades de ajustarse a cambio de intensidad luminosa, bajo costo, compatibilidad con sistemas



Figura 4.3: Distorsión Cámara Ojo de Pez

basados en Linux, y principalmente un bajo consumo de energía y peso. Parámetros que intervienen para un tiempo de operación mayor en comparación con otros prototipos.



Figura 4.4: Cámara Ojo de Pez Seleccionada

Además para realizar el envío de datos de control se realizó la implementación de un adaptador USB a serial de tecnología TTL 4.5, el cual posee un peso muy bajo, lo cual es de apoyo al sistema.

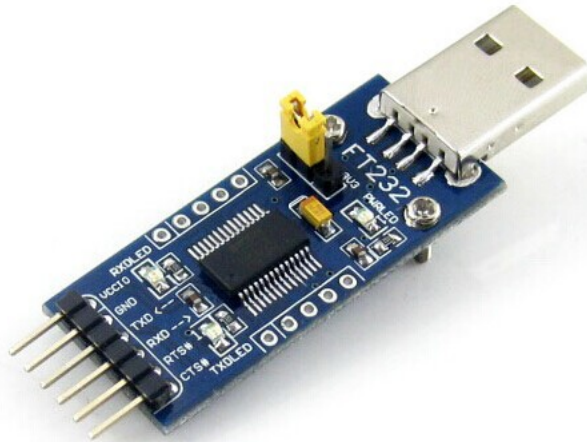


Figura 4.5: Cámara Ojo de Pez Seleccionada

Por su parte para llevar a cabo el control del desplazamiento del vehículo aéreo se propone una plataforma Pixhawk compatible, similar a 4.6, basada en programación en C, el cual está encargado de las manipulaciones de los motores en base a los datos recibidos por la computadora Odroid XU4. Dicha plataforma tiene sensores internos como son magnetómetros, giróscopos y acelerómetros, así como también accesorios adicionales para la interacción con el medio ambiente, tales como sensor de flujo óptico y ultrasónico, telemetría, GPS, entre otros.



Figura 4.6: Autopiloto Pixhawk

Finalmente los elementos descritos anteriormente en conjunción con una estructura general para un cuadri-rotor, un auto piloto pixhawk con múltiples sensores y mediante un rediseño, se propone un prototipo capaz de realizar la detección, seguimiento de un rostro mediante una plataforma basada en código abierto.

4.2 Estructura Original y Rediseño

Una vez revisados los dispositivos y sensores a implementar el siguiente reto a cumplir fue la distribución y acomodo de los mismos dentro de una estructura.

Por lo que partiendo de este hecho la estructura con la que se trabajó es producto de un rediseño, ya que se tiene como base una estructura de un cuadri-rotor comercial; cuyas características principales pueden ser encontradas en [5]; la cual se ha sido modificada, para albergar, los sensores antes mencionados dentro de los que destacan: una cámara USB 4.3, una computadora embebida (Odroid XU4)4.1, que son componentes necesarios para llevar a cabo tareas de visión artificial, y que a diferencia del autopiloto 4.6 no poseen un lugar determinado.

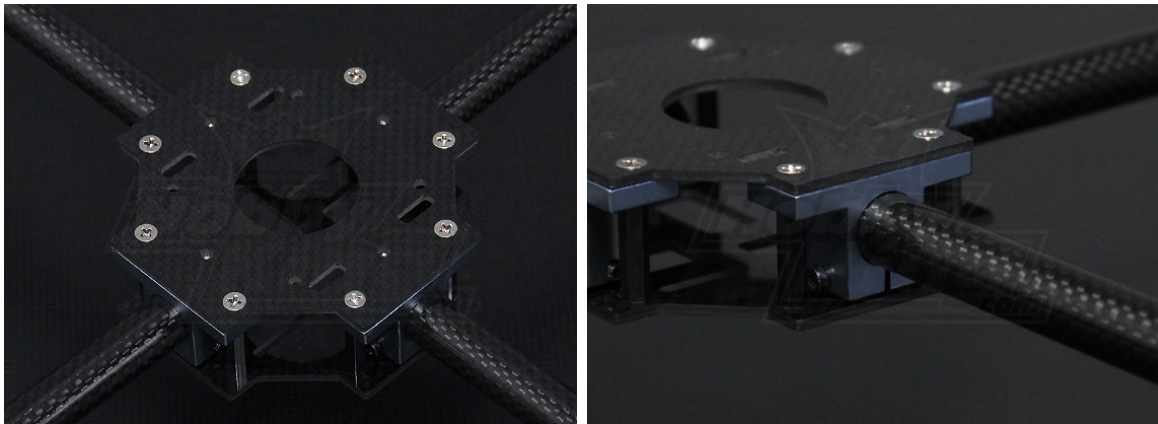
Debido a esto el cambio principal del prototipo recae en una segunda planta adicionada a la estructura original 4.7, además de un reforzamiento de la misma, así como el diseño de conexiones para su respectiva integración.



Figura 4.7: Estructura comercial.

Aunque es preciso mencionar que dichas mejoras tuvieron más de una versión previa a las presentadas. Para ilustrar esto se mostrarán algunas de las versiones anteriores como evidencia de la transformación intermedia a la presentada. Por lo que se mencionaran a razones principales por las cuales dichos cambios fueron hechos.

Con respecto a la segunda planta implementada, fue agregada a la estructura original, ya que el espacio de la estructura comercial, es insuficiente como se puede ver en 4.7. Y ya que la computadora embebida Odroid XU4 ocupa un espacio similar al autopiloto, la ubicación de ambos dispositivos en el mismo espacio es imposible.



(a) Estructura Vista Frontal

(b) Base vista superior

Figura 4.8: Base estructura comercial

Por otra parte el reforzamiento de la estructura fue hecho en dos partes principalmente, en la base central 4.8 y en los soportes para los motores 4.9, esto debido a que los vehículos aéreos hasta ahora desarrollados, por efectos de vibración en pleno vuelo, los tornillos tienden a sufrir un desajuste. El cual principalmente afecta a la parte que sujeta los motores a la plataforma, pues usualmente causa una rotación en 4.9, lo cual puede provocar en un accidente.



Figura 4.9: Brazo Estructura

Además de las modificaciones anteriores se diseñaron y construyeron las sujeciones pertinentes para soportar de manera adecuada a la segunda planta. Estas sujeciones se hicieron para albergar una distancia de 6 centímetros entre las plantas. Esta característica no solo permite la libre conexión de los sensores para el autopiloto y la computadora embebida, así como su interconexión, sino también la instalación de la cámara.

Por su parte tanto la base para la fijación de la cámara 4.10, fue llevada a cabo mediante el diseño y fabricación de piezas, las cuales armonizan con el prototipo, aprovechando en gran medida la estructura del vehículo.

Además puede apreciarse en 4.10, en la parte del acrílico blanco un modulo de flujo óptico que también puede ser agregado, debido al diseño implementado, ya que se pretende emplearlo a futuro.

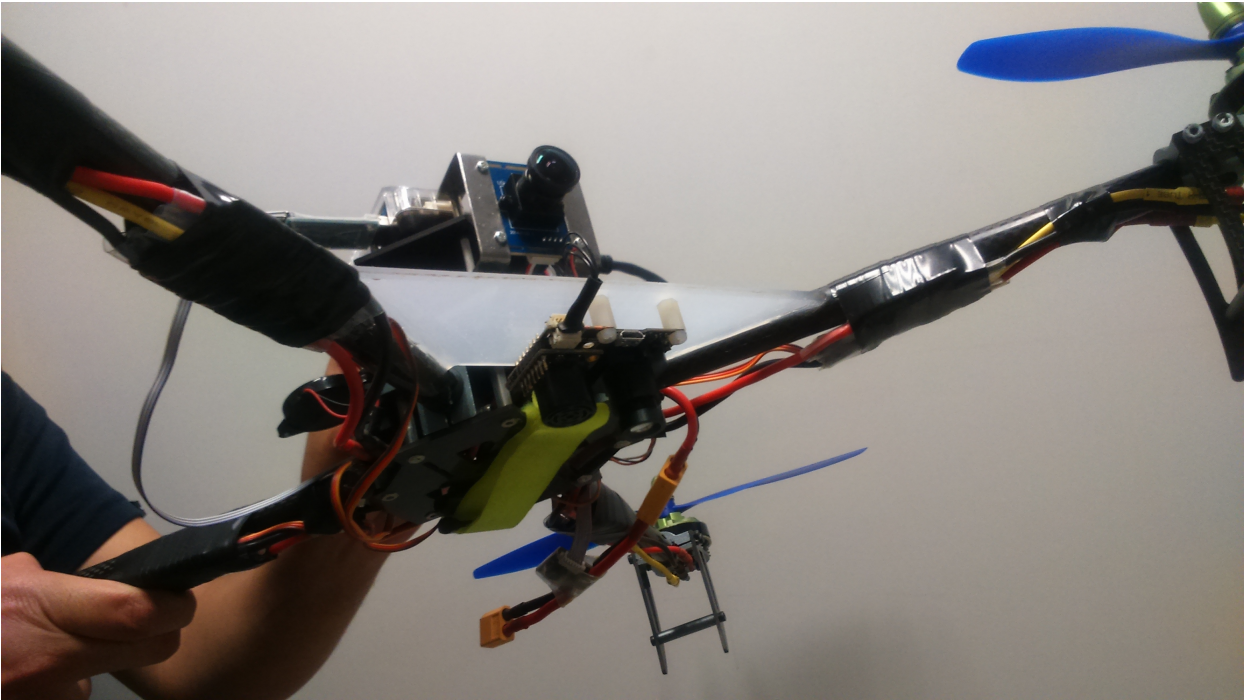


Figura 4.10: Fijación Cámara

4.3 Configuración de los Motores Plataforma Experimental

Como la principal tarea cuando de vehículos aéreos se habla, es de gran importancia elegir el tipo de arreglo con que se llevara a cabo el ajuste del movimiento en pleno vuelo, para ello tras una extensa búsqueda tanto de configuraciones como de ventajas y desventajas de las mismas. Siendo seleccionada de las dos configuraciones encontradas la de tipo x, ya que presenta un mejor control al trabajar con 2 motores para los 4 principales movimientos que se realizan al maniobrar, los cuales son dirección frontal, lateral y trasera.

Dichas configuraciones pueden ser vistas en 4.3, imagen que fue obtenida de [3].

4.3. CONFIGURACIÓN DE LOS MOTORES PLATAFORMA EXPERIMENTAL23

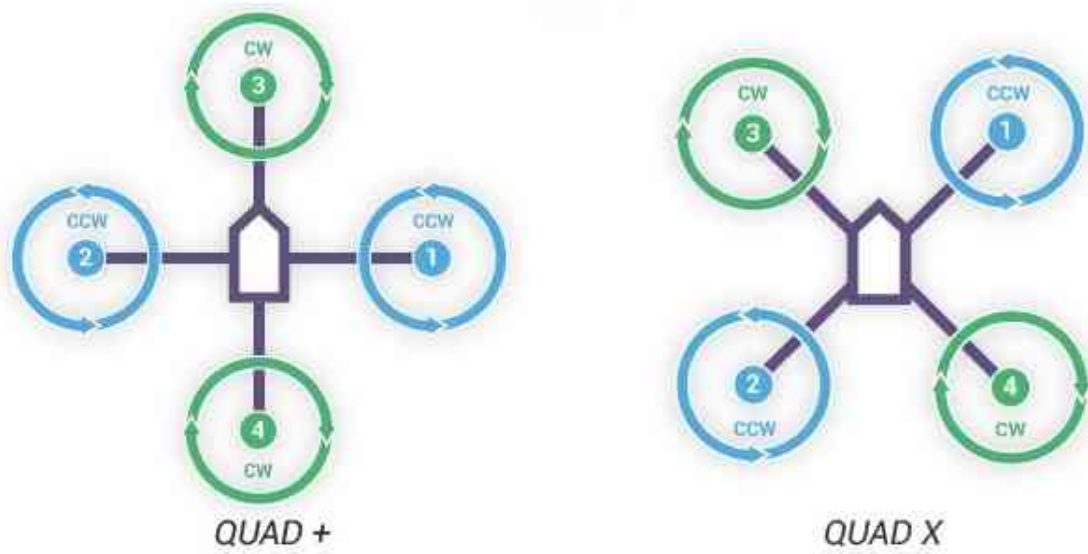


Figura 4.11: Tipos de Configuraciones

Donde los giros de las hélices se clasifican de acuerdo al tipo de giro ya sea en dirección horaria o antihoraria, movimiento tomado en base al realizado por el reloj. Prueba del movimiento de los giros es encontrado en 4.3, también obtenida de [3].



Figura 4.12: Tipos de giros para las hélices

4.4 Plataforma experimental final

Hasta ahora en la sección anterior solo se ha hablado de las partes que fueron modificadas en la estructura comercial explicando claramente la razón y beneficio para el prototipo. Esta sección tiene como finalidad mostrar la plataforma final. Además de algunas versiones anteriores.

Una de las versiones durante el proyecto puede observarse en 4.13 y 4.14, caracterizada por una segunda planta de fibra de carbono, diseñada para contener a la computadora embebida, cuyo defecto es que carece de espacio para llevar el dispositivo de seguridad para la interacción humano cuadri-rotor, y dado que el presente proyecto interactúa de manera mas estrecha con humanos, debido a que trabaja con rostros de manera cercana, la seguridad es un factor de prioridad máxima, por lo que esta segunda plataforma tuvo que ser replanteada y eventualmente remplazada por otra.

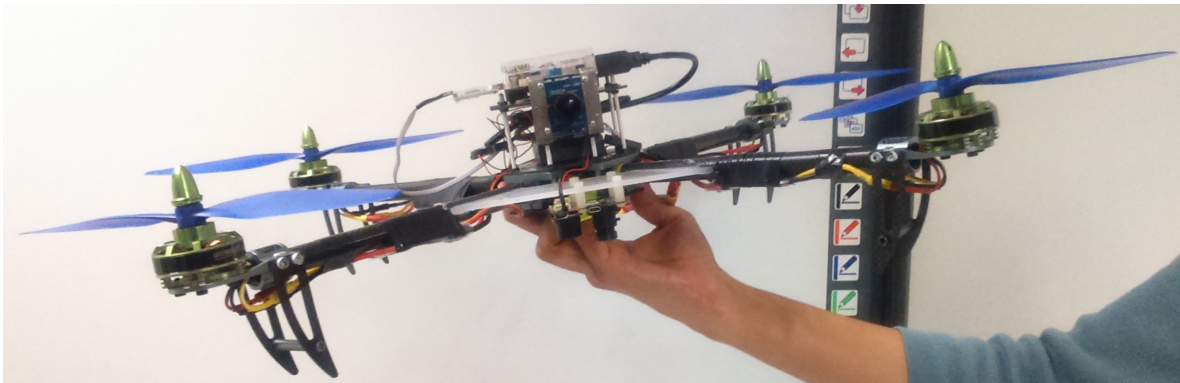


Figura 4.13: Versión Previa Vista Frontal

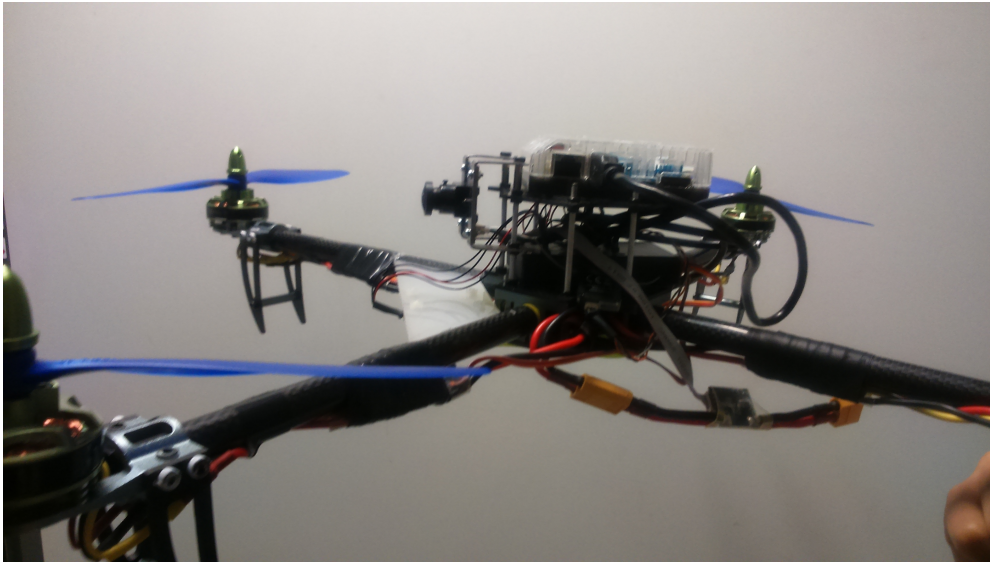


Figura 4.14: Versión Previa Vista Lateral

Como parte de la tercera versión se implemento el dispositivo para manipulación de seguridad el cual permite al usuario poder mantenerlo con las manos aun cuando esta en vuelo a baja altura, donde su principal utilidad es mayormente apreciado cuando se requieren realizar pruebas en las que se desee observar el comportamiento del cuadri-rotor sin poner en riesgo la integridad física. Aunque pese a estar presente este dispositivo de seguridad, se sugiere el apoyo de guantes y lentes de protección, como equipo complementario.

Finalmente 4.15 corresponde a la versión final del prototipo, la cual consta de un refuerzo del sistema de manipulación, mediante fibra de carbono, la cual le añade mayor resistencia y estabilidad a la estructura. Cabe mencionar que este recubrimiento se hizo mediante la ayuda de Laura Guadalupe Martinez Santos, quien posee los conocimientos para el manejo de la fibra de carbono para el reforzamiento de estructuras.



Figura 4.15: Versión 3 Final

CAPÍTULO 5

ALGORITMOS DE VISIÓN ARTTIFICIAL

El capítulo actual tiene la intención de presentarlos algoritmos necesarios para llevar a cabo la detección, seguimiento, estimación de distancia al usuario, que son embarcados en la computadora embebida 4.1, cuyas características según [1], son las adecuadas para realizar las tareas antes mencionadas.

Dado que estos procesos son diferentes, para su correcta comprensión cada uno de ellos se abordará de manera individual, no sin antes mencionar que dichos procesos están relacionados estrechamente, debido al objetivo del proyecto.

Es decir por ejemplo, para poder estimar la distancia entre el vehículo, es preciso conocer la detección del rostro, pues en base a estos parámetros arrojados por este algoritmo es posible obtener ecuaciones que gobiernen al vehículo aéreo.

5.1 Detección de Rostros

Para llevar a cabo la detección de los rostros se seleccionó un clasificador de tipo Haar, el cual esta basado en el algoritmo Piramidal, el cual fue diseñado por Paul Viola y Michael Jones [15]. Dicho algoritmo hace uso de un conjunto de patrones que son guardados en un archivo, en el cual existen datos que pertenecen a una compilación de imágenes con objetos de interés, además es posible ajustarlo para detectar cualquier objeto que deseemos, tan solo es necesario entrenarlo mediante haar training. Lo cual hace de este algoritmo una poderosa herramienta para la detección aun cuando es sometido ante escalamiento del objeto de interés.

Para entrenarlo es necesario tener una compilación de imágenes donde el objeto que deseamos detectar no se encuentre y otra donde si, estas ultimas deben tener marcado el objeto, ya que estas dimensiones serán las mediciones útiles mas adelante. Además estas imágenes deben ser de diferentes escenarios y tamaños.

Es preciso decir que dado el entrenamiento este algoritmo tiene la propiedad de ser robusto ante cambios de tamaño del objeto de interés lo cual proporciona una ventaja al proyecto en cuestión.

Como el objetivo principal es el seguimiento de la detección frontal de un rostro se implemento una detección facial por medio de algoritmo haar, el cual se ha descrito con anterioridad.

Además de la detección y con el fin de corroborar las detecciones en video se ha determinado que el perfil tenga un color especifico, complementado con un identificador que también es capaz de detectar la ausencia de rostros en el escenario.

De manera general la detección de rostros queda expresada de la siguiente manera:

1. Cara frontal identificado con color verde.
2. Sin Rostros en el escenario sin color

Es de suma importancia mencionar que los parámetros que son utilizados para identificar las detecciones del rostro son: los puntos iniciales donde el rostro ha sido detectado, así como los valores referentes a la altura y ancho del rostro. Datos con los que es posible enmarcar el rostro detectado por una gran cantidad de formas geométricas, tales como cuadros círculos o formas personalizadas. Para efectos prácticos solo se ha utilizado el cuadrado como marcador de rostro.

Muestra de la detección frontal puede observarse en 5.1, donde el cuadro de color verde, enmarca a la cara detectada.

Como dato adicional otra opción de detección que puede ser empleada a futuro, es basarse en el algoritmo que se basa a puntos el característicos, realizador por el MIT [9].

5.2 Seguimiento de Rostros

Con respecto al seguimiento de rostros, es de suma importancia para el controlador que mas adelante se expondrá, el poder conocer tanto las posiciones como velocidades del objeto detectado. Pero dado que la cámara solo puede brindarnos mediciones de posición relativas a la misma, se representa una limitante de primera instancia. Limitación que se resolvió implementando un filtro de Kalman [4] inventado por Rudolf



Figura 5.1: Filtro de Kalman Velocidades

E. Kalman en 1960, cuya función es la de estimar cada uno de las posiciones y velocidades de nuestro sistema en un tiempo de muestreo dado, teniendo incluido un factor de incertidumbre.

De manera general el filtro de kalman puede considerarse bajo dos acciones una parte predictora y una correctora, como puede observarse en 5.2.

Donde A Y B son los parámetros de nuestro sistema, particularmente en nuestro caso B es 0 ya que no es contemplada la señal de control. Q es la matriz de covarianza la cual se encargara de darle peso a las mediciones del modelo o de la predicción.

Es importante decir que los datos de entrada para este filtro son solo los estados

Ecuaciones de predicción

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^H + \mathbf{Q}$$

Ecuaciones de corrección

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^H (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^H + \mathbf{R})^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (z_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$$

Figura 5.2: Filtro de Kalman Ecuaciones

que podemos cuantificar. Es decir solo pueden ser ingresados las posiciones de nuestro sistema que obtenemos del detector.

Otro dato importante en el algoritmo es el color negro asignado para el recuadro que representa las posiciones estimadas, el cual aparece en los videos recopilados durante la ejecución.

CAPÍTULO 6

COMUNICACIÓN

Hasta ahora a lo largo del presente trabajo solo se ha mostrado aquello referente a la obtención y manejo de datos, tales que el objetivo planteado en un principio se cumpla, pero sin embargo otra cosa de gran importancia es la interacción entre el auto piloto y la computadora embebida, para efectos de control. Por lo que el presente capítulo hace referencia a dicha interacción que existe entre a ambos.

Por lo que primeramente se especificara cuales son los parámetros relevantes para el control de posición, entiéndase por relevantes, todos aquellos parámetros que son necesarios para la ley de control PD.

Posterior a esto se hablará acerca del protocolo de comunicación diseñado, el cual es especial para este proyecto.

6.1 Parámetros de Control

Teniendo en mente que el control de posición a realizar es uno de tipo PD, los parámetros necesarios para ponerlo en operación son los errores en posición y velocidad, los cuales consideran, la relación entre el valor deseado y el valor registrado a lo largo de la ejecución del programa.

Por lo que esta sección tiene como cometido dar a conocer los parámetros que son requeridos para poder gobernar el comportamiento del posicionamiento del cuadri-rotor.

De manera general los errores de posición necesarios para el control están dados por 6.1, y los de velocidad en 6.2, tablas en las cuales se muestran tantas variables de control como de ejes a controlar.

Tabla 6.1: Parámetros de Error en Posición del Cuadri-rotor

Parámetro	Nomenclatura	Valor Deseado	Unidades
Error Posición Eje Z	e_z	Centro Imagen	cm
Error Posición Eje Y	e_y	Centro Imagen	cm
Error Posición Eje X	e_x	120	cm

Tabla 6.2: Parámetros de Error en Posición del Cuadri-rotor

Parámetro	Nomenclatura	Valor Deseado	Unidades
Error Velocidad Eje Z	\dot{e}_z	0	$\frac{cm}{s}$
Error Velocidad Eje Y	\dot{e}_y	0	$\frac{cm}{s}$
Error Velocidad Eje X	\dot{e}_x	0	$\frac{cm}{s}$

6.2 Protocolo Envío

Una vez identificado los datos necesarios para el funcionamiento de esta aplicación, la presente sección tiene como finalidad mostrar el protocolo empleado para el envío y recepción de datos.

Datos que gobiernan el comportamiento del Cuadri-rotor, mediante el autopiloto, que a su vez controlara la cantidad energía suministrada a los motores, los cuales gestionan el posicionamiento del vehículo.

Para comenzar es importante mencionar que se realizo una codificación especial para los datos, con propósito de enviarlos al autopiloto.

Esta codificación tuvo que ser implementada debido a que los datos negativos que son obtenidos por los datos de error, requieren una cantidad mas grande de procesamiento y ya que el envío de datos positivos es una forma mas simple de envío, la mejor opción fue la de implementar este tipo de proceso a los datos.

Y es así que gracias a esta codificación la cantidad de datos que se ha logrado enviar es de 16 datos y probado hasta 18 datos de manera satisfactoria.

La codificación tiene 2 seguros incluidos los cuales tienen como finalidad asegurar la detección del dato correcto.

Por ultimo ya enviados los datos, la recepción se ha realizado mediante un algoritmo de decodificación programado en el autopiloto el cual es capaz de recibir satisfactoriamente los 16 datos de información.

Los datos enviados pueden ser vistos con mas detalle en 6.3, tomando en cuenta

que por cada parámetro que se muestra en esta tabla debe considerarse como 2 datos exceptuando los primeros 2, cuya función es la de indicar el dato de importancia.

Tabla 6.3: Datos Enviados

Parámetro	Nomenclatura
Inicio trama	<i>dataident</i>
Detección Rostro	<i>sideface</i>
Error Posición Eje Z	e_z
Error Posición Eje Y	e_y
Error Posición Eje X	e_x
Error Velocidad Eje Z	\dot{e}_z
Error Velocidad Eje Y	\dot{e}_y
Error Velocidad Eje X	\dot{e}_x
Checksum	chksum

CAPÍTULO 7

ESTRATEGIA DE CONTROL

Una vez revisado los tópicos pertinentes al rediseño estructural, los algoritmos de visión artificial, así como el de comunicación, lo restante por abordar es la técnica de control a emplear con el propósito de supervisar el seguimiento del rostro.

Cabe mencionar que para realizar el seguimiento, es necesario tener un control de orientación, ya que es el requerimiento básico para poder despegar y tener un vuelo estable. Por lo anterior en concreto el proyecto consta de dos controles, uno para el de orientación como se menciono anteriormente y otro para el seguimiento del rostro, ambos en conjunto interactúan, para realizar el objetivo planteado.

Para dejar mas en claro lo anterior las secciones que a continuación se presentan, no solo tienen como finalidad el explicar su funcionamiento, sino también profundizar en la importancia de su uso.

7.1 Control de Orientación

Esta sección tiene como finalidad, mostrar el control que se ha realizado para la estabilización de un cuadri-rotor. Para lo cual se parte de las ecuaciones que resultan del modelado de Newton-Euler.

Es preciso decir que el control implementado en el auto-piloto es un control de tipo PD el cual actúa directamente con las medidas que son obtenidas de la central inercial que esta contenida en el auto-piloto. Es decir el control es ingresado en los torques(momentos) los cuales son representados por τ_ψ , τ_θ y τ_ϕ en 3.5, los cuales están relacionados con las aceleraciones angulares, que a su vez, modifican el comportamiento de los desplazamientos traslacionales.

Con lo anterior se consigue orientar el vehículo en posición angular cero respecto a cada eje, lo cual en términos técnicos es lo que se denomina estabilidad.

Para ilustrar como operan estas orientaciones surge 7.1, así pues el movimiento de roll esta asignado a la rotación que el vehículo hace de manera lateral, por su parte el giro en dirección frontal o trasera pertenece al ángulo de pitch y por último a rotación sobre el mismo eje se denomina yaw.

Lo anterior es en base a que el frente del Cuadri-rotor es la referencia, para la identificación de los ejes de rotación. Además en 7.1, los signos que se muestran pueden considerarse de manera inversa, dependiendo la aplicación.

Más aun si de estabilidad se habla es conveniente decir que no solo estas rotaciones deben ser consideradas, sino también las fuerzas que mantendrán la sustentación por lo que de manera mas detallada se muestra 7.1.



(a) Ángulo de Rotación en Roll



(b) Ángulo de Rotación en Pitch



(c) Ángulo de Rotación en Yaw

Figura 7.1: Ángulos de Rotación

En otras palabras la interacción en conjunto de las fuerzas y momentos pueden visualizarse de manera mas concreta en 7.2, ya que en esta imagen es posible percibir estas relaciones físicas que intervienen en el cuadri-rotor.

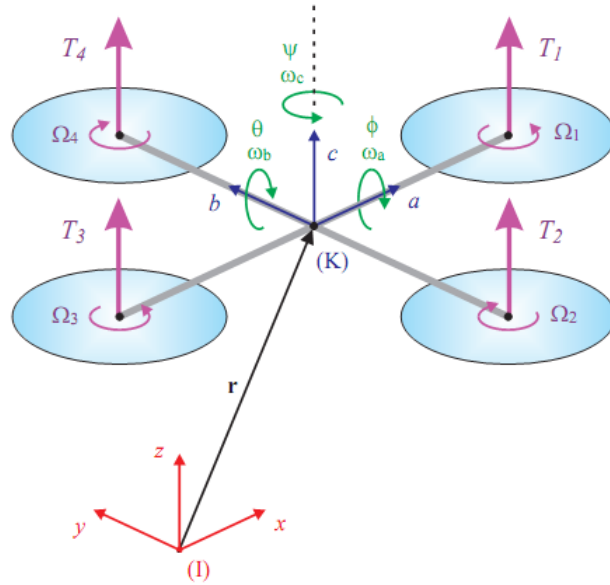


Figura 7.2: Dinámica Completa

7.1.1 Desarrollo Estrategia de Control

Como tópico adicional se agrega el proceso del control para acreditar estabilidad para vehículos miniatura propuesta por [6], la cual surge al desarrollar el control del cuadricóptero.

Ahora partiendo de la ecuación 3.5, vista en el capítulo 3 se procede a proponer a u como una retroalimentación de estado 7.1 para la parte correspondiente a la altitud. Donde 7.1 surge a partir del despeje del mismo subsistema.

$$u = \frac{1}{\cos \phi \cos \theta} (r_1 + mg) \quad (7.1)$$

Donde r_1 es la señal de control deseada descrita en 7.2 y 7.3, donde podemos observar

un controlador de tipo PD para el eje z y el ángulo de yaw, donde los errores se definen como $e_{pz} = z - z_d$ y $e_{p\psi} = \psi - \psi_d$.

$$r1 = -k_{pz}e_{pz} - k_{vz}\dot{z} \quad (7.2)$$

$$\tau_\psi = -k_{p\psi}e_{p\psi} - k_{v\psi}\dot{\psi} \quad (7.3)$$

Posterior a esto el control de altura encontrado es sustituido en los ejes traslacionales restantes, es decir en los ejes x , y . Quedando como 7.4 y 7.5 respectivamente.

$$m\ddot{x} = (r1 + mg) \left(\frac{\sin \psi \tan \phi}{\cos \theta} + \cos \psi \tan \theta \right) \quad (7.4)$$

$$m\ddot{y} = (r1 + mg) \left(\sin \psi \tan \theta - \frac{\cos \psi \tan \phi}{\cos \theta} \right) \quad (7.5)$$

De estas estrategias implementadas en z y ψ , la literatura recomienda la selección de los parámetros k_{pz} , k_{vz} , $k_{p\psi}$ y $k_{v\psi}$, ganancias encargadas de la penalización de la intensidad en la respuesta del sistema en estos dos subsistemas.

Dado que al llegar a las posiciones deseadas en z_d y ψ_d y al ser muy pequeños la ecuación 7.4 y 7.5 pueden reescribirse como 7.6 y 7.7.

$$\ddot{x} = g \tan \theta \quad (7.6)$$

$$\ddot{y} = -g \frac{\tan \phi}{\cos \theta} \quad (7.7)$$

Hecho esto ahora se supone que el ángulo θ es muy pequeño, se obtiene así el subsistema x, \dot{x} como se muestran en 7.8.

$$\begin{aligned} \ddot{x} &= g\theta \\ \ddot{\theta} &= \tau_\theta \end{aligned} \quad (7.8)$$

Después de una manipulación en espacio de estados es posible obtener .

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ \tau_\theta \end{bmatrix} \quad (7.9)$$

Integrando a este espacio de estados una referencia que deseamos alcanzar se puede reescribir el 7.9 como 7.10.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3^{ref} + \tilde{x}_3 \\ x_4 \\ \tau_\theta \end{bmatrix} \quad (7.10)$$

Para $x_3^{ref} = \theta_{ref}$ y $\tilde{x}_3 = \theta - \theta_{ref}$, por lo que ahora se define el una función 7.11 en base a los parámetros x y \dot{x} , que representan posición y velocidad.

$$\theta_{ref} = V(x, \dot{x}) \quad (7.11)$$

Por lo que un control de tipo PD puede ser definido para quedar como 7.14.

$$\tau_{\theta} = -k_{p\theta} (\theta - \theta_{ref}) - k_{v\theta} \dot{\theta} \quad (7.12)$$

Por ultimo considerando el ángulo ψ cercano a cero el subsistema comprendido por y y ϕ

$$\begin{aligned} \ddot{y} &= -g\phi \\ \ddot{\phi} &= \tau_{\phi} \end{aligned} \quad (7.13)$$

Y de manera similar al anterior se puede obtener el siguiente control

$$\tau_{\theta} = -k_{p\phi} (\phi - \phi_{ref}) - k_{v\phi} \dot{\phi} \quad (7.14)$$

Lo cual completa los 4 controles necesarios para el funcionamiento del cuadri-rotor. Como nota principal el correcto funcionamiento es adecuado siempre y cuando las ganancias $k_{p\phi}$, $k_{v\phi}$, $k_{p\theta}$ $k_{v\theta}$ sean de tipo Hurwitz tal que se cumpla la ecuación $s^2 + k_v s + k_p$ para cada par de parámetros por ángulo.

7.2 Control PD para Seguimiento Facial

En la presente sección se muestran los resultados obtenidos al unir los controles de orientación e identificación, los cuales resultan en el seguimiento de un rostro humano.

Ya que una imagen es una representación bidimensional y es captada de un mundo tridimensional se presenta una perdida de una dimensión la cual representa la profundidad. Para ello se ha implementado una interpolación. La cual relaciona el tamaño del cuadro identificado con la distancia que este presenta.

Respecto a las variables Z y Y , estas representan la altura y el desplazamiento lateral respectivamente en el marco de la cámara.

7.2.1 Desarrollo Estrategia de Control

De manera homologa a lo presentado en 3.5 y tomando la demostración en [8], se parte del modelo expuesto en el capítulo 3 linealización del sistema alrededor de $\theta_d \approx \phi_d \approx 0$ para el modelo traslacional del cuadri-rotor que es lo que se desea controlar.

Al realizar la linealización es posible representar la parte traslacional en x y en y de la manera .

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \approx u \begin{bmatrix} \sin \psi & \cos \psi \\ -\cos \psi & \sin \psi \end{bmatrix} \begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} \quad (7.15)$$

Y si se tiene que $\phi \approx \phi_d$ y $\theta \approx \theta_d$ la ecuación 7.15, queda modificada de la siguiente manera 7.16.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \approx \frac{u}{m} \begin{bmatrix} \hat{\phi} \\ \hat{\theta} \end{bmatrix} \quad (7.16)$$

Ahora en vase a estas nuevas variables $\hat{\phi}$ y $\hat{\theta}$, se definen los controles virtuales que gobernarán estos ejes lo cual puede igualarse a las posiciones deseadas como se muestra en .

$$\begin{bmatrix} \hat{\phi} \\ \hat{\theta} \end{bmatrix} \approx \frac{m}{u} \begin{bmatrix} -k_{ptx}e_{ptx} - k_{vx}\dot{e}_{ptx} \\ -k_{pty}e_{pty} - k_{vy}\dot{e}_{pty} \end{bmatrix} \quad (7.17)$$

Donde e_{ptx} representa la relación entre el valor deseado y el obtenido en ejecución de la parte traslacional de manera que se tiene lo siguiente 7.18 y 7.19.

$$\begin{aligned} e_{ptx} &= x - x_d \\ e_{\dot{p}tx} &= \dot{x} - \dot{x}_d \end{aligned} \quad (7.18)$$

$$\begin{aligned} e_{pty} &= y - y_d \\ e_{\dot{p}ty} &= \dot{y} - \dot{y}_d \end{aligned} \quad (7.19)$$

La única condición que deben cumplir las constantes $k_p > 0$, $k_v > 0$. Dicho esto se transforma a las coordenadas originales 7.20.

$$\begin{bmatrix} \hat{\phi} \\ \hat{\theta} \end{bmatrix} \approx \frac{m}{u} \begin{bmatrix} \sin \psi & \cos \psi \\ -\cos \psi & \sin \psi \end{bmatrix} \begin{bmatrix} -k_{ptx}e_{ptx} - k_{vx}\dot{e}_{ptx} \\ -k_{pty}e_{pty} - k_{vy}\dot{e}_{pty} \end{bmatrix} \quad (7.20)$$

Cabe mencionar que para el eje en z también se emplea un controlador de tipo PD el cual esta dado por ??

$$\ddot{z} = -k_{ptz}e_{ptz} - k_{vz}\dot{e}_{ptz} \quad (7.21)$$

Siendo este ultimo el control que complementa el vector $(x_d, y_d, z_d)^T$.

CAPÍTULO 8

RESULTADOS

EXPERIMENTALES

El capítulo actual esta dedicado a presentar los resultados que se han obtenido del presente trabajo. Dentro de los cuales se encuentran los comportamientos de los algoritmos de calibración implementados, los errores en posición y velocidad, provenientes de las mediciones y estimaciones del algoritmo de filtro de Kalman respectivamente.

También se muestran de los datos almacenados dentro del autopiloto, datos que no solo son calculados en su interior, sino también recibidos desde la computadora embebida.

Además se presenta el desempeño en la ley de control aplicada, la cual artera consigo el comportamiento de los motores.

8.1 Resultados Computadora Embebida (Odroid XU4)

8.1.1 Calibración de Distancia a Objeto

Hasta ahora solo se ha mostrado el material necesario para el control de un plano YZ, el cual comprende los ejes correspondientes a la altura y desplazamiento lateral. Con el fin de proteger la integridad del usuario, se ha implementado una calibración para poder estimar la distancia entre el usuario y el cuadri-rotor, con lo cual se completa un control en el eje X, lo que representa un control en un plano tridimensional.

El proceso de esta calibración puede ser observado en 8.1, donde además se puede observar que el caso mas drástico para llevarse a cabo este proceso es en el exterior y a diferente hora del día.



Figura 8.1: Calibración Cámara

Además es preciso decir que al implementar este control adicional puede manipularse el alejamiento o acercamiento del cuadri-rotor dependiendo si la ubicación actual es menor o mayor a la establecida.

Es preciso mencionar que dicha calibración posee un rango de operación entre 0 a 1.90 m. Para observar dicho comportamiento se puede agregar 8.2, gráfica en la cual es posible observar que es posible determinar la distancia a la cual se encuentra la persona siempre y cuando este dentro del rango de operación ya antes mencionado.

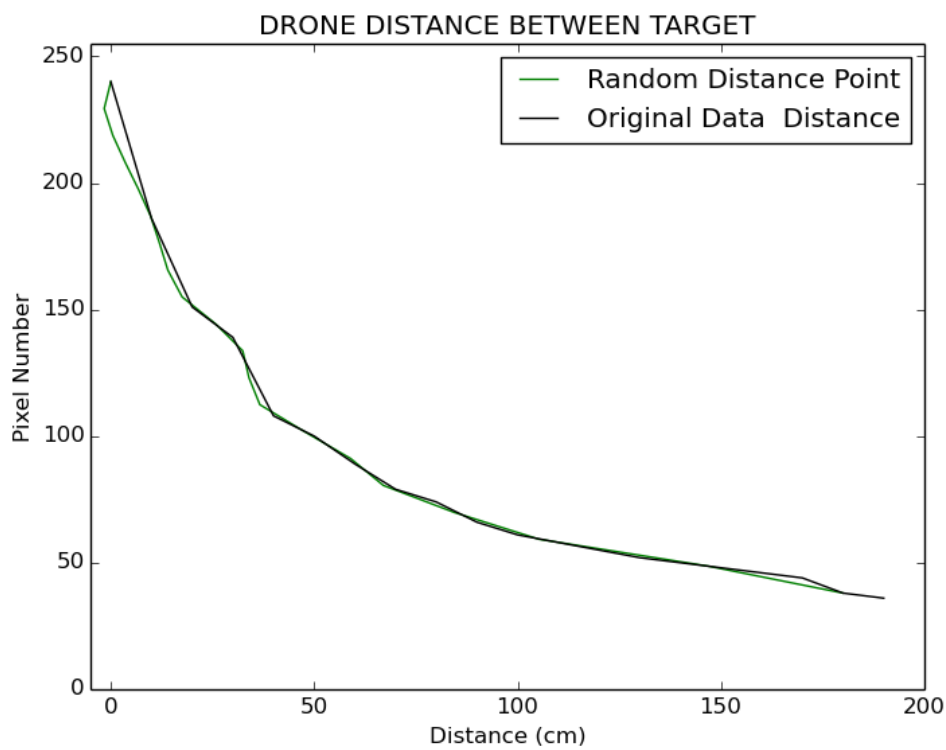


Figura 8.2: Calibración Cámara Distancia a Persona

8.1.2 Calibración del Desplazamiento Lateral del Objetivo

En la presente subsección se muestra la calibración de la cámara que se ha realizado en base a 2 parámetros conocidos, donde el primer parametro conocido es la dimensión de un objeto, y posteriormente la distancia a ese objeto, calculada en la subsección anterior, todo esto con el propósito de conocer la distancia en cm, debido al desplazamiento lateral del centro de la cara detectada respecto el cetro global de la imagen. Para así lograr la transformación de píxeles a cm.

Cabe hacer mención que el manejo de los centímetros es mas adecuado para esta aplicación ya que son medidas mas pequeñas que pueden ser enviadas en menor cantidad de paquetes. Este beneficio puede ser observado con mayor facilidad cuando la resolución de la cámara es aumentada, ya que las medidas en píxeles se incrementan considerablemente, afectando directamente el envío de datos.

Dado que la cantidad de información que debe enviarse deber ser codificada y sobre todo lo mas compacta que se pueda, la resolución adecuada para esta aplicación es 640x480. Resolución con la que después de una gran cantidad de pruebas se ha obtenido el resultado observado en 8.3, donde se muestra el comportamiento de la mejor curva de ajuste para los desplazamientos laterales.

Cabe hacer mención que debido a que los datos tanto de la interpolación anterior con esta presentan un comportamiento muy peculiar, los métodos convencionales de interpolación no logran ajustar de manera correcta los datos, por lo que se recurrió a un tipo de interpolación exacta, que a diferencia de las polinómicas o convencionales, tiene que pasar por los puntos que se conocen, por lo que el ajuste es mejor considerablemente. Para mayor información sobre esta interpolación puede consultarse [2].

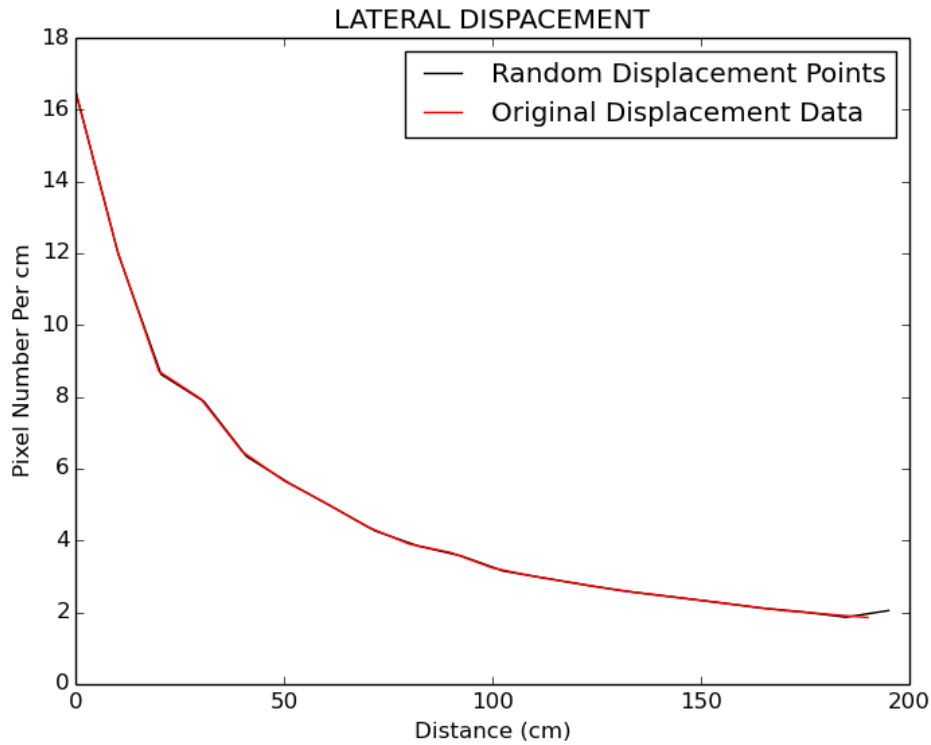


Figura 8.3: Calibración Cámara Pixel por cm para Distancia Lateral

8.1.3 Comportamiento de posiciones del cuadri-rotor

Esta subsección está enfocada a los comportamientos obtenidos de los desplazamientos del rostro con respecto al centro de la imagen, ya que nuestro objetivo es mantener posicionado respecto al centro de la imagen al vehículo. Es decir el propósito de esta subsección es encontrar el error que existe entre la detección facial y el centro global, entendiéndose como centro global, al píxel central de la imagen tal que tenga coordenadas (c_x, c_y) .

Dado que las mediciones que se obtienen inicialmente están ubicadas en una posición

diferente al centro de la imagen, es preciso realizar un desplazamiento en los píxeles. Tal que nos ubique en (c_x, c_y) . Para observar estas medidas que son obtenidas a partir del algoritmo Haar, se presenta la gráfica 8.4, donde son mostradas las mediciones son procesar.

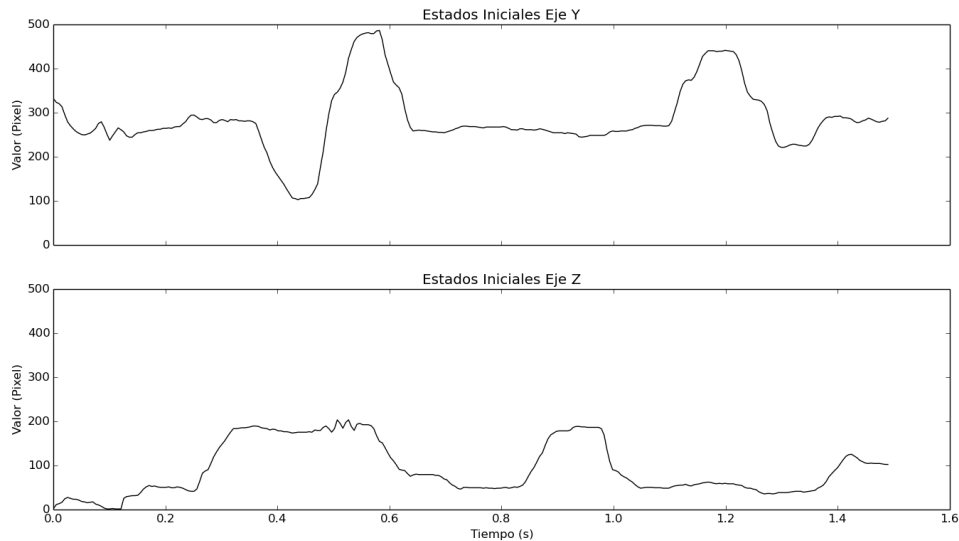


Figura 8.4: Posiciones Originales

Una vez realizado el ajuste de parámetros para la obtención del centro del rostro lo siguiente realizado fue la estimación de estos parámetros, mediante el filtro de kalman.

Prueba de una de las múltiples pruebas que se han realizado a lo largo del proyecto es detallado en 8.5. Gráfica en la cual se pueden apreciar 2 señales que representan la posición asignada por la detección del rostro (color azul) y su respectiva estimación (color negro).

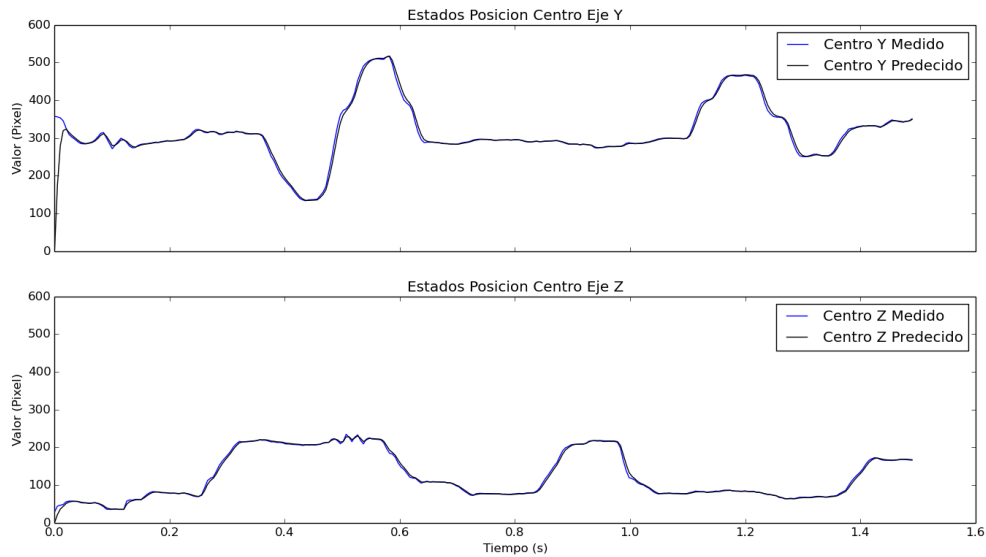


Figura 8.5: Filtro de Kalman Posiciones

Por lo tanto una vez que se ha conseguido obtener las estimaciones, tanto de posición y como veremos mas adelante también de velocidad, se tiene lo necesario para con estos datos poder calcular el parámetro de error que genera cada detección de rostro, respecto al centro global.

Así para fines de control el error generado que expresa la relación entre la posición deseada (centro de la imagen) y la posición obtenida a lo largo del tiempo, puede ser apreciada en 8.6, donde también es posible conocer el desplazamiento que el rostro realiza. Es decir podemos saber si el rostro esta ubicado de lado derecho, izquierdo, sobre o debajo de la posición deseada.

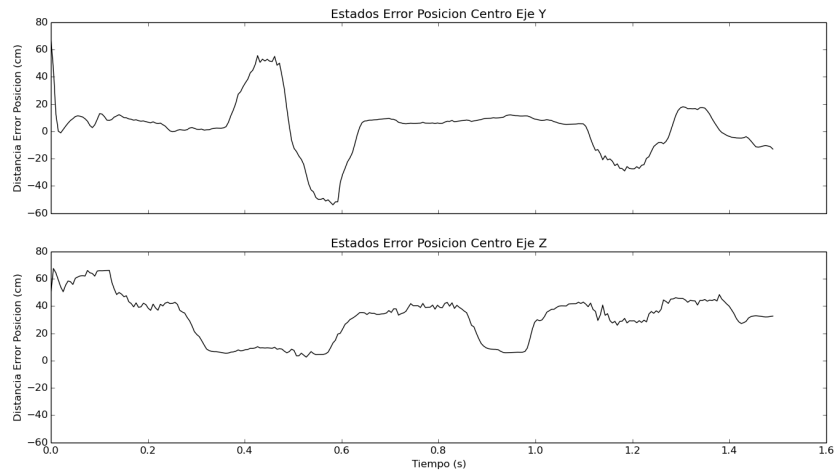


Figura 8.6: Error Posicionamiento Respecto al Centro de La Cámara

Una vez visto el comportamiento de las posiciones que la detección del rostro genera directamente se agrega 8.7, gráfica que muestra la integración del tercer eje que controla el acercamiento o alejamiento lo cual completa el control en 3 dimensiones.

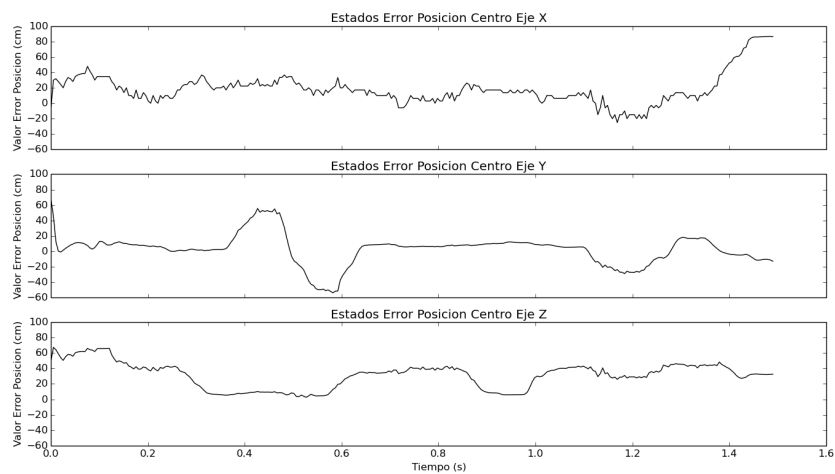


Figura 8.7: Error Posicionamiento Respecto al Rostro

8.1.4 Comportamiento de velocidades del cuadri-rotor

Así como se ha observado con anterioridad con los datos referentes a las posiciones que registra el vehículo mientras realiza las detecciones faciales, de manera equivalente es posible conocer el comportamiento de la velocidad del movimiento del rostro con respecto al observador que en este caso es el quadri-rotor.

De manera análoga que la sección anterior la velocidad que es obtenida nos proporciona información necesaria para realizar el control del cuadri-rotor. Muestra del comportamiento de estas velocidades, puede observarse en 8.8. Donde un incremento en la velocidad del vehículo podrá notar cuando el signo sea positivo y por el contrario un decremento en la velocidad se denotara cuando este valor sea negativo.

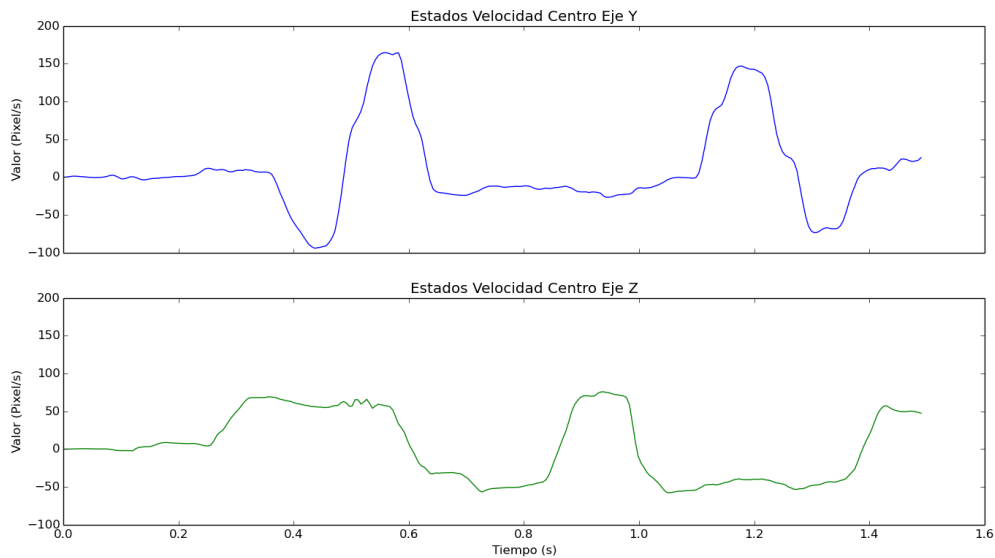


Figura 8.8: Filtro de Kalman Velocidades

Además del comportamiento de la velocidad se ha logrado obtener la diferencia existente entre la velocidad que se desea que el vehículo conserve de la imagen y la velocidad del rostro detectado obteniendo los datos que pueden observarse en 8.9. Lo cual a su vez nos brindara información del aumento y decremento de velocidad, con respecto a la posición deseada.

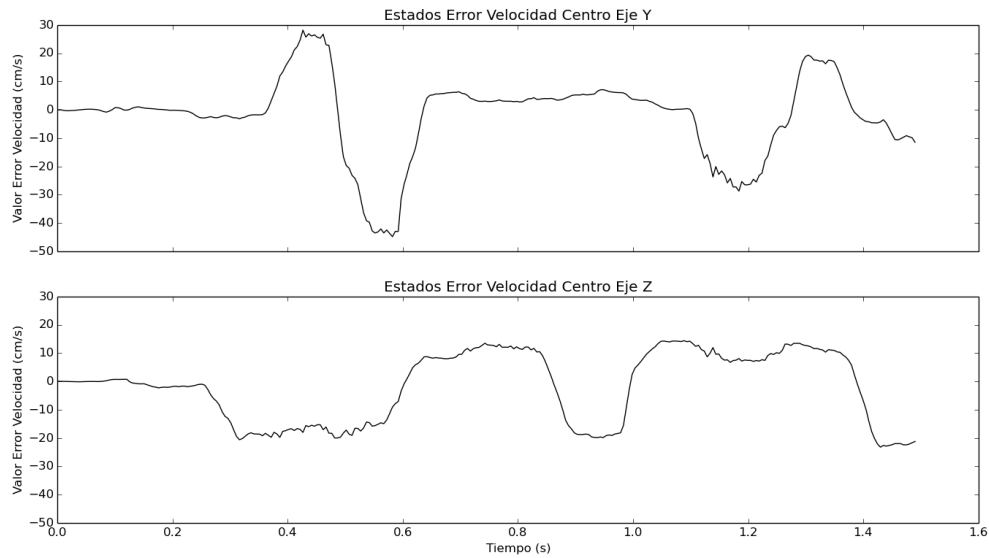


Figura 8.9: Error Velocidad Respecto al Centroide de La Cámara

De la misma forma que en la subsección anterior al a los controles en los ejes lateral y de altitud se ha agregado el eje de proximidad, lo cual completa el control tridimensional.

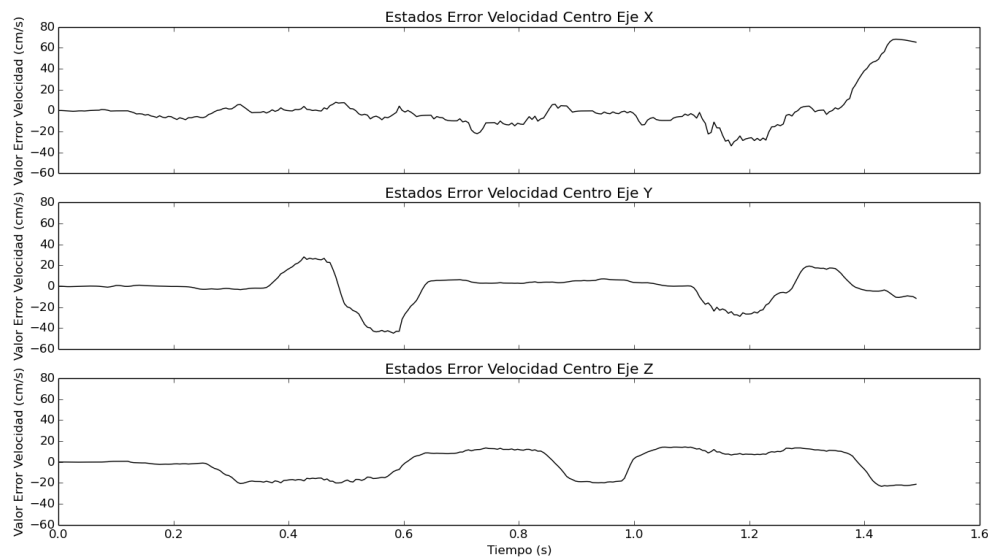


Figura 8.10: Error Velocidad Respecto al Rostro

8.2 Resultados Pixhawk Compatible (Autopiloto)

8.2.1 Datos Recibidos

Hasta ahora solo se han mostrado, los datos que son almacenados dentro de la computadora embebida, en esta presente subsección se muestran los datos que son guardados dentro del autopiloto, los cuales no solo se han generado dentro del mismo, sino también recibidos desde la computadora embebida.

Para mostrar estos comportamientos almacenados se muestran 8.11, 8.12, y 8.13, en donde los errores enviados desde la computadora embebida, pertenecientes al eje z, y, x, respectivamente.

No olvidando que lo visto en 8.11, representa el error de altitud de un rostro detec-

tado respecto al centro de la imagen obtenida.

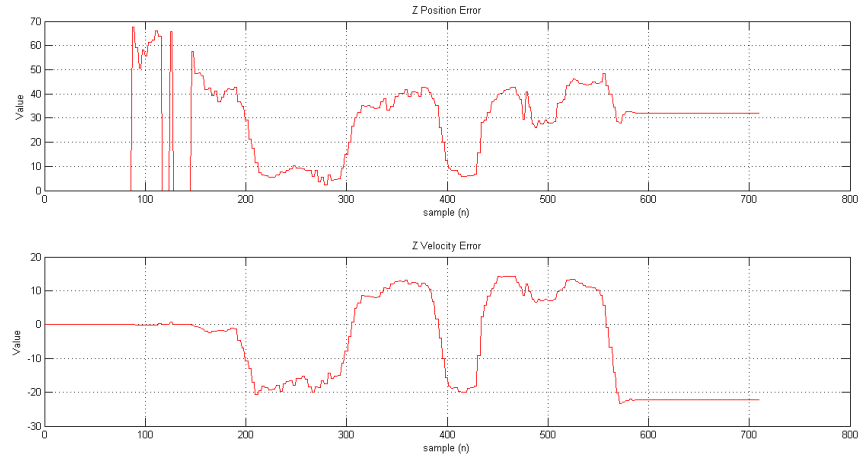


Figura 8.11: Error de Posición y Velocidad en el Z

De manera similar 8.12, representa el desplazamiento lateral de un rostro respecto al centro de la imagen captada.

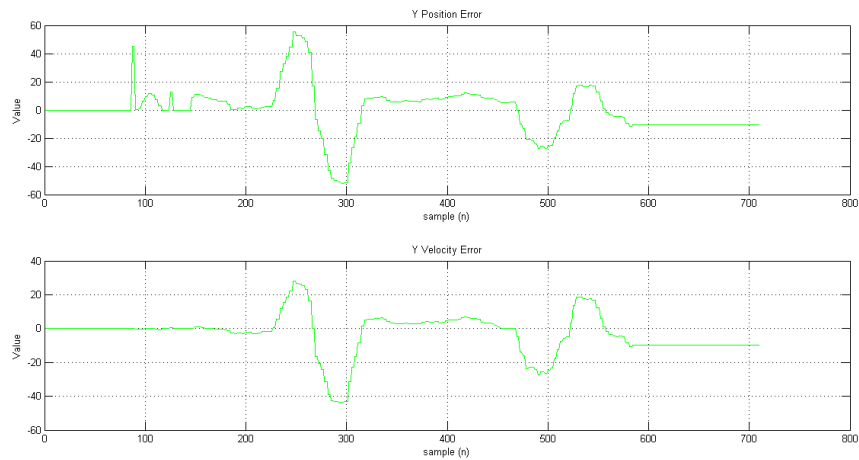


Figura 8.12: Error de Posición y Velocidad en el Y

Por su parte 8.13, representa la cercanía de un rostro al cuadri-rotor.

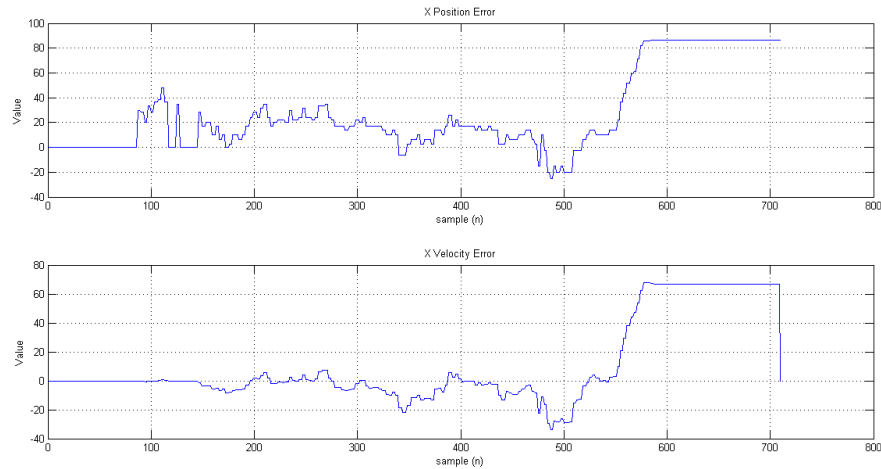


Figura 8.13: Error de Posición y Velocidad en el X

Y como era de esperarse las gráficas 8.11 8.12, 8.13, son muy parecidas a 8.6 y 8.9, pues la única diferencia radica en que el autopiloto comienza a guardar información desde el momento es que esta energizado y la computadora embebida solo guarda los datos cuando hay detección de un rostro.

Debido a que las curvas recibidas de algunas pruebas poseían cambios súbitos en algunas medidas, se decidió la implementación de un filtro tal que suavizara las curvas obtenidas.

El resultado de ese filtro para la altura respecto a la cara puede ser visto en 8.14, donde es posible ver que el filtro logra su cometido, eliminando las mediciones que cambian muy súbitamente.

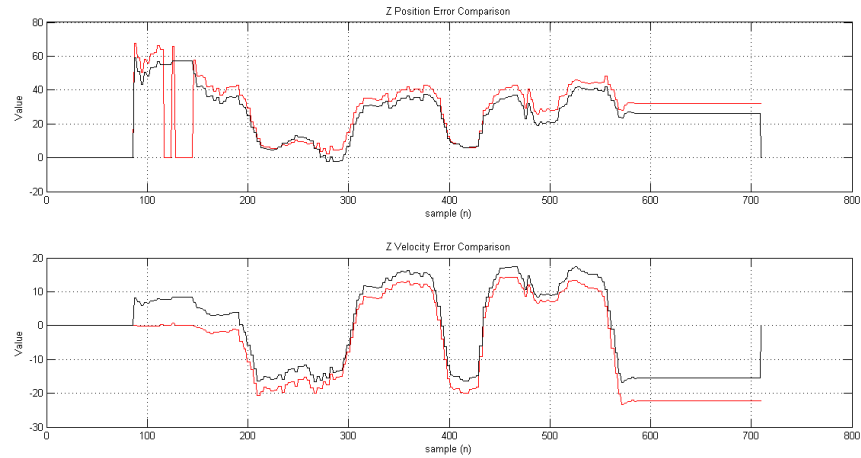


Figura 8.14: Error Filtrado de Posición y Velocidad en el Z

Del mismo modo el proceso de filtrado se ha realizado para los desplazamientos laterales, con el propósito de eliminar los cambio súbitos. Obteniendo los resultados 8.15.

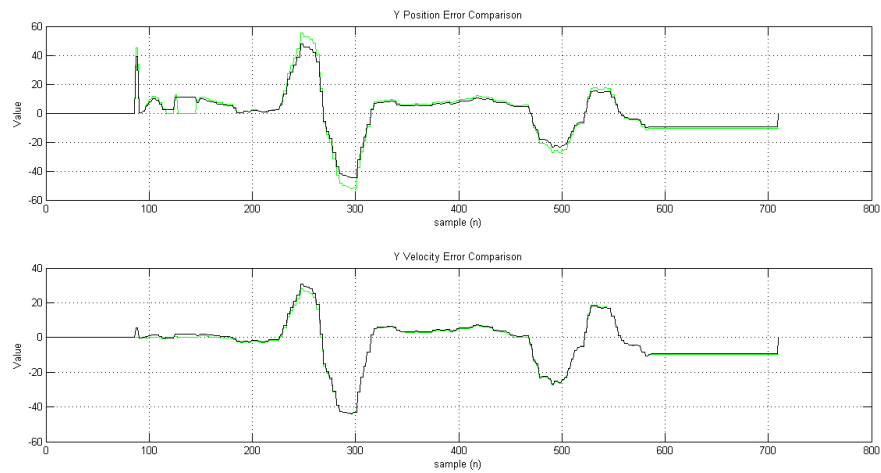


Figura 8.15: Error Filtrado de Posición y Velocidad en el Y

Cabe mencionar que de los 3 ejes que están implementados el eje que ha mostrado menos eficiencia ha sido el del acercamiento en la variable de velocidad, en cuestión de filtrado. Esto puede ser visto en 8.16.

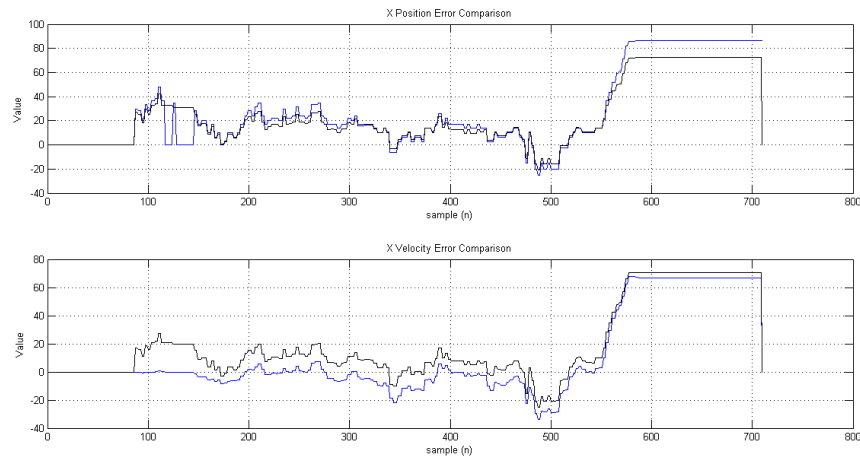


Figura 8.16: Error Filtrado de Posición y Velocidad en el X

Dadas estas posiciones y velocidades se ha podido implementar el control PD, cuyo comportamiento puede ser visto en 8.17, el cual tiene como objetivo llevar al cuadricóptero a la referencia deseada.

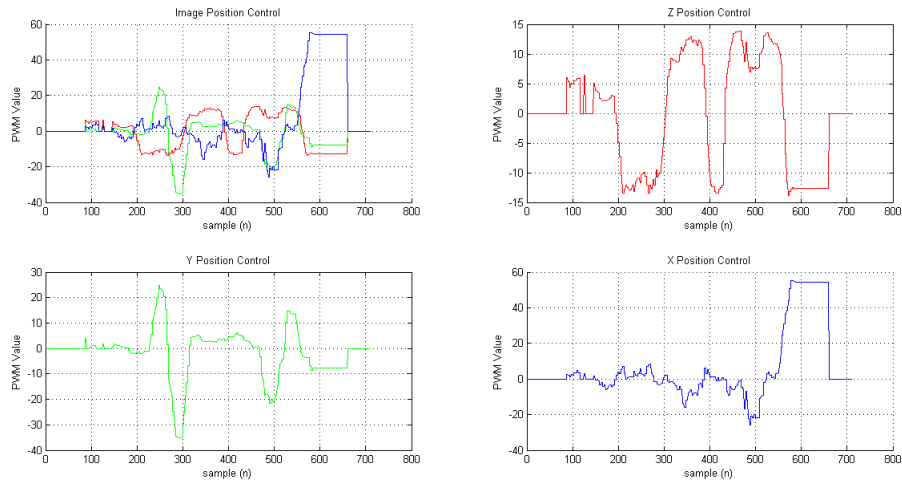


Figura 8.17: Señales de Control

Por otra parte estas señales generadas en los controles actúan directamente en el comportamiento de los motores como puede ser observado en 8.18.

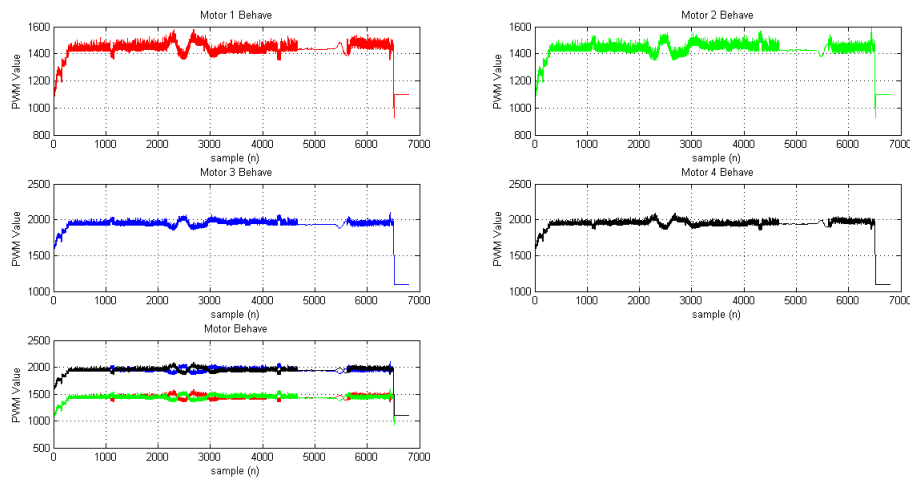


Figura 8.18: PWM en Motores

Prueba de la comunicación existente entre la computadora embebida y el autopiloto se muestra 8.19 ,en donde se puede apreciar que datos de acuerdo con la detección son recibidos en el pixhawk, de manera correcta. Cabe destacar que esta es solo una prueba fuera de vuelo, ya que el objetivo principal es solo la de mostrar que el flujo de datos se realiza correctamente.



Figura 8.19: Versión 3 Final

CAPÍTULO 9

CONCLUSIONES

El presente trabajo ha hablado de un avance significativo ya que se logró el posicionamiento de un drón respecto a un objetivo con algoritmos de visión embarcados, lo cual trae nuevos panoramas, cuando de visión artificial se habla. Aunado a esto se generó una plataforma que en comparación con [10] y [14], es abierta ya que el vehículo puede ser programado tanto en el autopiloto, como en la computadora embebida. Además el vehículo puede ser modificado, para desarrollar otras tareas.

El embarcamiento fue realizado mediante la implementación de un nuevo tipo de computadora embebida y cámara para trabajos de visión artificial, trayendo como consecuencia un mejor desempeño para este tipo de algoritmos, ya que la computadora seleccionada es inclusive 10 veces mas rápida que su competencia directa (Raspberry Pi), y la cámara seleccionada es capaz de captar mayor información de un escenario, en comparación con las convencionales.

Al embarcar los codigos en el cuadri-rotor se han lograron los objetivos planteados al principio de la tesis, con lo cual quedan eliminadas las estaciones en tierra, dejándolas como opcionales para tareas de supervisión en comparación con los trabajos

que anteceden a este, generando nuevas opciones de trabajos futuros.

Pese a que los perfiles laterales no pudieron ser probados completamente a bordo, se tiene una aportación significativa, ya que los algoritmos de visión se encuentran embarcados, además de que los las librerías son funcionales aun cuando se aumentan el numero de detecciones.

CAPÍTULO 10

TRABAJO FUTURO

Dentro de las mejoras que se pueden realizar al presente trabajo son:

1. Integración del modulo de flujo óptico como auxiliar para el posicionamiento.
2. Integración de los flancos laterales al control de posición frontal.
3. Evasión de obstáculos mientras se realiza el seguimiento del objetivo.
4. Corrección de la distorsión de la imagen producida por la cámara con lente de ojo de pez, sin generar retardos dentro del procesamiento embebido.
5. Implementación de una estación en tierra para la supervisión del mismo.
6. Construcción de una estructura con mejor presentación.

ANEXOS A

ANEXO TRABAJO ADICIONAL REALIZADO

En el presente anexo se agregan todo aquel trabajo que pese a estar fuera de los objetivos se ha realizado, pero aun no se ha implementado.

A.1 Clasificador Aumentado

Para comenzar de manera general la clasificación de rostros puede extenderse de la siguiente manera:

1. Cara frontal identificado con color verde.
2. Perfil izquierdo identificado con color azul.
3. Perfil derecho identificado con color rojo.

Con lo que el número de detecciones aumenta en 2 perfiles. Prueba del algoritmo de detección extendido puede ser observado en las imágenes A.1, donde cada perfil

diferente es marcado con un color diferente.



(a) Perfil Frontal

(b) Perfil Izquierdo



(c) Perfil Derecho

Figura A.1: Detecciones y su identificación

Posterior a la extensión lo siguiente realizado es denotar una codificación, la cual permita la distinción de cada flanco detectado.

A.1.1 Detección de 3 Flancos de un Rostro

Esta sección adicional se proporcionan los resultados pertinentes a la detección de los flancos del rostro humano. Lo cual aunque esta fuera del objetivo principal se ha implementado dentro del algoritmo de visión artificial. Comprobando así el desempeño de la plataforma ante los algoritmos de visión. Donde para efectos prácticos se codifico cada una de las detecciones de la siguiente manera:

1. Cara frontal codificado con el número 1
2. Perfil izquierdo codificado con el número 2
3. Perfil derecho codificado con el número 3
4. Ningún rostro detectado codificado con el número 4

Para muestra gráfica puede observarse A.2, en donde es posible ver que el algoritmo diseñado es capaz de realizar dichas detecciones.

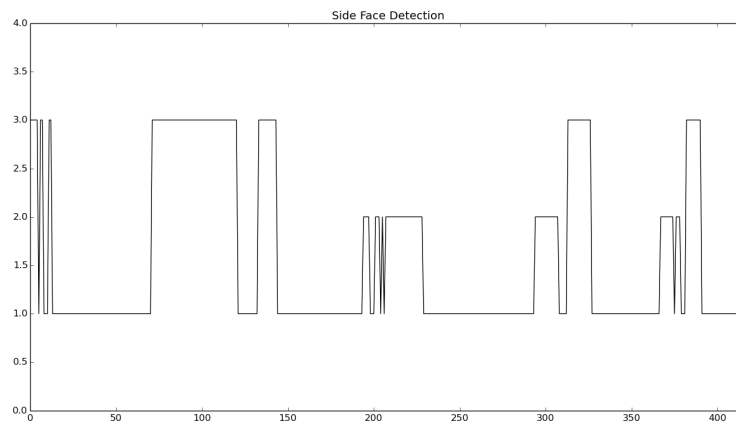


Figura A.2: Relación de Perfiles Detectados

Con el propósito de verificar que estas detecciones se hayan hecho de manera correcta, se han separado según el tipo de detección ,en base a la lista anteriormente mencionada.

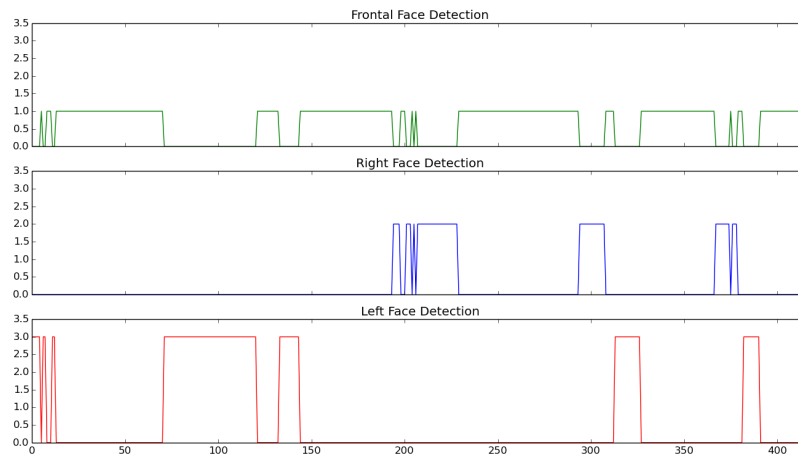


Figura A.3: Perfiles Detectados Decodificados

A.2 Datos Procesados Computadora Embebida (Odroid XU4)

A.2.1 Comportamiento de Posiciones y Velocidades Procesadas

Además de lo anterior se han recopilado datos de las detecciones tal cual se ha hecho para la cara frontal, obteniendo así lo que a continuación se muestra.

La primer gráfica mostrada en A.4, hace referencia, a las posiciones que puede ser medidas.

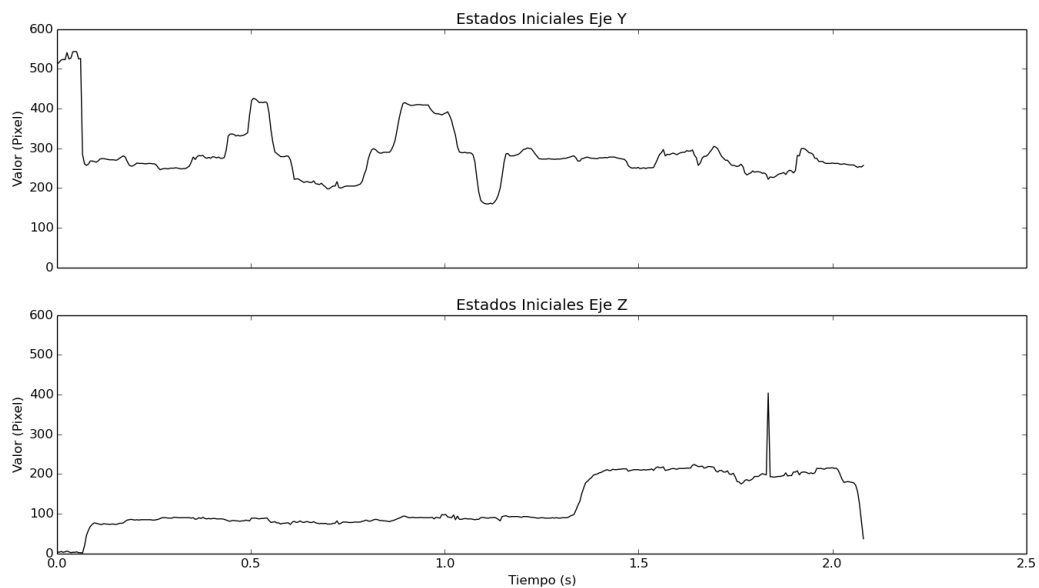


Figura A.4: Posiciones Originales

En la segunda gráfica A.5, encontramos la respuesta del filtro de calman ante las mediciones de las detecciones.

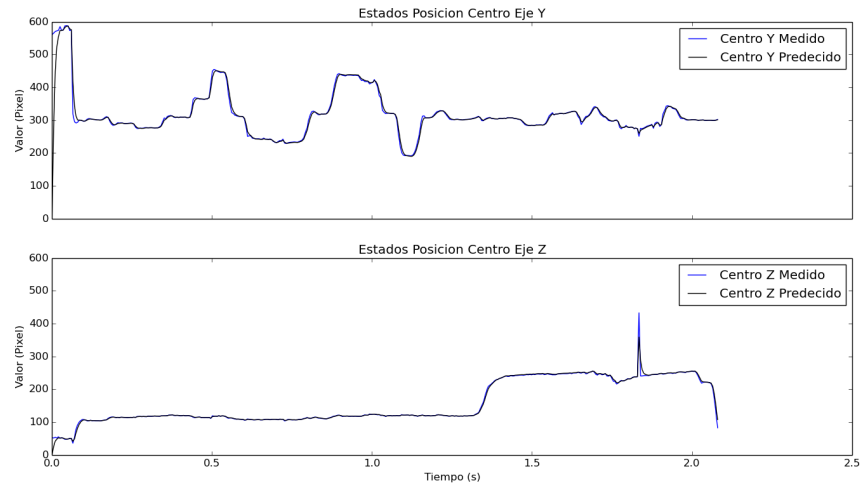


Figura A.5: Posiciones Filtro de Kalman en Ejes Y y Z

La tercera gráfica A.6, contiene las posiciones ajustadas al centro de cada detección.

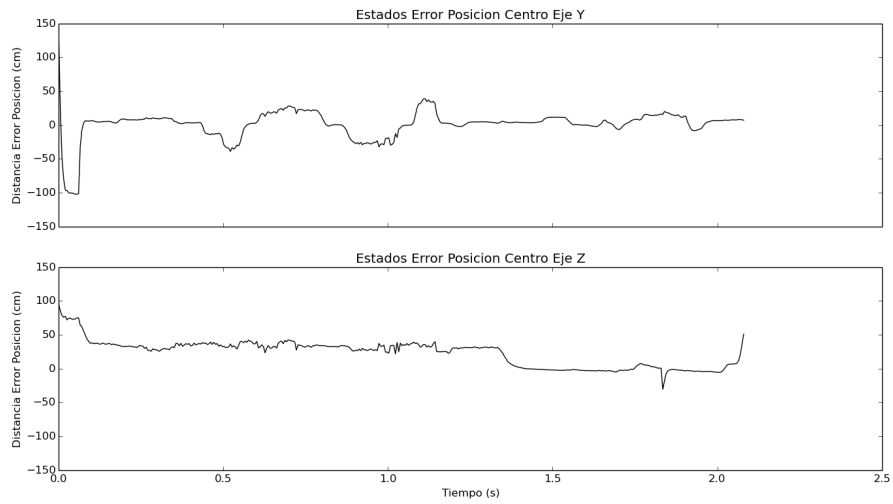


Figura A.6: Velocidad Filtro Kalman en Ejes Y y Z

Para la cuarta gráfica A.7 se muestran las velocidades obtenidas del algoritmo del Filtro de Kalman, las cuales son estimadas.

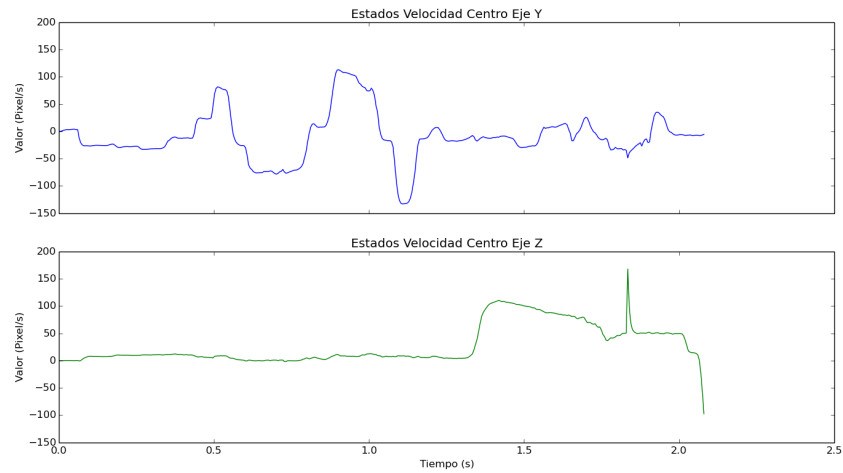


Figura A.7: Posiciones Ajustadas al Centro en Ejes Y y Z

Para finalizar se incluyen, los errores de posición en A.8, y los respectivos a velocidad en A.9, esto para en 3 dimensiones.

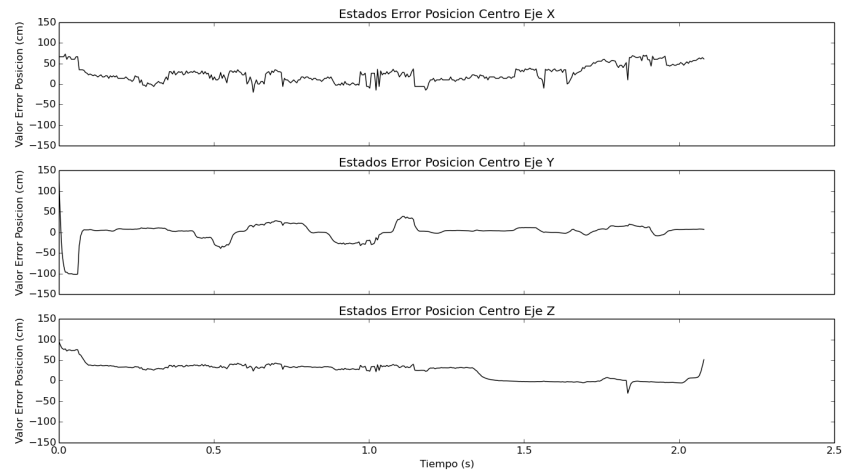


Figura A.8: Errores de Posición en 3 Ejes

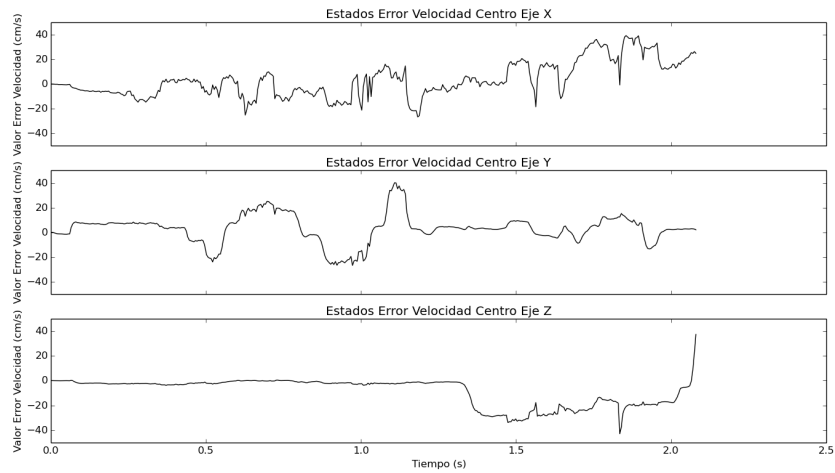


Figura A.9: Errores de Velocidad en 3 Ejes

A.3 Datos Recibidos Autopiloto (Pixhawk)

A.3.1 Comportamiento de Posiciones y Velocidades Recibidas

La presente subsección tiene como cometido al igual que en el capítulo 8 para la cara frontal, mostrar los comportamientos que han sido recibidos al autopiloto y cuya función es la de controlar el vehículo.

Para iniciar se muestra A.10, imagen en la que es posible observar el desempeño de los parámetros de control para la altitud.

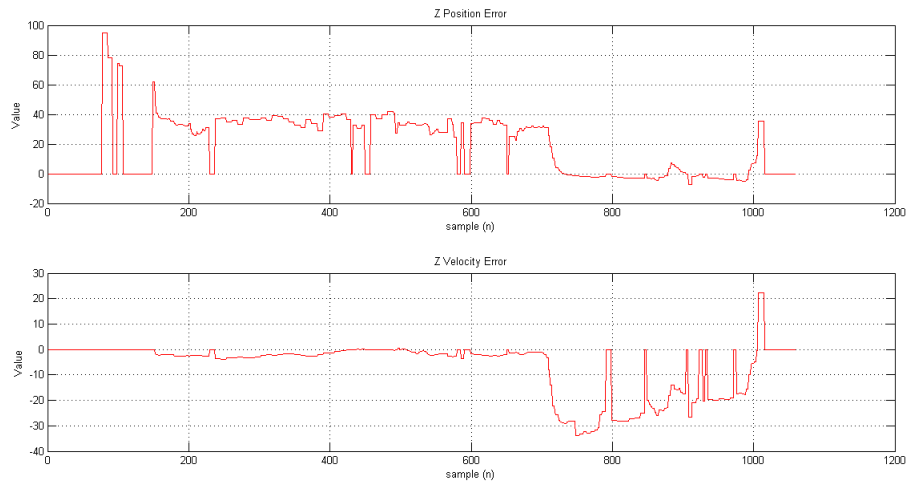


Figura A.10: Errores de Posición y Velocidad en Eje Z

De igual manera en A.11, se encuentra el desempeño que los desplazamientos laterales generan.

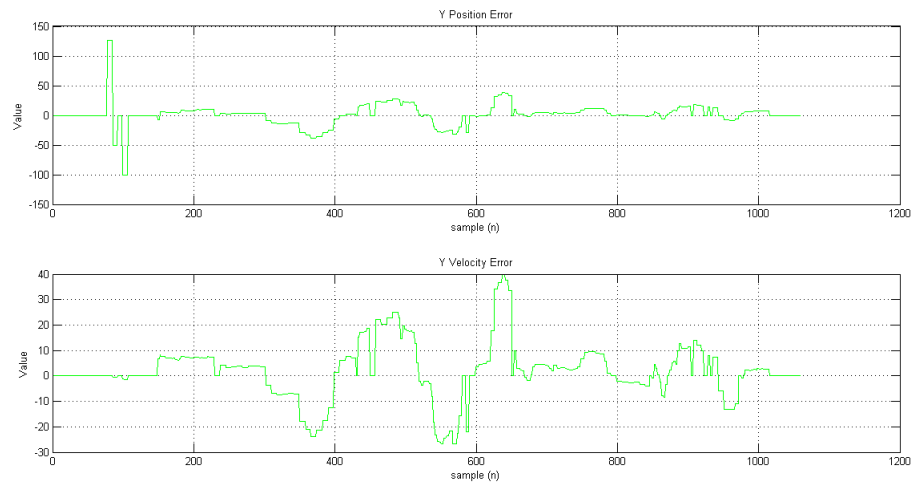


Figura A.11: Errores de Posición y Velocidad en Eje Y

Y el desempeño para el desplazamiento frontal esta dado por A.12, acción que determina el acercamiento o alejamiento.

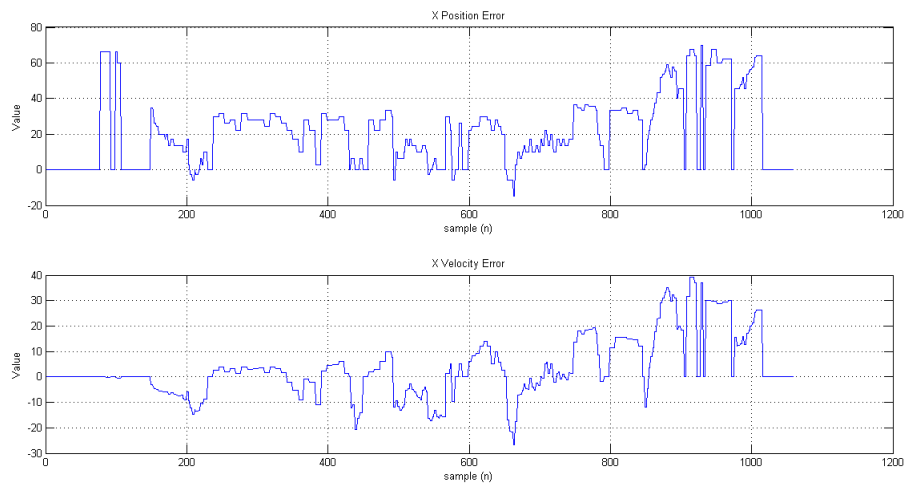


Figura A.12: Errores de Posición y Velocidad en Eje X

A continuación se muestran las señales de control que pese a no estar bien ajustadas, revelan el progreso de algunas pruebas realizadas.

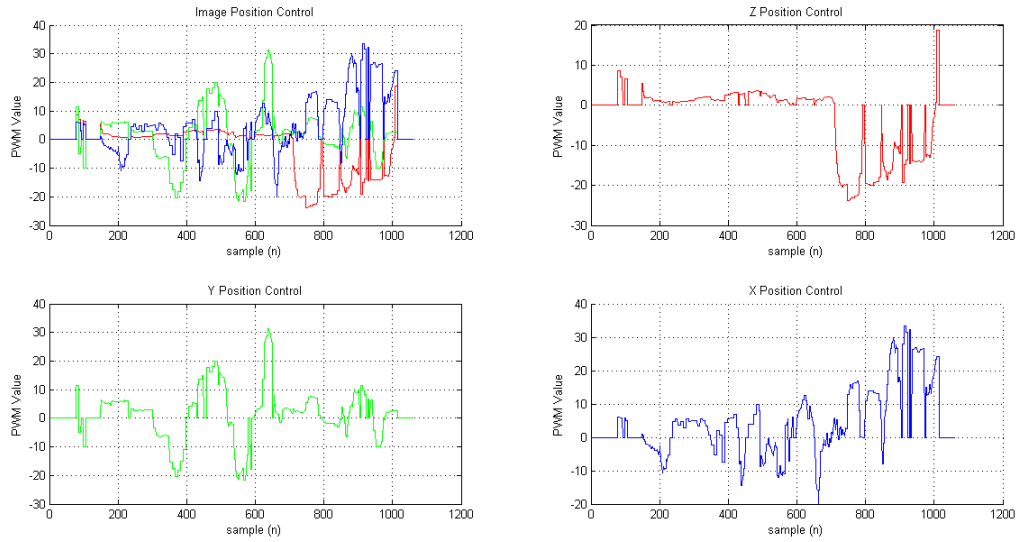


Figura A.13: Errores de Velocidad en 3 Ejes

ANEXOS B

ANEXO DE PSEUDOCÓDIGOS

En el presente anexo se agrega el código principal, en forma de un pseudo código, ya que para el correcto funcionamiento y ejecución del programa, es necesaria la ejecución de librerías especialmente creadas para este proyecto.

1. Data FaceLibrary Addition
2. Kalman Filter Parameter Definition
 - (a) Parameter Definition Kalman Filter Face
 - (b) Parameter Definition Kalman Filter Distance
3. Storage Variables Initialization
 - (a) Face Data Variable
 - (b) Current Center Face Measurement Variables
 - (c) Last Center Face Measurement Variables
 - (d) Current Distance Face Measurement Variables
 - (e) Last Distance Face Measurement Variables

- (f) Face Data Variable
 - (g) Current Center Face Predicted Variables
 - (h) Last Center Face Predicted Variables
 - (i) Current Distance Face Predicted Variables
 - (j) Last Distance Face Predicted Variables
 - (k) Deteccion Variables
 - (l) Center Face Error
 - (m) Distance Cuadcopter Error
 - (n) Pixels Per cm
4. Video Initialization
 5. Kalman Filter Face Definition
 6. Kalman Filter Distance Definition
 7. Serial Port Initialization
 8. Start Algorithm
 - (a) Video Capture
 - (b) Inicial Image Processing
 - (c) Image Center Ubication
 - (d) Parameter Initialization
 - (e) if face is detected
 - i. Frontal Face Code Identifier

- ii. Face Track
 - iii. Distance Estimation Between Face and Cuadcopter
- (f) else
- i. Non Face Code Identifier
- (g) Data Encryption
- (h) Sending Data
9. Recording Video
10. Video Display
11. Exit Condition
12. Clear Video
13. Data Storage

El pseudocódigo puede ser modificado para aumentar los perfiles detectados, simplemente agregando en la línea de detección tantos perfiles como se quieran detectar.

BIBLIOGRAFÍA

- [1] ameridroid high performance embedded computers.
- [2] Arcgis pro.
- [3] Ardupilot autopilot suite.
- [4] Chapter 11 tutorial: The kalman filter.
- [5] hobbyking.
- [6] *Quad Rotorcraft Control: Vision-Based Hovering and Navigation*.
- [7] *Unmanned Aerial Vehicles: Embedded Control*.
- [8] R. Lozano D. A. Mercado, P. Castillo. Quadrotor's trajectory tracking control using monocular vision navigation*. 2015.
- [9] Roman Pflugfelder Georg Nebel. Consensus-based matching and tracking of keypoints for object tracking. 2014.
- [10] Alessandro Giusti Gianni A. Di Caro Luca M. Gambardella Jawad Nagi. Human control of uavs using face pose estimates and hand gestures. 2014.
- [11] Shen Hao Li Shuxiao Zhu Chengfei Chang Hongxing Zhang Jinglan. Moving object detection in aerial video based on spatiotemporal saliency. 2013.

- [12] Eduardo F. Morales José Alberto Méndez Polanco, Angélica Muñoz Meléndez. People detection by a mobile robot using stereo vision in dynamic indoor environments. 2009.
- [13] Andrew J. Davison Walterio W. Mayol and David W. Murray. Real-time localisation and mapping with wearable active vision. 2003.
- [14] Diego A. Mercado. Autonomous navigation and teleoperation of unmanned aerial vehicles using monocular vision, 2015.
- [15] Michael Jones Paul Viola. Rapid object detection using a boosted cascade of simple features. 2001.
- [16] Nuri Yilmazer Shreyamsh Kamate. Application of object detection and tracking techniques for unmanned aerial vehicles. 2015.
- [17] Markus Achtelik Michael Achtelik Stephan Weiss Roland Siegwart. Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments. 2011.
- [18] Stephan Weiss Markus W. Achtelik Margarita Chli Roland Siegwart. Versatile distributed pose estimation and sensor self-calibration for an autonomous mav. 2012.
- [19] Masaharu Kobashi Jenny Yuen Linda G. Shapiro Yi Li, Indriyati Atmosukarto. Object and event recognition for aerial surveillance.
- [20] Jakob Engel Jürgen Sturm Daniel Cremers. Camera-based navigation of a low-cost quadcopter. 2012.