

**CONSTRUCCIÓN DE UN VEHÍCULO AÉREO NO TRIPULADO
PARA EL DESARROLLO DE UN ALGORITMO DE CONTROL
ADAPTABLE.**

RANGEL FUENTES EDWIN YESID

Autor

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERIAS Y ARQUITECTURA
PROGRAMA DE INGENIERIA MECATRÓNICA
PAMPLONA**

2020

**CONSTRUCCIÓN DE UN VEHÍCULO AÉREO NO TRIPULADO
PARA EL DESARROLLO DE UN ALGORITMO DE CONTROL
ADAPTABLE.**

**TRABAJO DE GRADO PARA OPTAR POR EL TITULO DE
INGENIERIO MECATRONICO**

RANGEL FUENTES EDWIN YESID

Autor

JAVIER ADOLFO CORREDOR CAMARGO

Director

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERIAS Y ARQUITECTURA
PROGRAMA DE INGENIERÍA MECATRÓNICA
PAMPLONA**

2020

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Pamplona 09 de Junio de 2020

AGRADECIMIENTOS

Dedico este logro a mis padres, a mis amigos y a los profesores que me ayudaron a realizar este trabajo.

TABLA DE CONTENIDO

Contenido

| | |
|--|----|
| 1. INTRODUCCIÓN | 15 |
| 1.1. PLANTEAMIENTO DEL PROBLEMA | 15 |
| 1.1.1..... | 16 |
| ANTECEDENTES | 16 |
| 1.1.2. JUSTIFICACIÓN..... | 18 |
| 1.2. DELIMITACIÓN..... | 18 |
| 1.2.1. OBJETIVO GENERAL..... | 18 |
| 1.2.2. OBJETIVOS ESPECÍFICOS..... | 18 |
| 2. MARCO TEÓRICO Y ESTADO DEL ARTE..... | 19 |
| 2.1. MARCO TEÓRICO..... | 19 |
| 2.1.1. TIPOS DE CONFIGURACIÓN | 20 |
| 2.1.2. PARTES DE UN DRONE | 22 |
| 2.1.3. DINAMICA..... | 23 |
| 2.2. ESTADO DEL ARTE | 25 |
| 3. INGENIERÍA DE PROYECTO | 29 |
| 3.1 DISEÑO MECÁNICO | 30 |
| 3.1.1. Selección de componentes | 31 |
| 3.1.2 DISEÑO DE BASE DE PRUEBAS 1 GDL | 43 |
| 3.1.3. DISEÑO MECÁNICO DEL DRONE..... | 49 |
| 3.1.4 DISEÑO MECÁNICO DE LA PLATAFORMA DE PRUEBAS 2 GDL..... | 57 |
| 3.1.5. ANÁLISIS DE ELEMENTOS FINITOS | 60 |
| 3.2 DISEÑO ELECTRÓNICO..... | 70 |
| 3.2.1. CONEXIONES Y FACTORES A TENER EN CUENTA | 70 |
| 3.3 DISEÑO DE CONTROL | 73 |
| 3.3.1. DISEÑO DE CONTROL CLASICO PD-PID (POSTURA – POSICIÓN) | 73 |
| 3.3.2. DISEÑO DE CONTROL ADAPTATIVO | 89 |
| 4. IMPLEMENTACIÓN | 95 |
| 4.1. Ensamble en la plataforma de 2 GDL..... | 95 |
| 4.1.2. Preparación del sistema | 97 |
| 4.2. Calibración de la postura del dispositivo | 97 |

| | |
|--|-----|
| 5. RESULTADOS | 99 |
| 5.1. Resultados experimentales | 99 |
| 5.1.1. RESULTADO DE SIMULACIONES REALIZADAS CON CONTROL CLÁSICO PD | 99 |
| 5.1.2. RESULTADO DE SIMULACIONES REALIZADAS CON CONTROL CLASICO Y CAMBIO DE MASA | 100 |
| 5.1.3. RESULTADO DE CONTROL ADAPTATIVO | 102 |
| 6. CONCLUSION Y DISCUSION | 105 |
| BIBLIOGRAFIA | 106 |
| ANEXOS | 110 |
| 1. IMPRESIÓN 3D..... | 110 |
| 2. DATOS DE LOS MOTORES..... | 111 |
| 3. AJUSTE PARAMETROS SIMULINK | 112 |
| 4. CALIBRACION INICIAL BEAGLEBONE BLUE | 123 |

LISTA DE TABLAS

| | |
|--|----|
| Tabla 1 Características de los Motores <i>brushless</i> encontrados en el mercado. Fuente: Relacionada en cada motor en anexos 2. | 37 |
| Tabla 2 Características de las tarjetas de control encontrados en el mercado. Fuente: Paginas de venta en internet..... | 39 |
| Tabla 3 Características de los sensores de proximidad encontrados en el mercado. Fuente: Paginas de venta en internet..... | 41 |
| Tabla 4 Características de las hélices encontradas en el mercado. Fuente: rcgroups.com..... | 42 |
| Tabla 5 Características de los Motores <i>brushless</i> encontrados en el mercado. Fuente: Paginas de venta en internet..... | 42 |
| Tabla 6 Características de las baterías LiPo encontrados en el mercado. Fuente: Paginas de venta en internet..... | 43 |
| Tabla 7 Caracterización del motor 1. Fuente: Autor..... | 48 |
| Tabla 8 Caracterización del motor 2. Fuente: Autor..... | 48 |
| Tabla 9 Caracterización del motor 3. Fuente: Autor..... | 48 |
| Tabla 10 Caracterización del motor 4. Fuente: Autor..... | 48 |
| Tabla 11 Peso de los elementos del sistema. Fuente: Autor..... | 61 |

TABLA DE FIGURAS

| | |
|---|----|
| Ilustración 1 Configuración “x” y “+”, Fuente: Ardupilot | 20 |
| Ilustración 2 Fuerza y Torque sobre un sistema de referencia. Fuente: Coursera | 24 |
| Ilustración 3 Fuerzas y torques aplicadas a un drone. Fuente: Coursera. | 25 |
| Ilustración 4 Velocidades angulares y fuerzas aplicadas a un drone configuración X. Fuente: Coursera | 26 |
| Ilustración 5 Modelo quanser 3 DOF. Fuente: Quanser..... | 28 |
| Ilustración 6 Estructura mecánica F330. Fuente: Autor..... | 29 |
| Ilustración 7 Motor <i>Brushless</i> A2212 1000KV. Fuente: Autor..... | 31 |
| Ilustración 8 Esc 30A. Fuente: Autor. | 32 |
| Ilustración 9 Beaglebone Blue. Fuente: Autor | 33 |
| Ilustración 10 Batería LiPo. Fuente: Autor | 33 |
| Ilustración 11 Hélice 1045. Fuente: Autor | 34 |
| Ilustración 12 Producto final de plataforma de pruebas de 1gdl. Fuente: Autor..... | 43 |
| Ilustración 13 Pieza uno de plataforma 1gdl. Fuente: Autor. | 44 |
| Ilustración 14 Pieza dos de plataforma 1gdl. Fuente: Autor..... | 44 |
| Ilustración 15 Pieza tres de plataforma 1gdl. Fuente: Autor. | 44 |
| Ilustración 16 Pieza cuatro de plataforma 1gdl. Fuente: Autor..... | 44 |
| Ilustración 17 Pieza cinco de plataforma 1gdl. Fuente: Autor. | 45 |
| Ilustración 18 Pieza seis de plataforma 1 gdl. Fuente: Autor..... | 45 |
| Ilustración 19 Proceso de ensamblaje de base de pruebas de 1gdl. Fuente: Autor..... | 46 |
| Ilustración 21 Señal de control enviada al ESC. Fuente: Autor | 47 |
| Ilustración 22 Diagrama de bloques para las pruebas de 1GDL..... | 47 |
| Ilustración 23 Diseño del brazo del drone. Fuente: Autor. | 49 |
| Ilustración 24 Vista frontal, izquierda y superior del brazo del drone. Fuente: Autor | 50 |
| Ilustración 25 Soporte Superior del drone. Fuente: Autor. | 51 |
| Ilustración 26 Vista frontal, izquierda y superior del soporte superior del drone. Fuente: Autor ... | 51 |
| Ilustración 27 Soporte inferior del drone. Fuente: Autor | 52 |
| Ilustración 28 Vista frontal, izquierda y superior del soporte superior del drone. Fuente: Autor ... | 52 |
| Ilustración 29 Ensamble de las piezas del drone. Fuente: Autor. | 53 |
| Ilustración 30 Recreación de las fuerzas ejercidas sobre el drone. Fuente: Autor..... | 54 |
| Ilustración 31 Vista general del análisis de tensión en el drone. Fuente: Autor..... | 54 |
| Ilustración 32 Vista de corte del drone en puntos críticos del análisis de tensión. Fuente: Autor. . | 55 |
| Ilustración 33 Vista de corte del drone en puntos críticos del análisis de estrés. Fuente: Autor..... | 55 |
| Ilustración 34 Factor de seguridad del drone. Fuente: Autor. | 56 |
| Ilustración 35 Vista de desplazamiento máximo del drone. Fuente: Autor..... | 56 |
| Ilustración 36 Quanser Hover 3 DOF. Fuente: Quanser..... | 57 |
| Ilustración 37 Sistema de rotación del Quanser hover 3 DOF. Fuente: Quanser. | 57 |
| Ilustración 38 Sistema de referencia Quanser 3 dof. Fuente: Quanser. | 58 |
| Ilustración 39 Soporte de rectangular de la base de pruebas de 2gdl. Fuente: Autor. | 58 |
| Ilustración 40 Piezas de unión con el eje del drone con la plataforma de 2gdl. Fuente: Autor. | 59 |

| | |
|--|----|
| Ilustración 41 Sujeción del rodamiento circular. Fuente: Autor..... | 59 |
| Ilustración 42 Soporte de sujeción. Fuente: Autor. | 59 |
| Ilustración 43 Eje de rotación del segundo grado de libertad. Fuente: Autor..... | 60 |
| Ilustración 44 Sujeción del segundo eje de rotación. Fuente: Autor. | 60 |
| Ilustración 45 Análisis de elementos finitos en la base de 2 gdl. Fuente: Autor. | 62 |
| Ilustración 46 Vista de corte del análisis de estrés para el primer eje de rotación. Fuente: Autor.. | 62 |
| Ilustración 47 Vista de corte de análisis de estrés para el segundo eje de rotación. Fuente: Autor. | 63 |
| Ilustración 48 Vista de corte análisis de tensión para el primer eje de rotación. Fuente: Autor. | 63 |
| Ilustración 49 Vista de corte de análisis de tensión del segundo eje de rotación. Fuente: Autor.... | 64 |
| Ilustración 50 Prueba general de factor de seguridad. Fuente: Autor..... | 64 |
| Ilustración 51 Análisis de desplazamiento del sistema. Fuente: Autor | 65 |
| Ilustración 52 Análisis dinámico de fuerza contraria necesaria para el levantamiento del dispositivo. Fuente: Autor..... | 66 |
| Ilustración 53 Vista de corte de análisis de estrés para el primer grado de libertad. Fuente: Autor. | 67 |
| Ilustración 54 Vista de corte análisis de estrés para el segundo grado de libertad. Fuente: Autor. | 67 |
| Ilustración 55 Vista de corte de análisis de tensión para el primer eje de rotación. Fuente: Autor. | 68 |
| Ilustración 56 Vista de corte de análisis de tensión para el segundo eje de rotación. Fuente: Autor. | 68 |
| Ilustración 57 Prueba general de factor de seguridad. Fuente: Autor..... | 69 |
| Ilustración 58 Análisis de desplazamiento del sistema. Fuente: Autor. | 69 |
| Ilustración 59 Motor <i>Brushless</i> . Fuente: Autor. | 70 |
| Ilustración 60 Esc 30A. Fuente: Autor. | 70 |
| Ilustración 61 Beaglebone Blue. Fuente: Autor. | 71 |
| Ilustración 62 Batería LiPo Turnigy 2200 mah. Fuente: Autor | 71 |
| Ilustración 63 Esquema eléctrico del dron. Fuente: Autor. | 72 |
| Ilustración 64 Esquema de control del dron. Fuente: Autor..... | 72 |
| Ilustración 65 Diagrama de control dinámico del dron. Fuente: Autor. | 73 |
| Ilustración 66 Diagrama de control dinámico del dron. Fuente: Autor. | 74 |
| Ilustración 67 Aplicación SI_quadrotor de la <i>toolbox</i> de Peter Corke. Fuente: <i>Toolbox</i> Peter Corke. | 76 |
| Ilustración 68 Control dinámico implementando las ecuaciones de control del curso "Aerial Robots". Fuente: Autor. | 77 |
| Ilustración 69 Controlador de postura. Fuente: Autor..... | 77 |
| Ilustración 70 Controlador de posición. Fuente: Autor..... | 78 |
| Ilustración 71 Controlador de postura. Fuente: Autor..... | 79 |
| Ilustración 72 Ángulos de control Yaw, Pitch y Roll. | 80 |
| Ilustración 73 Aceleraciones angulares de control. Fuente: Autor..... | 80 |
| Ilustración 74 Controlador PD para el los tres ejes de rotación. Fuente: Autor. | 81 |
| Ilustración 75 Esquema de la planta del sistema. Fuente: Autor..... | 82 |
| Ilustración 76 Respuesta del sistema en lazo cerrado para el giro en el eje "z", sin controlador. Fuente: Autor. | 83 |
| Ilustración 77 Respuesta de sistema, controlador Heurístico. Fuente: Autor. | 83 |
| Ilustración 78 Respuesta del sistema en lazo cerrado para el giro del eje "y". Fuente: Autor. | 84 |

| | |
|--|-----|
| Ilustración 79 Posición del dispositivo para la respuesta del sistema en el giro en el eje "y". Fuente: Autor. | 84 |
| Ilustración 80 Respuesta de giro del eje "y" implementando controlador Heurístico PD Fuente: Autor. | 85 |
| Ilustración 81 Respuesta de la posición con controlador en lazo cerrado para el giro del eje "y". Fuente: Autor. | 85 |
| Ilustración 82 Respuesta del sistema de giro en el eje "z" con señal realimentada sin controlador. Fuente: Autor. | 86 |
| Ilustración 83 Respuesta del sistema de posición en el eje "z" con señal realimentada y controlador heurístico. Fuente: Autor. | 87 |
| Ilustración 84 Respuesta del sistema de posición en el eje "x" con señal realimentada sin controlador. Fuente: Autor. | 88 |
| Ilustración 85 Respuesta del sistema de posición en el eje "x" con señal realimentada con controlador heurístico. Fuente: Autor. | 88 |
| Ilustración 86 Esquema de control adaptativo. Fuente: Autor. | 89 |
| Ilustración 87 Respuesta del sistema con incremento de masa en el segundo 5..... | 90 |
| Ilustración 88 Respuesta del sistema masa=1kg, controlador P=1000 D=250 | 91 |
| Ilustración 89 Respuesta del sistema masa 1kg, controlador P=1000 D=1400 | 91 |
| Ilustración 90 Error entre modelo de referencia masa 1 kg y controlador P=2 D=3 y sistema real con masa 1.2kg, controlador P=1000 D=1400 | 92 |
| Ilustración 91 Error entre modelo de referencia masa 1 kg y controlador P=2 D=3 y sistema real con masa 1.2kg, controlador P=2 D=3 | 92 |
| Ilustración 92 Error entre modelo de referencia masa 1 kg y controlador P=2 D=3 y sistema real con masa 1.2kg, controlador P=1000 D=250 | 93 |
| Ilustración 93 Plataforma de 2 gdl. Fuente: Autor. | 95 |
| Ilustración 94 Vista interna drone. Fuente: Autor. | 95 |
| Ilustración 95 Drone. Fuente: Autor. | 96 |
| Ilustración 96 Soporte de Beaglebone Blue. Fuente: Autor. | 96 |
| Ilustración 97 Drone en plataforma de 2gdl. Fuente: Autor. | 96 |
| Ilustración 98 Diagrama de bloques de calibración de giroscopio. | 98 |
| Ilustración 99 Respuesta del giroscopio de Beaglebone Blue ajustado. Fuente: Autor | 98 |
| Ilustración 101 Señal deseada de posición para el eje "x", "y" y "z". Fuente: Autor. | 99 |
| Ilustración 102 Respuesta del sistema con controlador PD heurístico. Fuente: Autor. | 100 |
| Ilustración 103 Respuesta del sistema con controlador PID heurístico y cambio en el peso del sistema. Fuente: Autor. | 101 |
| Ilustración 104 Respuesta del sistema con controlador adaptativo y masa constante = 1.2kg. Fuente Autor | 102 |
| Ilustración 105 Señal de respuesta con masa al segundo 0 = 1kg, masa al segundo 5=1.2kg y masa al segundo 18=1kg. Fuente: Autor. | 103 |
| Ilustración 106 Controlador PD para ajuste en la masa al segundo 5 y 18. Fuente: Autor | 103 |
| Ilustración 107 Controlador PD para ajuste en la masa al segundo 5 y 27. Fuente: Autor. | 104 |
| Ilustración 108 Respuesta del sistema para el controlador adaptativo con ajuste en la masa en el segundo 5 y 27. Fuente: Autor. | 104 |
| Ilustración 109 Ajuste de la propiedad de materiales en el programa Fusión 360. Fuente: Autor. | 110 |

| | |
|--|-----|
| Ilustración 110 Impresión 3D, del brazo del cuadricoptero. Fuente: Autor. | 110 |
| Ilustración 111 Diagrama de bloques de la ley de adaptación. Fuente: Autor. | 123 |
| Ilustración 112 Paso 1 Configuración de Beagle en Matlab. | 123 |
| Ilustración 113 Paso 2 configuración de Beagle en Matlab. | 123 |

GLOSARIO

VANT: Vehículo Aéreo no Tripulado

A_{HC}^S : Momento angular

$F = \frac{AdL}{dt}$: Tasa de cambio del momento lineal

U_1 : Empuje total del sistema

m : Masa

g : Gravedad

F = Fuerza de cada motor

K_f : Constante de fuerza del motor

K_m : Constante de momento del motor

ω_i : Velocidad angular del motor

m : Masa del dispositivo

g : Gravedad

M_i : Momento de cada motor

M : Momento total del sistema

R : Matriz de rotación.

\ddot{r} : Aceleración en el eje z.

I : Matriz de inercia

$A_{\omega B}$ Velocidad angular de un cuerpo rígido B, con sistema inercial A.

$\frac{AdH_C}{dt}$ Derivada del momento angular del centro de masa C, con sistema inercial A.

r_C : Ubicación del centro de masa "C".

p_i : Partícula i.

$\frac{AdAv^C}{dt}$: Derivada de la velocidad en el centro de masa C, de un sistema inercial A.

H_C : Momento angular de un cuerpo rígido C.

$\frac{BdH_C}{dt}$ Derivada del momento angular del centro de masa C, en un sistema de referencia B.

θ : Angulo roll

\emptyset : Angulo pitch

φ : Angulo yaw

U_2 : Vector de momentos

p : Velocidad angular en "x"

q : Velocidad angular en "y"

r : Velocidad angular en "z"

$k_{d,i}$: Constante de control derivativa de posición

$k_{p,i}$: Constante de control proporcional de posición

$\dot{r}_{i,c}$: Velocidad de posición controlada

$\dot{r}_{i,des}$: Velocidad de posición deseada

$\ddot{r}_{i,c}$: Aceleración de posición controlada

$\ddot{r}_{i,des}$: Aceleración de posición deseada

$r_{i,des}$: Posición deseada

$r_{i,c}$: Posición controlada

$K_{p,(ángulo)}$: Constante proporcional de control de postura (ángulo $\theta, \emptyset, \varphi$)

$K_{d,(ángulo)}$: Constante derivativa de control de postura (ángulo $\theta, \emptyset, \varphi$)

RESUMEN

Este trabajo presenta la implementación de un vehículo aéreo no tripulado (VANT) conocido como drone, cuadrirotor o quadrotor (del inglés *quadcopter*), en este documento se denominará indistintamente como drone.

El diseño y construcción del drone pretende fomentar la investigación de estrategias de control en bajo nivel para drones en la Universidad de Pamplona. Se implementa el sistema analizando la respuesta del controlador clásico, para esto se ha desarrollado el sistema de control con el sistema dinámico en SIMULINK de Matlab, además se presenta el diseño mecánico de las piezas que componen el drone, también se muestra la selección de los componentes eléctricos, se simula el modelo dinámico del drone y por último se presentan resultados en simulación del sistema de control adaptativo.

El documento presenta el diseño y análisis de elementos finitos de los componentes del sistema, después de que este análisis demostró que las piezas diseñadas cumplieron con el objetivo propuesto se procedió a la fabricación de las piezas en una impresora 3D y posteriormente se realizó el respectivo ensamble. Este análisis permitió desarrollar diferentes modelos mecánicos para obtener un modelo mucho menos pesado y lo suficientemente compacto para que no ocurra una catástrofe durante la operación del prototipo, siguiendo los requisitos fundamentales de la aplicación dado que por la rigidez de los componentes y que el objetivo principal del prototipo es el diseño de algoritmos de control, es necesario disponer de un diseño robusto. Durante este desarrollo se tuvo en cuenta el análisis de elementos finitos para cada pieza, modificando el diseño de ser necesario para luego realizar su respectivo proceso de fabricación.

En cuanto al desarrollo de los controladores, se realizó el curso de “Robotics: Aerial Robotics” de Coursera [1], impartido por la Universidad de Pensilvania, además de consultar en la documentación [2], [3]. Para la simulación del drone se aplicó el modelo dinámico del dispositivo de la clase de “Aerial Robotics” y se implementó el modelo teniendo como base la *toolbox* de Robótica de Peter Corke [4]. En seguida se realizó la simulación ajustando los diferentes parámetros necesarios para un buen desempeño del controlador adaptativo.

Se propone implementar el controlador en la tarjeta de desarrollo BeagleBone Blue, la cual cuenta con una *toolbox* desarrollada por Mathworks [5] para el uso de esta en SIMULINK. Además, se creó la librería *Brushless* Micro para ejercer control a un servo a través de una señal enviada en microsegundos y de esta forma controlar los motores *brushless* indispensables en el proyecto.

Por último, se desarrolla el modelo de control adaptativo, el cual está basado en un modelo de red neuronal que permite hacer una variación en el controlador clásico de posición, de esta forma cumplir con el cometido del desarrollo del proyecto, que es desarrollar un modelo de control adaptativo aplicado en un drone, con variaciones en los parámetros internos del sistema.

1. INTRODUCCIÓN

El desarrollo de robots aéreos ha sido una gran ayuda en la industria aeroespacial, militar, cinematográfica, agrícola, entre otras, pese al alto coste de inversión que esto requiere, y la poca confiabilidad con la que cuentan los proyectos de robótica para muchas empresas, donde algunas les ha generado excelentes resultados y aun se espera ver una gran implementación en los próximos años de este tipo de proyectos en sectores industriales más comunes y de pequeña escala, donde mejore la participación de medianas empresas en la industria 4.0.

Pero no son las únicas industrias en las que se está empezando a usar este tipo de dispositivos, incluso algunas se están reinventando debido a las problemáticas del mundo actual, grandes empresas como SpaceX o Uber entre otras, han querido implementar este tipo de dispositivos para el transporte de personas a corta distancia. Debido a la gran congestión vehicular que se vive en grandes ciudades [6], pero su gran problema es la responsabilidad que tiene desarrollar un dispositivo que no represente una catástrofe. Además de esta idea, que es apoyada por grandes empresas, también se han realizado proyectos de investigación en otros tipos de industria como en el campo de construcción [7], aplicando este tipo de robots para colaborar en procesos de construcción

De esta manera, el desarrollo de este proyecto trata de entender el modelo dinámico de un dispositivo aéreo “drone”, además de presentar modelos diferentes al modelo de control clásico que es usado comúnmente para este tipo de proyectos, además de realizar el proceso mecánico de diseño y fabricación del dispositivo y, por último, poder hacer pruebas en simulación donde se puede observar el comportamiento del sistema.

1.1. PLANTEAMIENTO DEL PROBLEMA

La implementación de robots móviles aéreos ha sido un campo de investigación muy grande a nivel mundial, inclusive algunas empresas los están implementando poco a poco como medio de transporte personal [8], pero, aunque estemos en el año 2020 y los avances tecnológicos están siendo parte de nuestra vida cotidiana, este tipo de proyecto no se pueden dar a la ligera.

Diseñar, construir, controlar un dispositivo aéreo siempre ha sido un gran reto para la comunidad académica, y es que una mínima falla puede afectar en gran manera el sistema, por eso mismo los aviones cuentan de sistemas robustos que permiten hacer un mejor control del sistema, teniendo incluso hasta cuatro sensores para una misma tarea [9].

Los drones son una nueva generación de dispositivos aéreos tele-operados, comúnmente estos dispositivos están desarrollados con un modelo de control clásico que permite desarrollar el desplazamiento de muy buena manera, este desplazamiento es de acuerdo a la señal deseada por el usuario, ya sea desde un *joystick*, señales neuronales [10], un radiocontrol de x canales, una trayectoria previamente programada, etc. Pero este controlador realmente no es viable en todas las aplicaciones que se pueden desarrollar en el dispositivo, esto es debido a que el desarrollo del controlador incluye un modelo del dispositivo, y los parámetros de intrínsecos del sistema como el peso afectan el controlador de posición implementado. Si el modelo difiere de los rangos permitidos para los cuales el controlador ha sido diseñado, el desempeño del sistema se degrada, por ejemplo, puede consumir más energía, desviarse de la trayectoria deseada, incluso perder total control del sistema lo cual puede ser catastrófico.

Considerando que en aplicaciones reales el drone está sometido a parámetros inciertos, como el peso que debe cargar, las perturbaciones del entorno como viento o a variaciones en la energía suministrada debido a las baterías, etc. Es necesario diseñar prototipos para realizar el estudio de ellos y proponer algoritmos de control que permitan al drone compensar estas limitaciones.

1.1.1 ANTECEDENTES

El desarrollo del proyecto se efectúa en la Universidad de Pamplona donde se busca toda la información acerca de un drone que sea de ayuda en el desarrollo del proyecto, tal como: diseño, fabricación, diseño de controladores, implementación, áreas de aplicación, etc. En la universidad se tiene el desarrollo de un controlador de alto nivel aplicado a un drone [11], y un controlador de bajo nivel para el control de postura, en ninguno de ellos se busca desarrollar una aplicación que tenga un ajuste en el cambio de su peso, ni implemente un controlador adaptativo.

En lo referente al diseño mecánico de este tipo de dispositivos, no se encuentra información desarrollada en la Universidad así que se consulta la información referente a diseños realizados que sean de software libre. También es necesario desarrollar una base de pruebas para el dispositivo debido a que no se tiene conocimiento de la implementación del controlador, por lo que se investiga en los prototipos desarrollados por la empresa Quanser para la prueba de los parámetros del controlador de postura.

Por la parte del control, se encuentra poca información acerca de desarrolla un controlador completo para este tipo de dispositivo, en la universidad se realizó un proyecto para el ajuste del controlador [12], pero no para el sistema en conjunto, el cual necesita controlador de posición. Se busca información referente a la implementación del controlador en conjunto, pero la mayoría especifican información de software libre para implementar en ciertas tarjetas de control, o es aplicado en bases de pruebas complejas de fabricar [13], lo cual limita a entender cómo funciona el sistema de control de bajo nivel. Con bajo nivel se refiere al sistema que permite controlar los motores, teniendo en cuenta la señal de los sensores de distancia del dron. Un control de alto nivel hace referencia al controlador que se encarga de guiar al dron en un espacio para que siga una trayectoria específica.

En cuanto a la información del controlador se realiza el curso de Aerial Robotics de los cursos en línea, de la plataforma Coursera, donde especifica toda la información del modelo dinámico y las ecuaciones del sistema. Además de contar con una *toolbox* de Peter Corke de MATLAB [14] la cual cuenta con diferentes modelos robóticos en los que se puede realizar su simulación.

Por su parte, en la implementación del control adaptativo se conoce que es difícil para este tipo de dispositivos que tienen funcionalidad de transporte de elementos, conocer la masa, ubicación del centro de masa, y matriz de inercia. En la documentación investigada [15], usan el método de “cable *force calculator*” o calculador de fuerza de cable para calcular el peso del dispositivo, resulta de gran ayuda, debido a que tienen varios dispositivos levantando un peso, pero para este caso es solo un dispositivo al cual se realizara el cambio de peso, lo más cercano a centro de masa para que no ocurra una gran variación en demás parámetros físicos del sistema.

1.1.2. JUSTIFICACIÓN

El desarrollo de un modelo de control adaptativo permite solucionar el problema de ¿Cómo ajustar el controlador cuando el drone contiene un valor diferente en la planta al que fue diseñado su controlador?, ya que este dispositivo dependiendo de su aplicación o el entorno puede generar variaciones en las constantes del controlador de posición implementado, lo cual mediante la implementación del controlador adaptativo permite que se implementen n cantidad de controladores. Por su parte en la documentación investigada en la Universidad de Pamplona [11]. [16], [12] no se encuentran estrategias de control adaptativo aplicadas a este tipo de dispositivos por lo que se empieza a investigar en otras fuentes acerca de este tipo de control.

Mediante este controlador se plantea entender cómo funciona nuestra planta mediante el software de simulación planteado, donde se modifican el parámetro de la planta, como lo es el peso. Luego se procede a implementar esta información en una red neuronal, la cual va a ajustar los parámetros de control de posición mediante un tipo de conmutación, ya que esta funciona de acuerdo a la información con la que se entrena.

En el área de la investigación de drones, es fundamental tener una configuración experimental que permita realizar pruebas de algoritmos que permitan solucionar este tipo de contingencias, es por esto que se plantea en esta tesis proponer un prototipo funcional para el desarrollo de pruebas experimentales sobre un drone que permita investigar en estrategias de control del drone a bajo nivel donde se pueda controlar el sistema en un rango dinámico amplio y que permita salvaguardar la integridad del prototipo y a su vez se puedan desarrollar algoritmos de control mediante diferentes.

1.2. DELIMITACIÓN

1.2.1. OBJETIVO GENERAL

Construir un prototipo de un drone para el desarrollo de estrategias de control.

1.2.2. OBJETIVOS ESPECÍFICOS

- Realizar el diseño mecánico en el software CAD.
- Realizar el estudio de elementos finitos al diseño mecánico.

- Desarrollar el modelo cinemático del dron.
- Diseñar un controlador para el vehículo aéreo no tripulado que se adapte a desviaciones de un modelo nominal.
- Realizar la simulación aplicando el modelo de control diseñado.
- Construcción del prototipo.

2. MARCO TEÓRICO Y ESTADO DEL ARTE.

2.1. MARCO TEÓRICO

El desarrollo de vehículos aéreos no tripulados, han tenido un gran auge a partir del siglo XX donde la implementación de estos vehículos en la segunda guerra mundial era un factor determinante. A partir de este momento se buscaba la implementación de un control a este sistema como herramienta de prueba para ser usado posteriormente por soldados en la guerra. Luego, gracias a la tele-operación se empezaron a usar estos dispositivos de diferentes formas, como lo era el reconocimiento al campo enemigo y traer esa información a la base militar, donde se realizaría el proceso de decodificación de las imágenes para desarrollar estrategias de guerra. Al pasar de los años se implementaron nuevos modelos de vehículos aéreos no tripulados, en ellos se implementó la idea de desarrollar un modelo de despegue vertical VTOL (vertical take-off and landing; aeronave de despegue y aterrizaje vertical), aplicado en el dispositivo Yamaha R50. Siguiendo esta idea, pero aumentando su control se desarrollaron múltiples modelos de vehículos aéreos no tripulados donde se encuentra uno de ellos llamado “*Quadrotor*” llamados “dron”, o también llamados helicópteros de cuatro rotores, el cual se va a abordar en el desarrollo del proyecto. El dron es un helicóptero con cuatro rotores para su sostén y su propulsión. Los cuatro rotores están generalmente colocados en las extremidades de una cruz. A fin de evitar que el aparato se tumbe respecto a su eje de orientación es necesario que dos hélices giren en un sentido y las otras dos en el otro sentido.

La robótica aérea, tiene una gran variedad de aplicaciones en grandes industrias, en ellas se encuentra el sector seguridad y vigilancia, donde incluso en Colombia ya se están

implementando este tipo de proyectos, utilizándose por parte de la policía colombiana drones tele-operados para colaborar en persecuciones o redadas brindando un apoyo aéreo, y conocer cualquier tipo de escape de presuntos criminales [7].

Por otra parte, en la industria civil estos nuevos proyectos de robótica tampoco se quedan atrás, en ellos se busca aplicar este tipo de dispositivos en pequeños sectores de la industria, colaborando en el proceso de monitoreo [17], análisis de terrenos [18], etc.

Uno de los sectores que ha tenido un mayor auge y aceptación de este tipo de dispositivos es la industria de artes visuales, donde se usan tanto en la grabación de escenas, tomas de fotos panorámicas, entre otros. Su gran auge se debe en gran medida el gran crecimiento que han tenido las cámaras fotográficas en la última década, y en el alto consumo audiovisual de las personas mediante las diferentes redes sociales.

2.1.1. TIPOS DE CONFIGURACIÓN

La clasificación de los drones es un proceso que se puede ajustar dependiendo de varios factores, por ejemplo: algunos hacen esta clasificación de acuerdo al peso, otros de acuerdo al tipo de aplicación, otros al tiempo de operación, etc. En la investigación que se realiza el objetivo es caracterizar el modelo del dispositivo y tener en cuenta la aplicación, ya que es

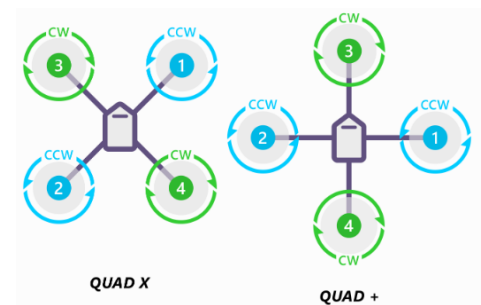


Ilustración 1 Configuración “X” y “+”, Fuente: Ardupilot

un dron de transporte, pero es de carga media (alrededor de 1kg-1.2kg) este puede implementarse en el área de agricultura o en la industria de transporte como repartidor a domicilio. También se debe tener en cuenta otra clasificación muy importante que interfiere totalmente en la implementación del controlador y es el tipo de configuración o la ubicación en la que se ubican los motores con el sistema de referencia del dispositivo. En estos dos tipos importantes que son la configuración “X” y “+”, como se observa en la Ilustración 1.

Software de diseño

El software de diseño, también llamado CAD, es un programa en computador especializado en el diseño, ensamble de piezas en entornos 3D. Los más reconocidos en la universidad son Autodesk Inventor, Autodesk Fusión, y Solidworks del cual se cuentan licencias para desarrollar diseños en este programa por parte de la Universidad. Este último presenta la gran ventaja de tener una interfaz sencilla y amigable para realizar el proceso de diseño.

Análisis de elementos finitos

El análisis de elementos finitos es una investigación que está surgiendo en la prefabricación de dispositivos, gracias a la computadora digital, ya que este, mediante el desarrollo de ecuaciones algebraicas [19], establece parámetros físicos como: análisis de estrés, análisis estático, análisis térmico, entre otros.

Se establece así que el análisis de elementos finitos se basa en la obtención de los datos físicos de las piezas, propiedades físicas como densidad, módulo de elasticidad, módulo de flexión, temperatura de fusión, Esfuerzo de tracción a la deformación, entre otros. Y estos datos son operados por el software CAE para estimar la respuesta de la pieza cuando estas realizan la actividad para la que fueron diseñados. Este análisis establece el desempeño mecánico de la pieza, y permite determinar si la pieza es viable antes de realizar el proceso de fabricación.

Para el proceso CAE, se utiliza el programa Fusión, el cual presenta una interfaz más amigable con el desarrollo del proceso, y además cuenta con la facilidad de lectura de una gran variedad de formatos de piezas desarrolladas en diferente software de diseño.

Microcontroladores

Los microcontroladores son una gran revolución tecnológica y es que son *microchips* que permiten resolver operaciones en cuestiones de milisegundos o menos. Y este proyecto no es ajeno a este tipo de dispositivo, ya que aparte del desarrollo de computadoras delgadas y dispositivos pequeños, también se están desarrollando tarjetas de control con microcontroladores para el área de robótica, lo que permite tener el control de un motor conectando unos cables y ejecutando unas líneas de programación. Hay una gran variedad de efectores finales, como motores, luces, servomotores, motores paso a paso, etc. Y también se

busca implementar una gran gama de sensores que aumente el área de trabajo de estas tarjetas de control en la industria, haciendo más fácil la automatización de decenas de procesos.

2.1.2. PARTES DE UN DRONE

Un drone es un dispositivo móvil, que consta de 6 o más piezas, y aunque suele ser un proyecto muy común sus partes no siempre son las mismas. En los dispositivos principales se encuentran: la tarjeta de control, los motores, las hélices, la estructura, sensor IMU, dispositivo de tele operación. Aunque algunos dispositivos varían algunos de estas, ya que contienen integrado el sensor IMU en la tarjeta de control, o incluso también cuentan con el dispositivo de tele operación mediante WIFI. En esta sección se habla con detalle de las partes más importantes, además de las otras que se añadirán para realizar los objetivos del proyecto.

1. **Batería:** Principal y única fuente de suministro de energía del sistema. Su gran importancia radica en que es la que limita la cantidad de revoluciones del motor, dependiendo de su número de celdas, y la constante de descarga. Además de restringir el tiempo de vuelo de acuerdo a su capacidad de almacenamiento.
2. **Tarjeta de control:** Principal unidad de procesamiento, es el cerebro del proyecto y la parte que toma los datos de los sensores (IMU, Ultrasonido), y envía señales a los motores para realizar los movimientos.
3. **Estructura:** Normalmente la estructura de un drone suele ser adquirida, debido a que viene compacta, sin problemas mecánicos, y ligera. Esta estructura consta de cuatro brazos donde se sujetan los motores, un soporte superior y un soporte inferior.
4. **ESC:** Son controladores de velocidad electrónicos que reciben la señal de control de la tarjeta, procesa esta información y realiza el suministro de energía a el motor.
5. **Motor *Brushless*:** Los motores *brushless* son los encargados de realizar el movimiento mecánico de acuerdo a la energía suministrada desde los ESC.
6. **IMU:** unidad de medición inercial, este dispositivo es el encargado de enviar la señal al controlador, de la inclinación en la que se encuentra el dispositivo para una posterior acción de control.
7. **Hélice:** Este elemento es el encargado de realizar el empuje inicial sobre el eje de rotación del motor.

8. Sensor proximidad: Este tipo de sensor, a diferencia del IMU, tiene como objetivo definir la posición del dispositivo, o detectar objetos para evitar colisiones durante su operación.

2.1.3. DINÁMICA

La intención del desarrollo del modelo matemático consiste en obtener el modelo en simulación para realizar pruebas en un software de simulación y así poder evitar posibles daños en la estructura física del dispositivo. Para entender el desarrollo de la dinámica implementada en el dispositivo, es necesario entender la Segunda ley de Newton para un sistema de partículas, la cual dice: El centro de masa para un sistema de partículas “S”, aceleradas en un sistema de referencia “A”, analizadas como una partícula singular de masa m (equivalente al total de la masa del sistema) actuado por una fuerza igual a la fuerza neta está dada en la ecuación $r_c = \frac{1}{m} \sum_{i=1,N} m_i p_i$ (1).

$$r_c = \frac{1}{m} \sum_{i=1,N} m_i p_i \quad (1)$$

Y la ecuación de Newton quedaría representada en la ecuación $F = \sum_{i=1}^N F_i = m \frac{A_d A_v^c}{dt}$ (2) (2):

$$F = \sum_{i=1}^N F_i = m \frac{A_d A_v^c}{dt} \quad (2)$$

Donde “v” corresponde a la velocidad del cuerpo en el sistema de referencia A.

Y la tasa de cambio del momento lineal está dada por la ecuación $F = \frac{A_d L}{dt}$ (3):

$$F = \frac{A_d L}{dt} \quad (3)$$

Donde “L” es el momento lineal del sistema de partículas en el sistema de referencia “A”.

También está la ecuación el momento angular, la cual está dada por la ecuación $A_{H_C}^S = I_C * A_{\omega^B}$ (4):

$$A_{H_C}^S = I_C * A_{\omega^B} \quad (4)$$

Donde “I” es el tensor de inercia con C como el origen.

Ahora, la tasa de cambio del momento angular del cuerpo rígido “B” con respecto a “C” en el sistema de referencia “A” es igual a la resultante de momentos de todas las fuerzas externas actuando sobre el cuerpo con respecto a “C”, como se muestra en la ecuación (5).

$$\frac{A_d A_{H_c}^S}{dt} = M_c^S \quad (5)$$

Se calcula el momento debido a las fuerzas externas y torques con respecto a “C” para el cuerpo “B”, como se ve en la Ilustración 2.

$$\frac{A_{dH_c}}{dt} = M_c \quad (6)$$

Diferenciando se tiene que:

$$\frac{B_{dH_c}}{dt} + A_{\omega_B} \times H_c = M_c \quad (7)$$

Reemplazando el valor de $\frac{B_{dH_c}}{dt}$:

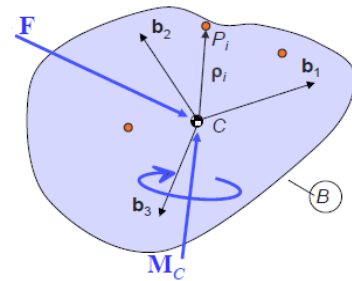


Ilustración 2 Fuerza y Torque sobre un sistema de referencia. Fuente: Coursera

$$\frac{B_{dH_c}}{dt} = [I_{11} \ 0 \ 0 \ 0 \ I_{22} \ 0 \ 0 \ 0 \ I_{33}] [\omega_1 \ \omega_2 \ \omega_3] \quad (8)$$

Y de H_c :

$$H_c = [I_{11} \ 0 \ 0 \ 0 \ I_{22} \ 0 \ 0 \ 0 \ I_{33}] [\omega_1 \ \omega_2 \ \omega_3] \quad (9)$$

Por otra parte el valor de A_{ω_B} es:

$$A_{\omega_B} = \omega_1 b_1 + \omega_2 b_2 + \omega_3 b_3 \quad (10)$$

Así la ecuación (7) resultaría de la siguiente manera:

$$[I_{11} \ 0 \ 0 \ 0 \ I_{22} \ 0 \ 0 \ 0 \ I_{33}] [\omega_1 \ \omega_2 \ \omega_3] + [0 \ -\omega_3 \ \omega_2 \ \omega_3 \ 0 \ -\omega_1 \ -\omega_2 \ \omega_1 \ 0] [I_{11} \ 0 \ 0 \ 0 \ I_{22} \ 0 \ 0 \ 0 \ I_{33}] [\omega_1 \ \omega_2 \ \omega_3] = [M_{c,1} \ M_{c,2} \ M_{c,3}] \quad (11)$$

Quedando la segunda parte de la sumatoria con un valor igual a 0.

$$[I_{11} \ 0 \ 0 \ 0 \ I_{22} \ 0 \ 0 \ 0 \ I_{33}] [\omega_1 \ \omega_2 \ \omega_3] + 0 = [M_{c,1} \ M_{c,2} \ M_{c,3}] \quad (12)$$

Resulta la siguiente ecuación:

$$\frac{B_{dH_c}}{dt} = I_{11} \omega_1 b_1 + I_{22} \omega_2 b_2 + I_{33} \omega_3 b_3 \quad (13)$$

Teniendo en cuenta este análisis se determinan las ecuaciones de movimiento del dron. Para ello primero se establecen las ecuaciones de fuerza y momento,

$$F_i = K_f \omega_i^2 \quad (14)$$

$$M_i = K_m \omega_i^2 \quad (15)$$

Dónde: ω_i : Velocidad angular de los motores.

K_f : Corresponde a la constante de los motores

K_m : Corresponde a la constante de los motores

F_i : Fuerza ejercida por cada motor

M_i : Momento ejercido por cada motor

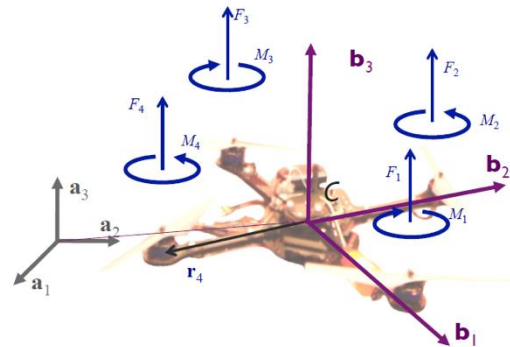


Ilustración 3 Fuerzas y torques aplicadas a un dron. Fuente: Coursera.

Así la segunda ley de Newton quedaría de la siguiente manera:

$$F = F_1 + F_2 + F_3 + F_4 - m g a_3 \quad (16)$$

Y la ecuación de momento:

$$M = r_1 \times F_1 + r_2 \times F_2 + r_3 \times F_3 + r_4 \times F_4 + M_1 + M_2 + M_3 + M_4 \quad (17)$$

Expresado en ecuación de espacio de estados quedaría:

$$m \ddot{r} = [0 \ 0 \ -m g] + R [0 \ 0 \ F_1 + F_2 + F_3 + F_4] \quad (18)$$

$$I [p \ q \ r] = [L (F_2 - F_4) \ L (F_3 - F_1) \ M_1 + M_2 + M_3 + M_4] - [p \ q \ r] \times I [p \ q \ r] \quad (19)$$

Donde la ecuación en espacio de estados de los componentes de la velocidad angular es:

$$[p \ q \ r] = \begin{bmatrix} \cos(\theta) & 0 & -\cos(\phi) \sin(\theta) & 0 & 1 & \sin(\phi) \sin(\theta) & 0 \\ \cos(\phi) \cos(\theta) & \cos(\phi) \sin(\theta) & \sin(\phi) \cos(\theta) & \sin(\phi) \sin(\theta) & 0 & 0 & 0 \end{bmatrix} [\dot{\phi} \ \dot{\theta} \ \dot{\psi}] \quad (20)$$

2.2. ESTADO DEL ARTE

En los diseños investigados [20]–[22] se encuentran la empresa Dji la cual es la principal empresa de fabricación y venta de drones a nivel mundial, reconocida por su alta gama de

dispositivos, donde varían desde su tamaño, tiempo de operación, funcionamiento, entre otros aspectos. Antes de desarrollar grandes prototipos como el Dji Mavic Mini o el Mavic 2 Pro, ellos diseñaron estructuras para implementar cualquier tipo de controlador y ensamblar el circuito de potencia y de control del mismo. A estos se les conoce como F330, F450 y F500, cada uno con el diseño semejante al otro, variando principalmente su tamaño, dependiendo de su aplicación.

Normalmente en el proceso de fabricación se guía de la aplicación para definir el tamaño de la estructura, si necesita levantar un gran peso, es necesario un gran dispositivo, que permita el uso de grandes hélices y una gran batería para su funcionamiento, o si necesita algo pequeño que cuente solo con una cámara de alta resolución, puede servir la estructura F330 [23]. El tamaño también presenta una característica mecánica y es que tiene una respuesta más rápida en cuanto a la variación en sus ángulos pitch y roll más rápida, pero a su vez es contra productiva en el movimiento en general del dispositivo, esto quiere decir que un dispositivo que es más grande entre las longitudes de sus brazos tiene una mayor aceleración angular, por lo que el controlador de postura debe estar bien sintonizado para que no ocurra una catástrofe con el dispositivo.

También se diseña una base de pruebas que resulta fundamental implementar si es un controlador que se desarrolla desde cero, así que lo mejor es crear el diseño y hacer pruebas mediante este dispositivo. La empresa QUANSER ha desarrollado el dispositivo 3 DOF Hover [24], el cual cuenta con 3 grados de libertad rotacionales, montado sobre un pivote como se muestra en la Ilustración 34 e Ilustración 35, este dispositivo representa para el proyecto una buena idea para realizar las pruebas del algoritmo de control de postura [25] [26]. Mediante el uso de la fuerza que proveen las hélices de los motores se pueden generar variaciones en los ángulos de cabeceo, balanceo y de orientación. Estas orientaciones serán visualizadas en el software Simulink donde se realizará la simulación del movimiento del vehículo aéreo no tripulado.

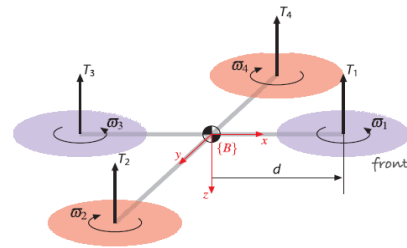
Al pasar de los años se han implementado nuevos controladores para diferentes modelos de robots aéreos [27], en este caso es bueno entender el modelo matemático de un drone en el cual se aplica un controlador cinemático, considerando así el modelo de cuerpo rígido y sus

Ilustración 4 Velocidades angulares y fuerzas aplicadas a un drone configuración X. Fuente: Coursera.

características físicas, teniendo estos parámetros en cuenta se desarrolla finalmente el análisis de movimiento que permiten al dispositivo moverse en su espacio de trabajo y girar en cada uno de sus ejes como se muestra en la Ilustración 4.

Inicialmente se busca en el estado del arte de la Universidad de Pamplona, donde se tiene la información de drones desarrollados con aplicaciones de agricultura [16], seguidores de trayectoria para aplicaciones de vigilancia [11], diseño de controladores de postura y análisis del modelo dinámico [12], pero en ninguno de ellos se implementa un control de bajo nivel que sea adaptativo, por lo que la información proporcionada por estas investigaciones sirven para el proceso de consultar recursos, información básica de operación, etc.

Por su parte, los desarrollos de los controladores en investigación se han venido ideando de acuerdo a las necesidades de las propias aplicaciones en las que se están implementando estos dispositivos. Principalmente, estos proyectos de investigación son realizados para el transporte de objetos [15], operaciones de búsqueda y rescate [28], asistencia remota [29], entre otros. Y es que en la implementación de un buen controlador es uno de los pilares más importantes en el desarrollo de un sistema, este implica variar las velocidades de los motores para obtener una inclinación en cierto eje y así poder desarrollar una trayectoria tridimensional del dispositivo. El desarrollo de un controlador clásico es uno de los más comunes para el control de este sistema, pero, aunque su desempeño es óptimo a modelos lineales este presenta inconvenientes cuando sus parámetros de operación varían. Esto lleva a desarrollar un nuevo controlador que permita el control del sistema aun teniendo incertidumbres paramétricas.



En la aplicación de transporte de objetos que es el objetivo de este proyecto, se observan diferentes tipos, en los que se encuentran: grupo de drones funcionando de manera colaborativa entre ellos [30], dispositivos individuales movimiento una masa específica [15], etc. Y dentro de cada uno de estos proyectos se observan diferentes estrategias de control, donde implementan algoritmos Fuzzy [30], redes neuronales [31], Controlador de H^∞ [32]. Para ello es necesario consultar los controladores adaptativos, que buscan adaptar el controlador teniendo en cuenta propiedades físicas que afectan drásticamente el control del dispositivo. En [33], [34], se observa el funcionamiento de un controlador Gaussiano lineal cuadrático, mediante el cálculo del ángulo de oscilación de la carga transportada por el dispositivo, la cual es sujeta por medio de un péndulo. El proceso es que se estima la masa utilizando mínimos cuadrados recursivos y también se estima la longitud del cable a partir de la frecuencia oscilante obtenida con la Transformada de Fourier. Con lo cual se obtienen resultados bastante buenos, donde hasta disminuye las oscilaciones causadas por el objeto colgante para el dispositivo. En otro proyecto similar [15], se sujeta la masa que se desea transportar a cables conectados de forma serial, en este se implementa un controlador de costo retrospectivo para compensar la incertidumbre de la masa desconocida.



Ilustración 5 Modelo quanser 3 DOF. Fuente: Quanser.

En otros proyectos se maneja no solo uno, sino varios drones con el cual se busca tener una mayor capacidad para levantar peso [15], pero en el momento de implementar un controlador se vuelve más complejo ya que todos tienen que mantener una conexión entre ellos. Estos llevan la carga mediante cables, en los cuales cada uno realiza el análisis de fuerza, el primero menciona en su modelo dinámico tanto el peso de los cables flexibles como la masa del sistema y el segundo utiliza un calculador de fuerza en cada uno de los cables que se conectan.

3. INGENIERÍA DE PROYECTO

La ingeniería de proyecto consta de las ideas que se plantean para solucionar los objetivos planteados, teniendo en cuenta que estos se correlacionan entre sí. Para ello, construir un dispositivo requiere de diseñar algo físico, y aunque puedas tener un simple diseño que pueda acoplarse y ser liviano algunas veces no basta. Se es necesario que la unión entre las piezas sea lo suficientemente fuerte para soportar el proceso de vuelo. Para ello se usa un recurso que es usado comúnmente en la fabricación de dispositivos, que es el análisis de elementos finitos, del cual se pueden obtener resultados de diseño, ejerciendo simulaciones de fuerza y observando las respuestas del sistema.

Inicialmente se busca toda la información de los diseños de este tipo de dispositivos, entre ellos los más importante son los de software libre, o los que vienen para armar sin ninguna tarjeta de control integrada u otro dispositivo electrónico, así que la gama de “F” de DJI es la más recomendable para desarrollar el objetivo de diseño. Uno de los diseños más importantes es el

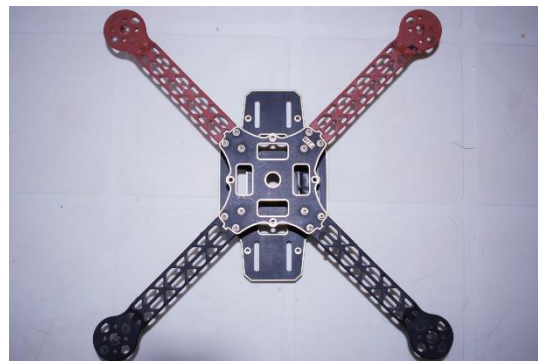


Ilustración 6 Estructura mecánica F330. Fuente: Autor.

F330, como se muestra en la Ilustración 6 que tienen una gran diferencia, la cual radica

principalmente en su tamaño, el cual permite utilizar unas hélices más grandes que la gran mayoría de diseños de mediano costo, pero esto significaría un consumo energético mayor.

Por otra parte, se tiene una gran información acerca de controladores de alto nivel en drones, pero se tiene muy poca información acerca de la información a bajo nivel, de hecho, resulto necesario realizar un curso en la página de cursos Coursera [1], llamado Robotics: Aerial Robotics, donde explican minuciosamente las ecuaciones de control, sistemas de referencia, matrices de transformación, entre otros temas relevantes en el desarrollo del control. Presentado por la Universidad de Pensilvania y dirigido por el profesor Vijay Kumar.

Luego de establecer el diseño y conocer la dinámica y ecuaciones de control del sistema se procede a realizar la implementación, y es que el drone no depende solo de la estructura y la tarjeta de control, si bien son partes importantes, también es necesario que los sensores y actuadores. En los cuales se encuentra una gran gama, en cuanto a tamaño y potencia, pero nuestra mayor limitante en este elemento resulta ser el recurso económico y en la poca variedad de estos tipos de motores en Colombia, por lo tanto, se observan proyectos realizados en la Universidad [11], [12], [16], con el fin de buscar motores que sean asequibles y fáciles de encontrar en el país, se realizan pruebas de consumo energético y de empuje con diferentes tipos de hélices encontrados en la sección 0 y 0.

Luego de esto, teniendo el sistema y el controlador clásico desarrollado, se realiza el control adaptativo, para el cual primero se observa el comportamiento del sistema cuando la masa es variable, para ello se modifica el modelo dinámico y se analiza la respuesta del sistema en estado transitorio y estacionario para establecer la estrategia de control adaptativo, con la idea de adaptar el controlador cuando la masa del sistema varié durante la operación del drone. Se consideran parámetros variables acotados en valores, específicamente el peso podrá variar en dos valores conocidos (1kg y 1.2kg).

3.1 DISEÑO MECÁNICO

La construcción de un drone es un proceso bastante amplio, en el cual debe desarrollarse cuidadosamente cada uno de los subprocesos. En donde ningún subproceso está desligado de otro, incluso aunque tenga un desarrollo de controlador óptimo, si la parte mecánica tiene variaciones, hace que este controlador, no funcione de la mejor manera. Y es que, hay que

tener en cuenta que el desarrollo mecánico no es un subproceso netamente de diseño, sino también se debe tener en cuenta que cada que se diseña una pieza, se debe analizar su modelo mecánico, ya que van a soportar esfuerzos, tensiones, desgaste, y tiene que ser compacto.

Lo primero es diseñar sin sobredimensionar las piezas, una buena forma de empezar a crear un boceto es mirando otros bocetos y observar las propiedades mecánicas más simples, las longitudes de piezas y el peso. ¿Porque resultan estas dimensiones tan importantes a la hora de empezar a diseñar?, esto es debido a que las longitudes de los brazos pueden limitar la dimensión de las hélices, y hay que tener mucho en cuenta esta pieza en el prototipo. Las hélices son las que van a realizar la fuerza de empuje por cada motor, entre más grande sea esta, más empuje ejercerá. También se debe tener en cuenta que, así como aumenta el empuje, aumenta el consumo eléctrico en los motores debido al aumento del par, lo cual es inversamente proporcional al tiempo de operación del dron. Y es que el tiempo de operación se ha convertido en uno de los aspectos más importantes, consecuencia de la gran cantidad de aplicaciones con las que cuenta hoy en día este tipo de dispositivos.

Entonces como conclusión, es necesario ejercer una buena relación entre consumo eléctrico, dimensionamiento y propiedades físicas, haciendo primero el análisis de consumo eléctrico de los motores con los diferentes tipos de hélices que se usaran.

3.1.1. Selección de componentes

La selección de componentes resulta ser uno de los procesos más importantes al momento de querer crear un dron, por eso era necesario hacer la base de pruebas de 1 GDL, ya que con esos resultados se puede empezar a tener una idea de que elementos con qué dimensiones son necesarios para que el dispositivo y el desarrollo del proyecto no tenga inconvenientes.

Motor *Brushless*

El motor *brushless* es uno de los más importantes, es un tipo de motor que presenta una característica importante para el desarrollo de vehículos aéreos no tripulados, y es que permite obtener grandes velocidades al no contar con escobillas para el cambio de su polaridad. Permitiendo según sus modelos realizar



Ilustración 7 Motor Brushless A2212 1000KV. Fuente: Autor.

una gran cantidad de revoluciones, en este caso de 5400 rpm-8000 rpm, de acuerdo a la señal de control y a la energía suministrada.

Al implementar un motor *brushless*, se cambia la forma de conexión con referencia a los motores de corriente continua comunes, ya que cambia de ser una forma de conexión con escobillas a una en la que el motor realiza su alimentación por medio de dispositivos semiconductores. Y no solo cambia el sistema de alimentación, si no que los motores *brushless* también realizan una conmutación por fuera del motor de forma estática y electrónica a diferencia de los motores de corriente directa [35], además dada su diferencia en el tipo de conexión, no cuenta con dos cables como los tienen los motores comunes, si no que cuenta con tres.

Controlador de Velocidad Electrónico

Para el control de los motores *brushless*, se necesita obligatoriamente implementar el controlador de velocidad electrónico (ESCS por sus siglas en inglés), debido a que estos son los que efectúan la variación de la velocidad de dichos motores. Para la selección de este componente, es necesario conocer las propiedades eléctricas de los motores



Ilustración 8 Esc 30A. Fuente: Autor.

brushless, medidas mediante las pruebas de la estructura de 1 gdl, la Tabla 7 a la Tabla 10 (página 48 y 48) se muestra la información acerca del consumo de corriente de los motores *brushless*, donde se toma el valor máximo como referencia para buscar el dispositivo.

Este dispositivo permite controlar los motores *brushless* mediante la variación de la señal de control cíclica enviada cada 20 ms, donde su señal en alto oscila entre 1ms y 2ms, que se puede interpretar como la velocidad mínima y la velocidad máxima que puede alcanzar el dispositivo. Al ser un dispositivo que se controla igual que un servo, se desarrolla la librería para controlar los motores *brushless* desde Simulink.

Tarjeta de Control

La implementación del microcontrolador es uno de los aspectos donde se puede tener más información, debido a la gran cantidad de tarjetas de control en las que se puede implementar el sistema de control para el manejo de un drone, entre ellas existen una gran variedad de precios desde la ESP8266 [36] hasta tarjetas de gran alcance computacional como la gama de Raspberry pi, BeagleBoard, etc.



Ilustración 9 Beaglebone Blue. Fuente: Autor

En este caso, es necesario tener una tarjeta con gran capacidad computacional, ya que se quiere implementar un control adaptativo, además de que es necesario que pueda conectarse fácilmente con el operador y sea lo más pequeña y compacta posible.

Batería LiPo

Las baterías LiPo son la fuente de alimentación más comunes usadas en este tipo de proyectos, y esto es debido a varios aspectos que les favorece, como su fácil accesibilidad, su bajo coste, y la más importante: sus características eléctricas, que permiten obtener una tasa de suministro de corriente mucho mayor a la de las baterías convencionales. Esto es debido a la constante de



Ilustración 10 Batería LiPo. Fuente: Autor

descarga con la que cuentan este tipo de baterías, que en la mayoría de los casos es mayor de 20C, lo cual significa que puede suministrar 20 veces la capacidad de corriente que almacena.

Y es que la corriente que consumen los motores es un valor considerable. Como se observa la Tabla 4 puede alcanzar valores de 6A, un valor exagerado si se tiene en cuenta que los cuatro motores se conectan a una sola batería, por lo tanto, esta debe tener una capacidad de descarga de 24A.

Hélices

Las hélices, como se comentaba anteriormente, establecen una relación directa entre el consumo energético y la capacidad de empuje por motor, mientras las hélices sean más grandes, el empuje es mayor, pero también afecta el tiempo de operación ya que consumen una mayor corriente, por ello hay que tener cuidado en la selección de este componente.



Ilustración 11 Hélice 1045. Fuente: Autor

Sensor de proximidad

Los sensores de proximidad suelen ser un caso específico en la implementación de este tipo de dispositivos. Normalmente son usados en proyectos donde el dispositivo puede esquivar obstáculos o necesita moverse en un entorno cerrado. Se pueden encontrar sensores de proximidad de muchos tipos:

Lidar: Laser Imaging Detection and Ranging, es un dispositivo electrónico que emite señales de luz infrarroja, las cuales rebotan cuando chocan con un objeto y esto permite determinar la distancia a la que se encuentra. Este es uno de los dispositivos de medición más caros que se pueden encontrar, debido a su gran aplicabilidad y precisión.

Ultrasónico: El sensor ultrasónico o de ultrasonido, como su nombre lo indica, maneja emisiones de sonido para detectar los objetos, cuenta con la propiedad de encontrarse una vasta información para su debida conexión y programación, también suele tener un rango limitado de operación. Son de bajo costo, y además presentan la característica de no detectar los objetos si este se encuentra con cierto grado de inclinación.

Este tipo de sensor maneja dos tipos de conexión. Una es de cuatro pines, donde permite mediante una señal de activación, realizar el envío de la onda ultrasónica cuando se desee, y luego a través de otro pin poder captar la señal procesada después de chocar con un objeto, entonces para poder hacer el cálculo de la distancia a la que se encuentra el objeto, se debe hacer una conversión matemática, que permita hacer una relación entre la distancia y el tiempo, teniendo en cuenta que entre más tiempo demore en volver la onda ultrasónica, más lejos estará el objeto.

El otro tipo de conexión es mediante una señal de voltaje, la cual debe interpretarse de forma que, a mayor voltaje, mayor la distancia a la que se encuentra el objeto. En este caso no es necesario contar cuanto tiempo ha pasado y hacer una relación matemática, si no que el mismo dispositivo hace esa relación y genera un valor de voltaje que es el que está relacionado directamente con la distancia a la que se encuentra el objeto.

Infrarrojo: Este sensor al igual que el lidar, maneja señales de luz infrarroja, con la diferencia de que este tipo de dispositivos solo puede captar información de proximidad en objetos que estén frente a él, de esa forma su aplicación se ve limitada, aunque su bajo coste y basta información en proyectos de código abierto, lo convierten en uno de los sensores con mayor uso para proyectos de detección de objetos.

Cámara: Este sensor suele ser más completo, y es que se puede obtener más información de una imagen, que de una onda enviada a reconocer la distancia a la que está el objeto de enfrente. La imagen suele tener mucha información, y por eso mismo suele ser tan complejo adquirir la información importante de este dispositivo. Incluso hay toda una rama de estudio llamada Visión Artificial, que realiza el estudio de toda la información que contiene una imagen, también estudia cómo se puede analizar, filtrar, seleccionar elementos dentro de una imagen, etc. Por esto mismo se convierte en uno de los sensores más completos, pero a su vez, que requiere una gran capacidad computacional para realizar su cometido.

TABLAS DE SELECCIÓN DE COMPONENTES

Tabla de Selección de Motor

El proceso de selección de motores, es el proceso más importante, de él se derivan otros procesos de selección de componentes: como por ejemplo el de las hélices, los controladores electrónicos de velocidad, y el de la batería. También se debe tener en cuenta que deben ser asequibles ya que este elemento puede variar su precio, además de esto, suelen ser piezas con un alto riesgo a dañarse, debido a que las hélices no están protegidas y pueden chocar con cualquier objeto durante el vuelo. Por lo tanto, se debe seleccionar un dispositivo que pueda ser cambiado fácilmente

Se hace necesario obtener los datos de empuje de cada motor, para ello se analizan el datasheet de cada uno de los motores, o si no se tiene, se puede consultar en internet algunas

pruebas de empuje del motor. También es necesario tener estos datos aplicando las mismas hélices, por lo que se escogen las 8045, que consumen una baja corriente con respecto a la 9045 y 1045.

| Elemento | Características | Empuje (N) | Corriente (A) | Valor (\$) |
|----------------------------------|---|------------------------------|---------------|------------|
| Motor A2212/13T 1000KV | <ul style="list-style-type: none"> Fácil de reemplazar, debido a la gran cantidad de vendedores en Colombia. Alimentación de energía de 2S-3S-4S. Baja cantidad de KV, por lo que se recomienda hélices grandes. | 2.81 N | 4.1 | 20,000 |
| | | 3.4 N | 4.85 A | |
| | | 4.12 N | 5.65 A | |
| | | 4.86 | 6.5 A | |
| Motor XA2212 1400KV Emax | <ul style="list-style-type: none"> Fácil de reemplazar, debido a la gran cantidad de vendedores en Colombia. Se usa comúnmente en este tipo de proyectos. | 9.22 N | 20.6 A | 43,800 |
| D2204 2300KV | <ul style="list-style-type: none"> Puede realizar un empuje máximo de 420 gr, aunque no se especifica con que hélices ni a qué valor de corriente. Se consigue en Colombia. Alimentación de energía de 2S-3S-4S. | No se encuentra información. | | 55,200 |
| Readytosky MT2204 – 2300KV | <ul style="list-style-type: none"> Alimentación de energía de 2S-3S-4S. Se consigue en estados unidos en páginas como Amazon, bangood, entre otros. | 2.36 N | 8 A | 34,900 |
| | | 3.9 N | 10.5 A | |
| | | 4.68 N | 13.7 A | |

| | | | | |
|--------------------------|--|--------|--------|--------|
| Racerstar 2205 2300KV | <ul style="list-style-type: none"> ● Se consigue en Colombia. ● Alimentación de energía de 3S – 4S. ● Recomendable usar con hélices pequeñas. | 3.66 N | 7.27 A | 45,000 |
| | | 6.7 N | 18 A | |

Tabla 1 Características de los Motores brushless encontrados en el mercado. Fuente: Relacionada en cada motor en anexos 2.

Los datos acá planteados corresponden a información consultada de internet, se encuentra en anexos 2.

Los motores con alto KV, presentan la particularidad de usarse para dispositivos pequeños donde, se cuentan hélices de 6045 o más pequeñas, esto se ve limitado en el momento de implementar el proyecto debido al requisito de control de implementar la tarjeta BeagleBone blue y ahí implementar el controlador de bajo nivel.

Se observa que, en cuanto adquisición, los motores A2212/13T 1000KV son los más asequibles, cuentan con el empuje suficiente para el diseño planteado, además de tener una de las mejores relaciones de empuje/potencia, y contar con un gran número de vendedores en Colombia, lo cual permite reemplazar la pieza de manera más sencilla.

Teniendo en cuenta esto, se observa que las hélices recomendadas para este tipo de motor, son las 8045, 9045, y 1045, donde su principal diferencia es el tamaño, teniendo en cuenta que su primer número significa la dimensión de esta en pulgadas, siendo así de 8, 9 y 10 pulgadas respectivamente.

Tabla de Selección de Tarjeta de Control

Para el proceso de selección de las tarjetas de control se muestran todas las características de cada una de las tarjetas, ya sean a favor o en contra para implementación del proyecto y mediante estos datos, se tienen en cuenta para la selección dependiendo de lo que se requiera en el proyecto.

| Elemento | Características | Valor |
|----------|-----------------|-------|
|----------|-----------------|-------|

| | | |
|--------------------------------------|--|-------------------|
| <p>BeagleBone blue</p> | <ul style="list-style-type: none"> ● Fácil implementación del controlador desde Simulink. ● Análisis de datos adquiridos por los sensores en tiempo real. ● Sensores de giroscopio integrados ● Pines para control de motores <i>brushless</i>. ● Librería de robótica open source. ● Puerto USB para implantación de cámara. ● Buena capacidad de procesamiento y almacenamiento. ● Alimentación directa desde batería de 3S. ● Conexión a radio control. ● Conexión a dispositivo GPS. | <p>\$ 370.000</p> |
| <p>Raspberry pi model 4B-2Gb</p> | <ul style="list-style-type: none"> ● Implementación de controlador desde Simulink requiere desarrollo para comunicación con sensores y actuadores. ● Desconocimiento de librería de robótica. ● Sensores no integrados. ● Pines globales de entrada y salida de datos. ● Muy buena capacidad de procesamiento y almacenamiento. ● Alimentación directa desde la batería de 3S. ● Puerto USB para implementación de cámara. | <p>\$ 214,200</p> |
| <p>ESP 8266</p> | <ul style="list-style-type: none"> ● Implementación de controlador desde Simulink desconocido. ● Baja capacidad de procesamiento. ● Necesita adaptador de corriente para alimentarse desde la batería. ● Sensores no integrados. ● Tiempo de respuesta desconocido. | <p>\$ 20.000</p> |

| | | |
|-------------------|---|------------|
| | <ul style="list-style-type: none"> ● No tiene capacidad para implementar sistema de control adaptativo. | |
| Ardupilot Apm 2.8 | <ul style="list-style-type: none"> ● No se puede implementar controlador de bajo nivel. ● No es posible implementar un controlador adaptativo. ● Sistema de auto calibración de los motores. ● Análisis de datos adquiridos en tiempo real ● Buena respuesta del controlador. ● Es necesario un dispositivo radio control. ● Pines para control de motores <i>brushless</i>. ● Conexión a radio control. ● Conexión a dispositivo GPS. | \$ 229.500 |
| Odroid XU4 | <ul style="list-style-type: none"> ● Alta capacidad de procesamiento. ● Puerto USB para implementación de cámara. ● Sensores no integrados. ● Necesita adaptador de corriente para alimentarse desde la batería. ● Necesita un adaptador tipo CP2102 para controlar los motores <i>brushless</i> desde un puerto USB. | \$363,000 |

Tabla 2 Características de las tarjetas de control encontrados en el mercado. Fuente: Paginas de venta en internet.

Teniendo en cuenta las características de cada una de las tarjetas y los diversos factores anteriormente planteados se selecciona la tarjeta BeagleBone Blue, la cual posee sensores integrados como: sensor de 9 ejes IMU (Acelerómetro, giroscopio y magnetómetro) barómetro, termómetro, conexión Wi-Fi y Bluetooth para realizar la tele operación del dispositivo, además de tener una buena capacidad de procesamiento, y ser compacto con la mayoría de sensores, también permite hacer calibración de los motores y lo más importante, implementar los controladores de bajo nivel, ya sea el clásico o el adaptativo.

Tabla de Selección de Sensor de proximidad

| Elemento | Características | Rango de detección | Valor |
|---------------------|--|--------------------|-----------|
| Ultrasonido HC-sr04 | <ul style="list-style-type: none"> ● Se puede activar en cualquier momento la detección de los objetos. ● Requiere un cálculo de tiempo para estimar la distancia a la que se encuentra el objeto con el que colisiona. ● Angulo efectivo de medición 15°. ● Consumo energético 0.015W | (2 cm – 500 cm) | \$ 6,000 |
| Ultrasonido US-016 | <ul style="list-style-type: none"> ● Fácil conexión con la tarjeta de control mediante el uso de pines GPIO. ● Angulo efectivo de medición 15°. ● Consumo energético 0.019W. ● Precisión 0.3cm + 1% | (2cm – 300 cm) | \$ 17,000 |
| Cámara USB | <ul style="list-style-type: none"> ● Pierde precisión a mayor altura. ● Amplio ángulo de medición. ● Fácil conexión con la tarjeta de control. ● Requiere un alto coste computacional para el proceso de medición. | (5cm – 200 cm) | \$ 25,000 |
| | <ul style="list-style-type: none"> ● Frecuencia de muestreo 60 Hz. ● Angulo de efectividad. | (20 cm – 150 cm) | \$ 29,000 |

| | | | |
|--------------------------------------|--|-------------------|------------|
| Sensor Sharp Infrarrojo GP2Y0A02YK0F | <ul style="list-style-type: none"> ● Fácil conexión con la tarjeta de control mediante el uso de pines GPIO. ● Consumo energético 0.18W. ● Precisión desconocida. | | |
| Makerfocus Lidar Range Finder Sensor | <ul style="list-style-type: none"> ● Frecuencia de muestreo 100 Hz. ● Fácil conexión con la tarjeta de control mediante la interface de comunicación UART. ● Angulo de efectividad de medición 2,3°. ● Precisión 2%. ● Consumo energético 0.6W. | (30 cm – 1200 cm) | \$ 256,000 |

Tabla 3 Características de los sensores de proximidad encontrados en el mercado. Fuente: Páginas de venta en internet.

Se observa que los sensores, tienen muchas variables a tener en cuenta, y claramente el sensor lidar es el que cuenta con las mejores características, pero su elevado coste impide poder implementarlo en el dron. Por otra parte, analizando el resto de sensores y viendo que su implementación sea fácil con la tarjeta de control seleccionada, se escoge el sensor de ultrasonido US-016 el cual tiene un coste aceptable, con un rango de implementación aceptable, bajo consumo energético y precisión aceptable para el desarrollo del proyecto.

Tabla de Selección de Hélices

| Elemento | Dimensión(cm) | Corriente (A) | Empuje (N) |
|-------------|---------------|---------------|------------|
| Hélice 8045 | 20,32 | 4.10 A | 2.82 N |
| | | 4.85 A | 3.4 N |
| | | 5.65 A | 4.12 N |
| Hélice 1045 | 25,4 | 8.45 A | 4.95 N |

| | | | |
|--|--|---------|--------|
| | | 9.9 A | 5.93 N |
| | | 11.45 A | 6.9 N |

Tabla 4 Características de las hélices encontradas en el mercado. Fuente: rcgroups.com.

Para este tipo de cuadro, se hace un proceso de agrupación que permita relacionar el consumo energético con el empuje y el consumo de corriente. Para esto se hacen 3 pruebas variando la señal enviada desde la tarjeta de control y se mide el empuje con un medidor de peso.

La señal de corriente se toma desde un multímetro teniendo en cuenta que los datos van desde 1A a 10A, Se debe tener en cuenta que las pruebas no deben durar mucho tiempo y se debe estar recargando la batería cuando se cambia de hélice, debido a que, si la batería disminuye su voltaje, se tomaran datos muy diferentes entre las hélices. Por otra parte, otro factor a tener en cuenta son las dimensiones de las hélices, ya que si se escoge una de dimensión grande puede afectar el dispositivo y hacerlo más pesado.

Teniendo en cuenta que el tamaño perjudica el peso del dispositivo, y que a su vez este reduce el tiempo de operación, se propone usar la hélice 8045, ya que puede alcanzar un empuje entre sus cuatro motores de 21.3N y permite que el diseño del dispositivo sea el más pequeño de las hélices propuestas, además disminuye la corriente pico, por lo que ayuda a proteger el controlador de velocidad electrónico (ESC).

Tabla de Selección de ESC

| Elemento | Características | VALOR |
|----------------------|--|-----------|
| ESC 30A RE123 | <ul style="list-style-type: none"> • Corriente pico: 35 A/12 s. • Voltaje soportado: 2S – 3S. • Se encuentra en Colombia. | \$ 30.000 |
| ESC Dys 20a Blheli-s | <ul style="list-style-type: none"> • Corriente pico: no especifica • Voltaje soportado: 3S – 4S. | \$52.500 |

Tabla 5 Características de los Motores brushless encontrados en el mercado. Fuente: Paginas de venta en internet.

Este suele corresponder al proceso de selección de componente más sencillo, debido a que como tal se rige de acuerdo al consumo de corriente de los motores, y al suministro de energía de la batería. Dicho esto, es necesario que el controlador de velocidad electrónico, pueda soportar una corriente de 6A, con un voltaje de 12.7V o 3S.

Tabla de Selección de Batería

| Elemento | Características eléctricas | Valor | Tiempo de operación estimado | Dimensión |
|-----------------------|--|-----------|------------------------------|--------------------|
| Turnigy 1500mah 3S | <ul style="list-style-type: none"> • 1500 mah. • 3S. • 30C. | \$ 69.000 | 5 min | 73 x 33 x 27 mm |
| Zippy 2200 mah 3S | <ul style="list-style-type: none"> • 1500 mah. • 3S. • 30C. | \$75.000 | 7 min | 107x21x34m m |

Tabla 6 Características de las baterías LiPo encontrados en el mercado. Fuente: Páginas de venta en internet.

Este último, pero no menos importante, también se debe tener en cuenta ya que interfiere hasta en el diseño del dispositivo. Es necesario tener cuidado la masa que este proporciona, ya que representa un gran porcentaje de la masa total del sistema, sin desestimar que la capacidad de la batería es la que permite tener más tiempo de vuelo.

3.1.2 DISEÑO DE BASE DE PRUEBAS 1 GDL

Las pruebas de 1gdl, son una operación necesaria en el proceso de caracterización de los motores, para conocer las revoluciones y el empuje ejercido por cada motor de acuerdo a una señal ocasionada de control dada por el operador.

Para realizar este proceso, se diseña una estructura, la cual se buscó que sea económica,



Ilustración 12 Producto final de plataforma de pruebas de 1gdl. Fuente: Autor.

sencilla y completamente funcional para la toma de mediciones de los motores. Teniendo en

cuenta estas condiciones, se realiza el diseño por capas de la estructura fija para luego fabricarse mediante corte a láser, y el diseño de la pieza móvil se fabrica mediante técnicas de impresión 3d.

Primero se diseñan las piezas que van a sujetar la estructura, con los respectivos ejes de unión. Debe recordarse que la fabricación de estas piezas se va a realizar mediante el proceso de corte a laser.

De esta forma se desarrolla un total de cinco piezas.

Pieza uno, como se muestra en la Ilustración

13 Ilustración 13 Pieza uno de plataforma 1gdl.

Fuente: Autor. se desarrollan cinco ejes de unión y un orificio de menor área en el cual se va a sujetar el elemento para medir el peso. Este elemento se fabricará dos veces.

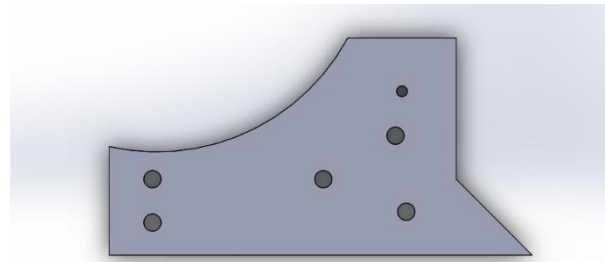


Ilustración 13 Pieza uno de plataforma 1gdl. Fuente: Autor.

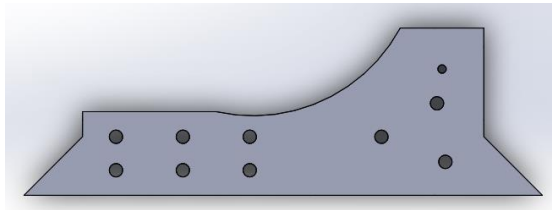


Ilustración 14 Pieza dos de plataforma 1gdl. Fuente: Autor.

Pieza dos, como se muestra en la Ilustración

14 Ilustración 14 Pieza dos de plataforma 1gdl.

Fuente: Autor., esta pieza es muy parecida a la pieza 1, con la leve diferencia que es más alargada, para desarrollar el espacio donde se va a acoplar la pieza móvil

va a acoplar la pieza móvil

Pieza tres, como se muestra en la Ilustración

15, es también muy parecida a la pieza uno,

pero en esta no se deja el espacio donde va

el tornillo de soporte del medidor de peso,

esto con el fin de dejar un espacio donde va

el soporte de este medidor.

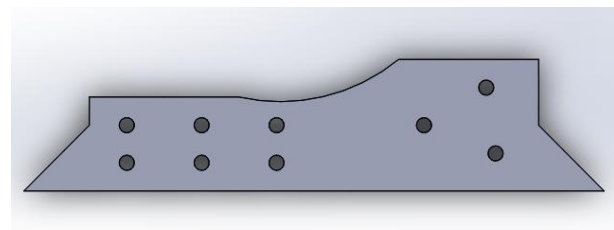


Ilustración 15 Pieza tres de plataforma 1gdl. Fuente: Autor.



Ilustración 16 Pieza cuatro de plataforma 1gdl. Fuente: Autor.

Pieza cuatro, como se muestra en la Ilustración 17, esta pieza es

alargada y servirá de sujeción entre la pieza 6 y las piezas 1, 2 y

3, con ella se busca realizar un soporte y la fuerza ejercida por el motor no ocasione perturbaciones en el ensamble de las piezas.

Pieza cinco, esta pieza corresponde a la pieza móvil, en la cual va a estar sujetado el motor *brushless*, debe contar con un eje de giro en donde se acoplan los rodamientos giratorios, y también los orificios donde se ajusta el motor.

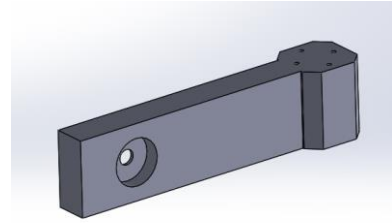
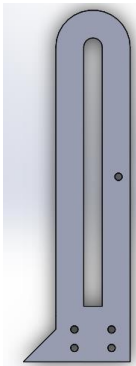


Ilustración 17 Pieza cinco de plataforma 1gd. Fuente: Autor.



Pieza seis, esta pieza es principal ya que es donde se va a ubicar la pieza cinco y va a ser la unión entre esta donde se sujeta el motor y las demás piezas. Esta cuenta con tres orificios donde se va a sujetar, y esta representa el complemento izquierdo que le hace falta a la pieza uno. Además del orificio que encuentra en la mitad es para unir la pieza cuatro, ya que es conveniente desarrollar un apoyo para cuando se realicen las pruebas del motor.

Ilustración 18 Pieza seis de plataforma 1 gd. Fuente: Autor.

3.1.2.1. Ensamble

Para realizar el proceso de ensamble es muy sencillo, primero se toma la pieza cuatro y se introduce los dos tornillos que van a sujetar las siguientes piezas, seguidamente se coloca la pieza uno y la pieza seis una al lado de la otra, posteriormente se coloca la pieza dos ajustándose el tornillo que viene desde la pieza cuatro, por último, se ensambla la pieza tres. Luego, el proceso es espejo, se conecta la pieza dos, seguido de la pieza uno y seis, así sucesivamente hasta conectar de nuevo la pieza cuatro y terminar de enroscar el tornillo.

Luego se procede a conectar el resto de tornillos de sujeción. Por otra parte, para ensamblar la pieza cinco, es necesario primero tener la pieza, luego conectar a ella los dos rodamientos circulares en su eje de rotación. Y luego se atraviesa un tornillo, que va atravesando el ensamble anterior, por el orificio de la pieza 6.

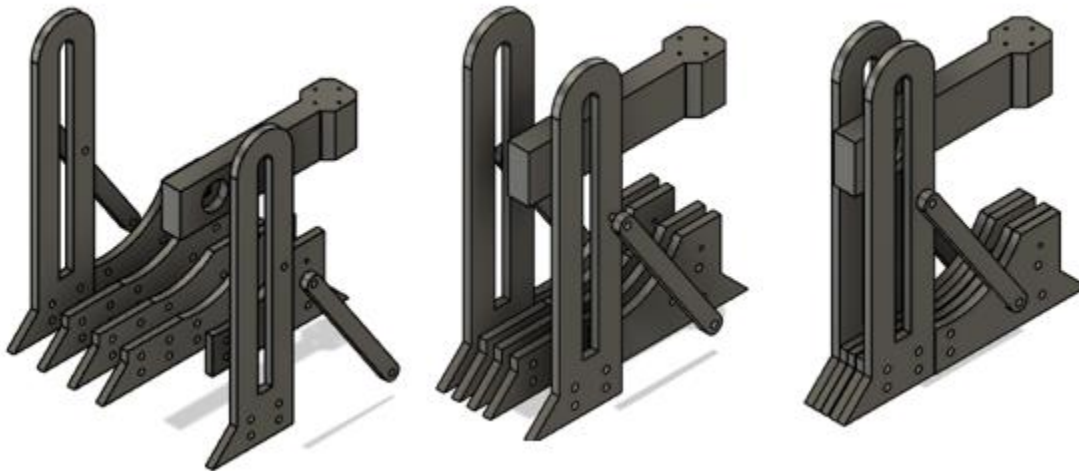
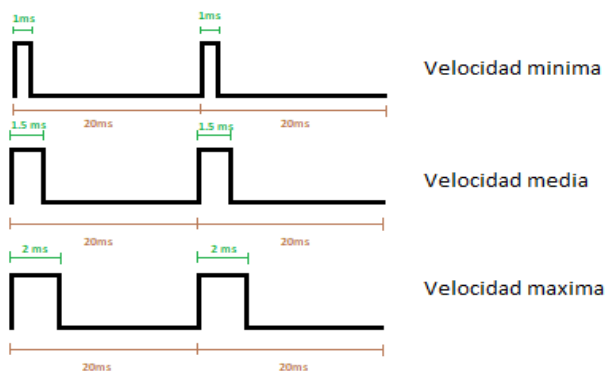


Ilustración 19 Proceso de ensamblaje de base de pruebas de 1gdl. Fuente: Autor.

3.1.2.2. Proceso de pruebas

En el desarrollo de pruebas, se conecta de la siguiente manera: Primero se conecta la batería a la tarjeta BeagleBone Blue, y a su vez al ESC mediante el uso de los cables (cables compartidores de energía), luego del ESC se conecta a la batería teniendo en cuenta las recomendaciones para cambio de giro por si es necesario, es necesario fijar la hélice para que no ocurra ninguna catástrofe. Por otra parte, del ESC también se dispone el cable de control en microsegundos que es conectado a la Beagle, Por lo tanto, nuestra base experimental queda como se muestra en la Ilustración 20.



Para iniciar el proceso de pruebas del sistema, se tiene en cuenta cómo funciona el control en microsegundos y el rango de operación que este tiene. El control en microsegundos es un control que se ejerce sobre los ESC, y la mayoría de servomotores. Este es un control que se ejerce cada 20ms, por lo que dependiendo

del pulso que se envié en esos 20ms, depende las revoluciones del motor, en el caso de este dispositivo y la mayoría de dispositivos que se manejan un controlador ESC,

Pero este dato es teórico, porque los ESC tienen diferentes modelos de operación, teniendo en cuenta que son realizados por diferentes empresas y manejan diferentes chips de operación. Para solucionar este problema, se tiene que consultar el intervalo de operación de cada uno de los motores, con su respectivo ESC experimentalmente.

Ilustración 21 Señal de control enviada al ESC. Fuente: Autor

3.1.2.3. Resultados

También se debe tener en cuenta que si el empuje medido es muy grande no será necesario los siguientes datos, debido a que para levantar el dispositivo no es necesario la mayor potencia de los motores, y de hecho es necesario limitarlo para que no se exceda el sobre pico de corriente, por lo que se establece un bloque en Simulink para ejercer un filtro pasa media en el pulso de control enviado a los motores como se observa en la Ilustración 22. En el desarrollo de las pruebas también se mide el voltaje proporcionado al motor en el momento de la prueba, sabiendo que este parámetro también afecta en la velocidad del motor, independientemente del pulso.

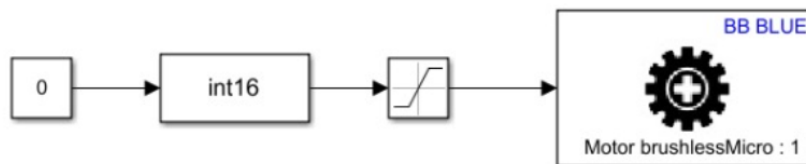


Ilustración 22 Diagrama de bloques para las pruebas de 1GDL.

Motor 1

| Pulso señal en microsegundos | Revolución | Voltaje medido | Empuje medido |
|------------------------------|------------|----------------|---------------|
| 1400 | 4600 | 11.45 V | 1.27 N |
| 1500 | 5850 | 11.45 V | 2.31 N |
| 1600 | 6824 | 11.43 V | 3.87 N |
| 1700 | 8080 | 11.4 V | 5.79 N |
| 1800 | 8870 | 11.4 V | 7.26 N |

Tabla 7 Caracterización del motor 1. Fuente: Autor.

Motor 2

| Pulso señal en microsegundos | Revolución | Voltaje medido | Empuje medido |
|------------------------------|------------|----------------|---------------|
| 1400 | 5470 | 11.36 V | - |
| 1500 | 6100 | 11.36 V | 1.27 N |
| 1600 | 6700 | 11.32 V | 2.35 N |
| 1700 | 7490 | 11.3 V | 3.83 N |
| 1800 | 8300 | 11.28 V | 5.35 N |

Tabla 8 Caracterización del motor 2. Fuente: Autor.

Motor 3

| Pulso señal en microsegundos | Revolución | Voltaje medido | Empuje medido |
|------------------------------|------------|----------------|---------------|
| 1400 | 5440 | 10.83 | 1.86 N |
| 1500 | 6200 | 10.81 | 2.45 N |
| 1600 | 6700 | 10.77 | 3.38 N |
| 1700 | 7400 | 10.75 V | 4.17 N |
| 1800 | 8080 | 10.7V | 5.35 N |

Tabla 9 Caracterización del motor 3. Fuente: Autor.

Motor 4

| Pulso señal en microsegundos | Revolución | Voltaje medido | Empuje medido |
|------------------------------|------------|----------------|---------------|
| 1400 | 5540 | 11.34 | 1.62 N |
| 1500 | 6260 | 11.32 | 2.5 N |
| 1600 | 6980 | 11.3 | 3.87 N |
| 1700 | 7700 | 11.24 V | 4.95 N |

Tabla 10 Caracterización del motor 4. Fuente: Autor.

| | | | |
|------|------|---------|--------|
| 1800 | 8420 | 11.19 V | 6.08 N |
|------|------|---------|--------|

3.1.3. DISEÑO MECÁNICO DEL DRONE

Inicialmente antes de iniciar cualquier proceso de diseño mecánico, se recomienda tener la siguiente información clara.

- Ubicación de los dispositivos.
- Tamaño de las hélices.
- Fuerza de empuje de los motores.
- Consumo energético de los motores

Todo está asociado de alguna manera, es importante saber la ubicación de los dispositivos porque es necesario encontrar el lugar donde se puede acomodar la batería y el controlador. Los tamaños de las hélices pueden generar una mayor fuerza de empuje de los motores, y así poder levantar un peso del dispositivo considerable, pero esto trae consecuencias, como un mayor consumo energético, lo que ocasiona, tener que replantear la capacidad de la batería que se dispondrá en el dispositivo. El consumo energético de los motores también es importante, limita hasta la forma de conectar los motores, se debe tener en cuenta que el cableado de energía debe soportar al menos unos 110W y como mencione anteriormente, esto está intrínsecamente arraigado al tamaño de las hélices que se vayan a usar.

BRAZO

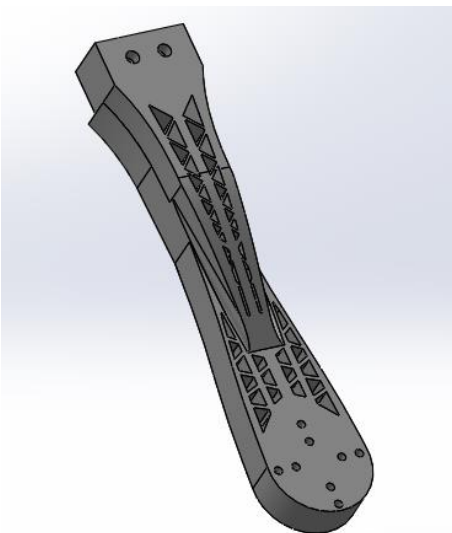


Ilustración 23 Diseño del brazo del drone.
Fuente: Autor.

El brazo de un drone, corresponde a la parte mecánica en la que va a estar apoyado el motor *brushless*. Y muy seguramente corresponda a la pieza más difícil de diseñar, ya que mecánicamente tiene la mayor responsabilidad, y es de llevar la fuerza de los motores hasta el punto de apoyo con la estructura central.

Tiene que ser estructuralmente fuerte, soportar la fuerza ejercida a cierta distancia de su punto de apoyo, y claramente tiene que ser liviano, para que el dispositivo final sea lo menos pesado posible.

Para esto se realiza el siguiente diseño mostrado en la Ilustración 23, donde se observa tiene unos orificios con el fin de hacerlo menos pesado, también se realizan los orificios para el acoplamiento con el motor.

Hay que tener en cuenta un factor muy importante y es el uso de los parámetros físicos del elemento en el que se va a fabricar nuestras piezas. El más importante es el límite de elasticidad, y permite analizar el límite de fuerza que puede soportar la estructura.

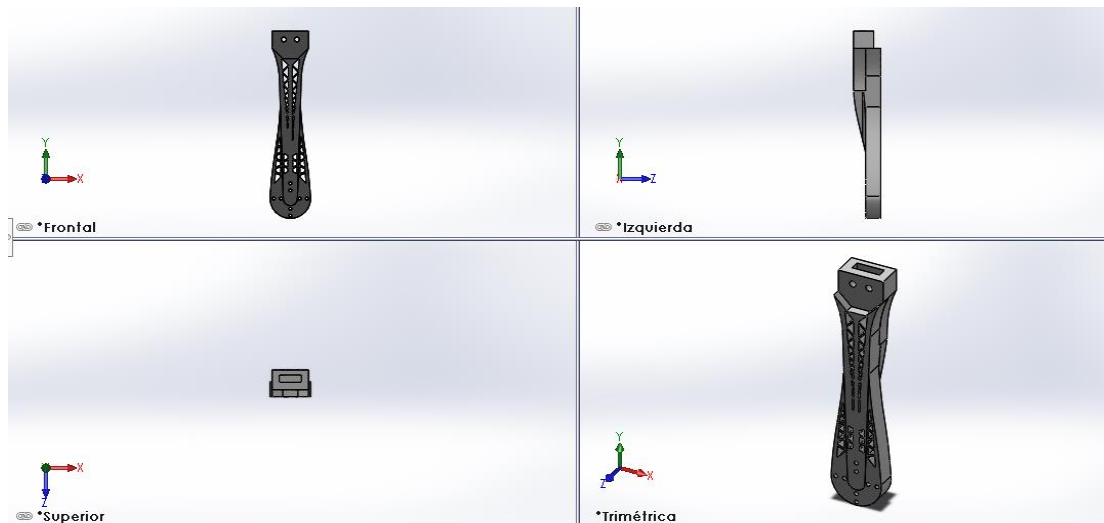


Ilustración 24 Vista frontal, izquierda y superior del brazo del drone. Fuente: Autor

Por eso mismo, como se ve en la Ilustración 24 (Izquierda, inferior) se observa cómo se deja un hueco en la parte inferior izquierda. Pero este, no se crea como un desnivel, si no que se diseña como una forma de rampa para que la pieza al estar operando no sufra inconvenientes que pueden ocasionar una catástrofe tanto en el dispositivo como a su alrededor.

SOPORTE SUPERIOR

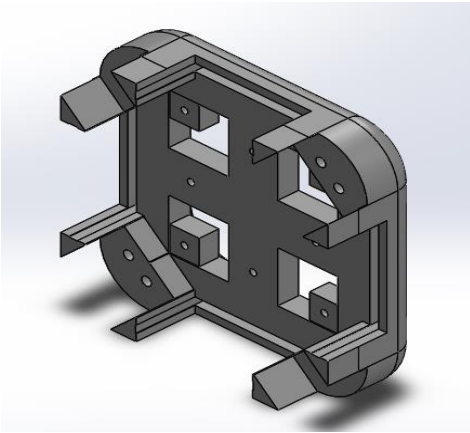


Ilustración 25 Soporte Superior del drone.
Fuente: Autor.

Siguiendo las ideas, y diseñando de forma esquelética las piezas, se procede a diseñar el soporte superior de la estructura, se tiene en cuenta que en esta pieza también hay limitantes de un espacio mínimo requerido. Por ejemplo, se debe tener en cuenta que la batería pasa por en medio de la pieza y que el controlador va a estar ubicado en la parte superior. Además, se diseñan unos orificios semi cuadrangulares para hacer la pieza menos pesada y tener una fácil conexión entre los controladores de velocidad electrónicos y la tarjeta de control.

Viendo las diferentes vistas se observa que gran parte de la pieza es hueca, y esto es una gran ventaja al momento de reducir el peso del dispositivo, pero también se debe tener en cuenta que ninguna de sus elementos quede frágil. Para esto, se desarrolla en cada pieza un análisis de elementos finitos en el mismo software que se diseña SOLIDWORKS.

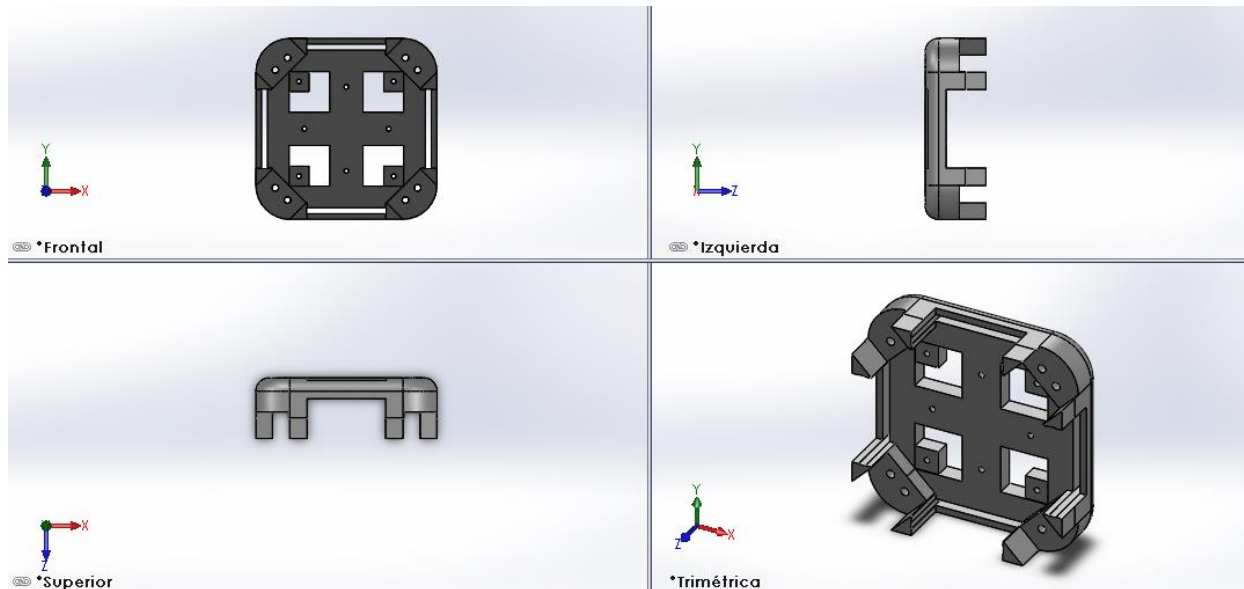


Ilustración 26 Vista frontal, izquierda y superior del soporte superior del drone. Fuente: Autor

SOPORTE INFERIOR

El diseño del soporte de inferior es sencillo, lo único a tener en cuenta son los orificios por dónde van los tornillos que ensamblan las piezas, además a esto se diseña un espacio donde se pueda ubicar un sensor de ultrasonido, con el fin de medir la posición de altitud del dispositivo.

Al igual que en las piezas anteriores, se busca reducir el peso sin ceder en propiedades físicas, también se debe tener en cuenta que este debe ir acoplado a la pieza de “soporte superior” mencionada anteriormente, y debe respetarse los espacios donde va ubicado la batería LiPo.

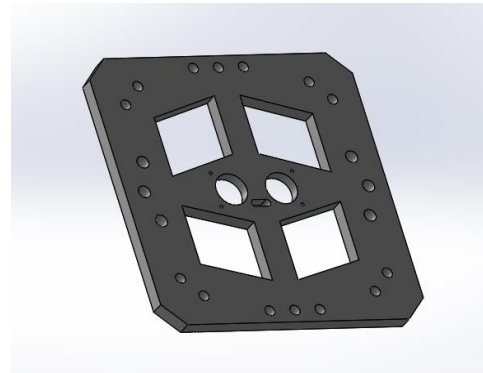


Ilustración 27 Soporte inferior del drone. Fuente: Autor

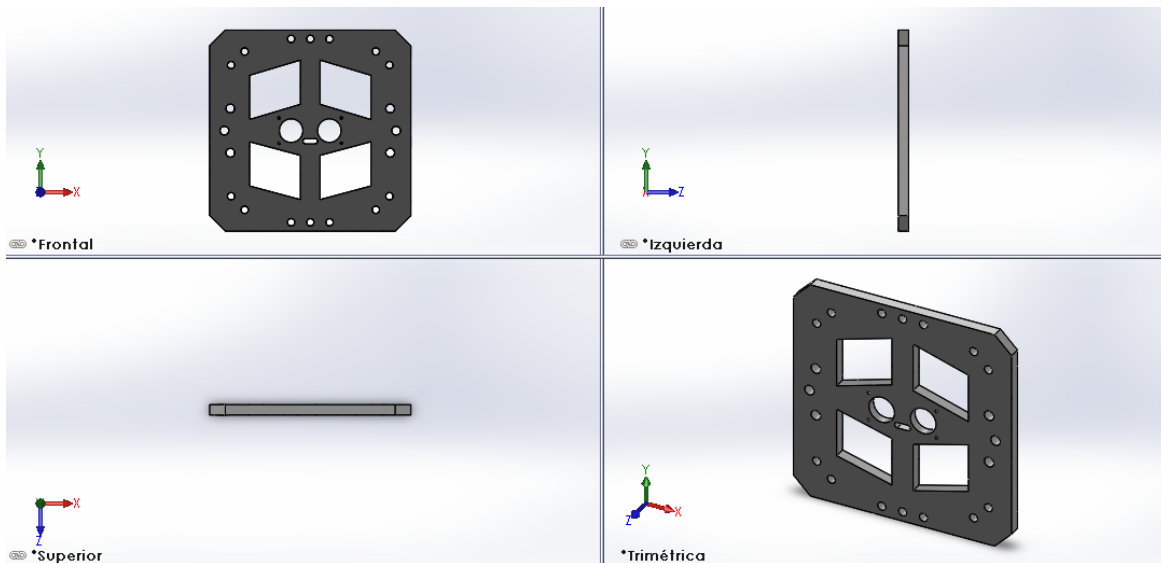


Ilustración 28 Vista frontal, izquierda y superior del soporte superior del drone. Fuente: Autor

3.1.3.1 ENSAMBLE Y ANÁLISIS DE ELEMENTOS FINITOS

Para el proceso de ensamblaje es necesario ubicar las diferentes relaciones mecánicas, entre ellas la más importante son las relaciones ejercidas sobre la base inferior, la cual va a quedar

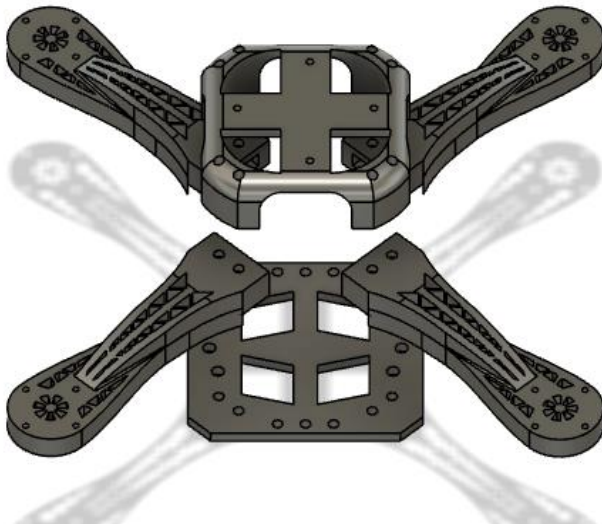


Ilustración 29 Ensamble de las piezas del dron. Fuente: Autor.

fija durante el análisis de elementos finitos.

Se realizó una relación de posiciones concéntricas ajustándola como si hubiera tornillos en la estructura.

Para el análisis de elementos finitos, es necesario conocer las propiedades mecánicas de la materia prima en la cual se va a fabricar la pieza, dentro de los requisitos para

realizar este análisis se debe tener en cuenta: el límite elástico, el límite de tracción, límite de compresión, módulo elástico, coeficiente de Poisson. Al usar un material estandarizado como lo es el PLA (Ácido Poliláctico) se encuentra fácilmente su información mecánica, aunque se debe tener en cuenta que muchas veces se presentan variaciones, debido a que en el proceso de fabricaciones algunas veces se mezcla con impurezas [37].

Luego para el desarrollo de análisis elementos finitos, es necesario especificar el tipo de conexiones que hay entre las piezas del dispositivo.

Posteriormente, se debe especificar las fuerzas ejercidas por los motores en los puntos específicos. Es necesario conocer el valor máximo que van a impulsar los motores, y no ajustar el valor del peso del dispositivo y dividirlo en cuatro, cabe recordar que para poderse levantar debe vencer la fuerza gravitacional y por eso mismo la fuerza debe ser mayor al peso.

Cabe recalcar que las mediciones más importantes son la de tensión y estrés, la cual se puede interpretar de la siguiente forma: La tensión es la fuerza normal ejercida por las partes mecánicas de un drone. Mediante esta fuerza, el dispositivo conjuntamente se levanta y desplaza a

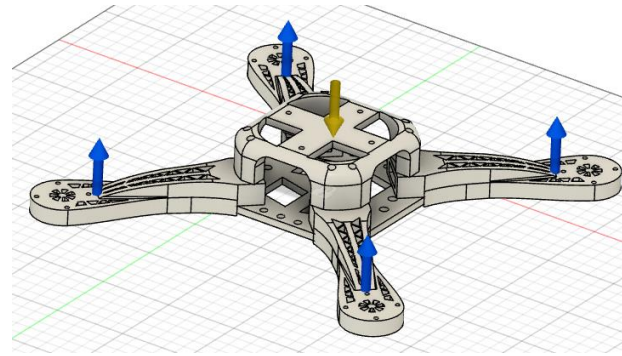


Ilustración 30 Recreación de las fuerzas ejercidas sobre el drone. Fuente: Autor.

través del aire, por ello es necesario conocer si la estructura puede soportar la tensión en el momento que el dispositivo se levanta, ya que cuando esto ocurre, es donde ejerce una mayor tensión su estructura. Por otra parte, el estrés previene de posibles daños después de tener un determinado tiempo de uso,

Análisis de Tensión

Se observa que se alcanza un máximo de tensión de 0.0001662N en los brazos del drone, pero en áreas pequeñas..

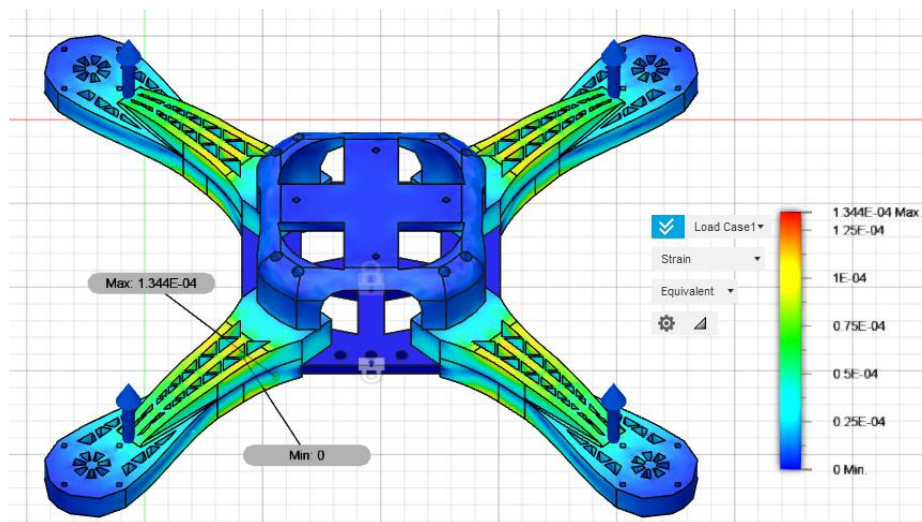


Ilustración 31 Vista general del análisis de tensión en el drone. Fuente: Autor.

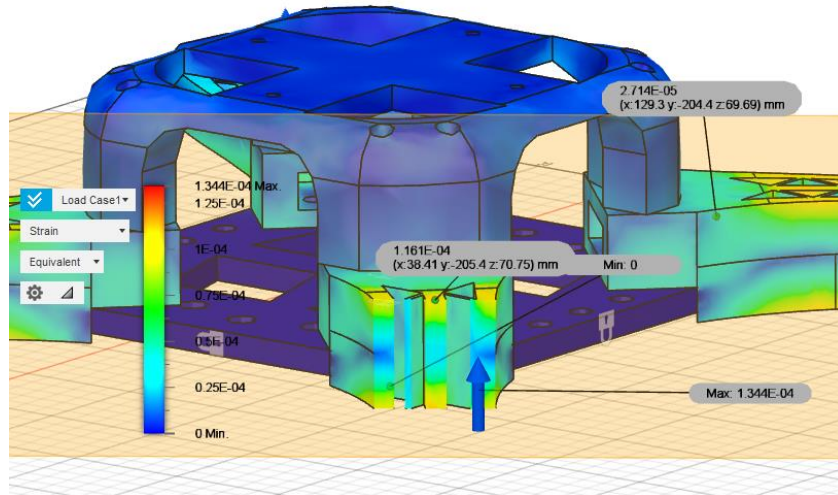


Ilustración 32 Vista de corte del drone en puntos críticos del análisis de tensión. Fuente: Autor.

Como se observa en la Ilustración 32, este valor además de tener poco valor cubre solo pequeños lugares, inclusive se puede observar que, en el intermedio de la pieza, tiende a tomar valores de alrededor 0.0001344N, y en pequeños puntos un valor máximo de 0.0006176N.

Análisis de Estrés

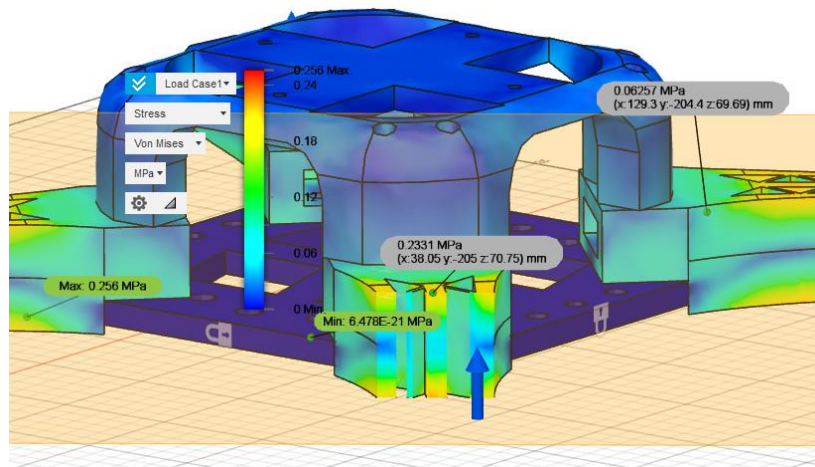


Ilustración 33 Vista de corte del drone en puntos críticos del análisis de estrés. Fuente: Autor.

Para el caso de análisis de estrés, se observa que la parte que sufre más de estrés es el brazo, y una parte del soporte superior más precisamente cerca al eje de unión con el acople, teniendo un valor de 0.256MPa, como se observa en la Ilustración 33. Pero al igual que la tensión, en el punto intermedio de la pieza toma valores mucho más bajos, que oscilan entre 0.05Mpa y 0.228 MPa.

Factor de seguridad

El factor de seguridad es una información de análisis de elementos finitos muy importante y es que este permite proveer cuanto puede soportar el diseño teniendo en cuenta las fuerzas que se están aplicando. En el caso del dispositivo se puede observar que este arroja un valor de 15, lo cual es un dato elevado e indica que el diseño es aceptable.

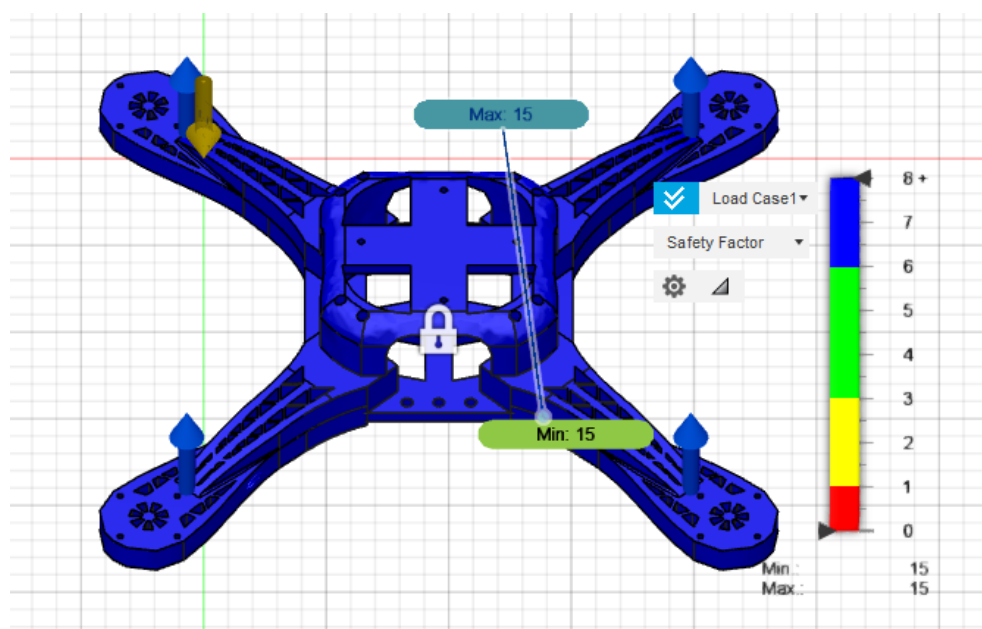


Ilustración 34 Factor de seguridad del dron. Fuente: Autor.

Desplazamiento máximo

En cuanto a desplazamiento, nuestras piezas no sufren un desplazamiento considerable, ya que su valor máximo es de 0.03799mm. Una medida imperceptible al ojo humano.

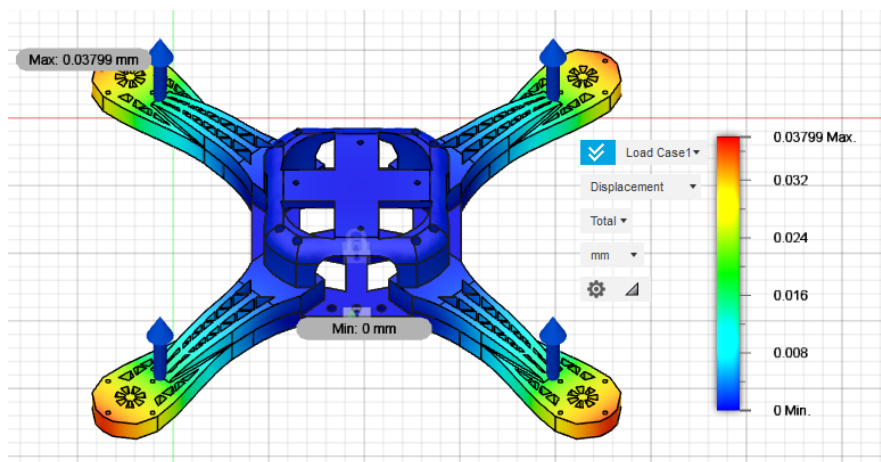


Ilustración 35 Vista de desplazamiento máximo del dron. Fuente: Autor.

3.1.4 DISEÑO MECÁNICO DE LA PLATAFORMA DE PRUEBAS 2 GDL

La plataforma de pruebas de 2 GDL es una estructura mecánica basada en la idea Quanser 3 DOF, esta plataforma busca proporcionar dos grados de libertad correspondientes a los ángulos pitch y roll observados en la Ilustración 38. Para ello, fue necesario entender como un dispositivo quanser 3 gdl, es capaz de generar los 3 grados de libertad sin perturbar el movimiento de las hélices.

Para observar a detalle se observa la Ilustración 36, que muestra el quanser hover 3DOF. Primero se tiene en cuenta la estructura circular en donde se sujetan los 4 motores, que son los que están protegidos por las mallas azules. Esta estructura circular está unida a un eje rotacional, la cual está anclada a un soporte rectangular. A su vez el soporte rectangular cuenta con un eje perpendicular al eje que soporta la estructura circular, y mediante estos dos ejes, el dispositivo puede moverse en los ángulos roll y pitch, correspondientes a los giros del eje “x” y “y” como se observa en la Ilustración 37.



Ilustración 36 Quanser Hover 3 DOF.
Fuente: Quanser.

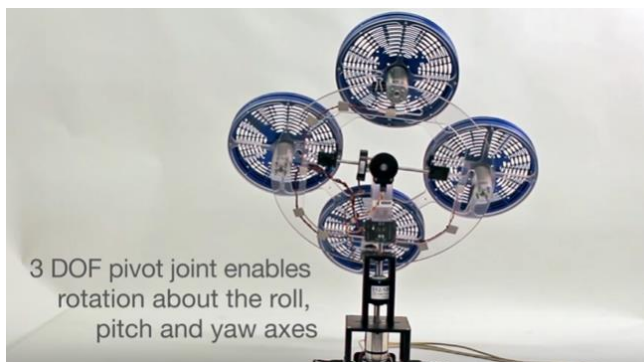


Ilustración 37 Sistema de rotación del Quanser hover 3 DOF.
Fuente: Quanser.

Es importante reconocer que este dispositivo está totalmente anclado a una superficie fija, y por esto mismo sus tres grados de libertad, son de rotación y no de traslación. También se debe tener en cuenta que el ángulo de giro “yaw”, no resulta ser tan importante en el control de un drone, a menos de que su aplicación lo requiera.

Por lo tanto, lo que se va a diseñar será una plataforma de 2 grados de libertad, la cual podrá permitir que el dispositivo en cuestión pueda girar con respecto al eje “x” y “y”. También se

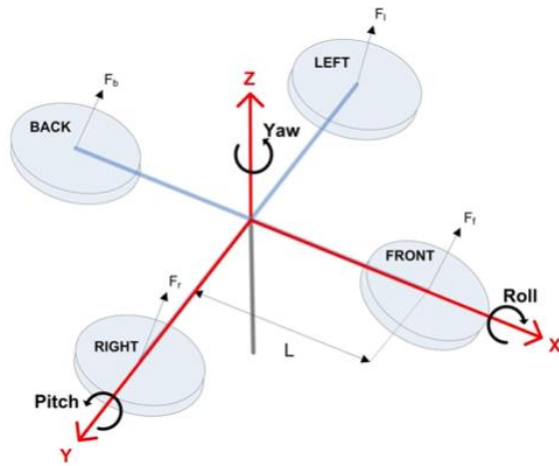


Ilustración 38 Sistema de referencia Quanser 3 dof. Fuente: Quanser.

plantea una unión al igual que el soporte rectangular, como el que cuenta el dispositivo quanser y en vez de diseñar una estructura circular se utiliza la misma plataforma del dispositivo conectado a través de unas varillas lisas a las cuales se realizó una rosca interna.

Lo más importante para el diseño es tener la idea básica, de cómo van a ir conectados los ejes de rotación, luego, el resto es ajustar este sistema, y que todo funcione

sin ninguna perturbación.

Como se observa en la Ilustración 39, se tiene los dos ejes uno vertical y otro horizontal, ambos unidos por dos piezas con un simple sistema de encaje en cruz, para una mejor transmisión de fuerza entre las piezas de giro y el soporte rectangular.

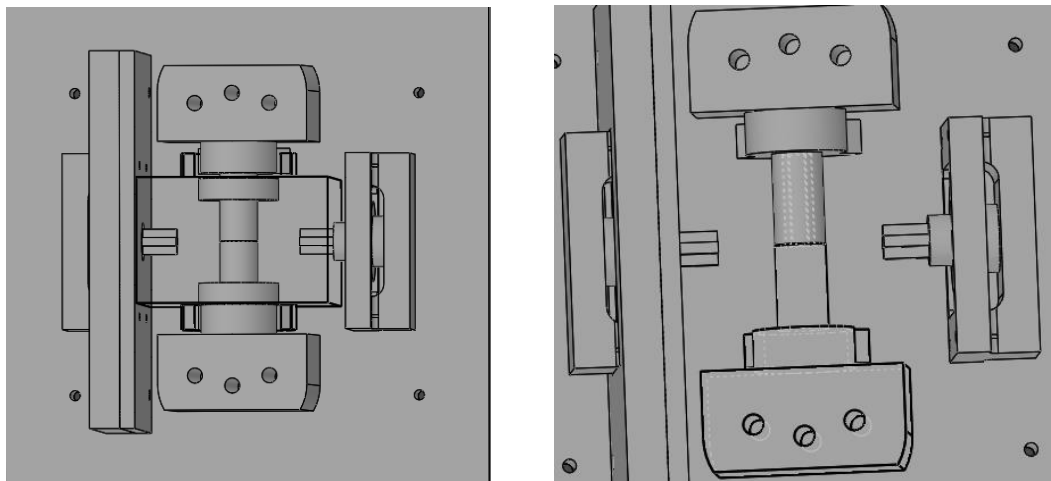


Ilustración 39 Soporte de rectangular de la base de pruebas de 2gdl. Fuente: Autor.

El eje horizontal por su parte, no atraviesa completamente la pieza rectangular de soporte, debido a que colisionaría con el otro eje. Lo que se busca es dejarlo hasta cierto punto y conectarlo de ambos lados, así se reducen los problemas de rozamiento.

En el eje vertical, las piezas que representan este eje, cada una cuenta con tres orificios en sus extremos, esto se diseña con el fin de acoplar el drone en la parte superior como se observa en la Ilustración 40.

Ahora, la parte inferior debe ir perfectamente ensamblada, de forma que la transmisión de fuerza y de movimiento a través de las piezas no tenga inconvenientes de colisión o rompimiento.

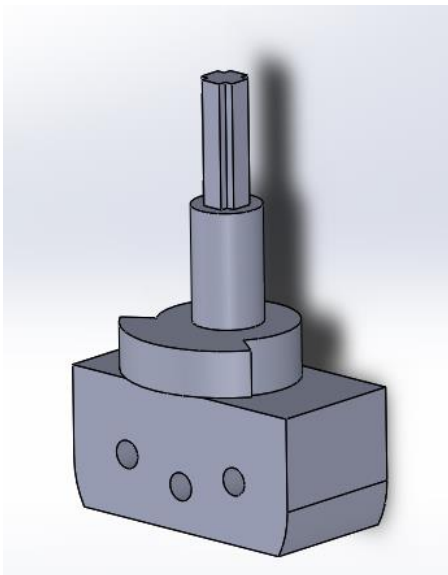


Ilustración 42 Soporte de sujeción. Fuente: Autor.

resistir de mejor manera la fuerza ejercida por el drone en la parte superior. También debe tenerse en cuenta que este debe ajustar con el diámetro interno del rodamiento circular que va a ir colocado en el soporte rectangular, como se ve en la Ilustración 41, donde el rodamiento está representado por la pieza seleccionada de color azul.

Se sigue analizando y se encuentra una superficie sobresaliente que crea una figura en

forma de anillo, esta corresponde a la limitante de giro por parte de la pieza, que choca con

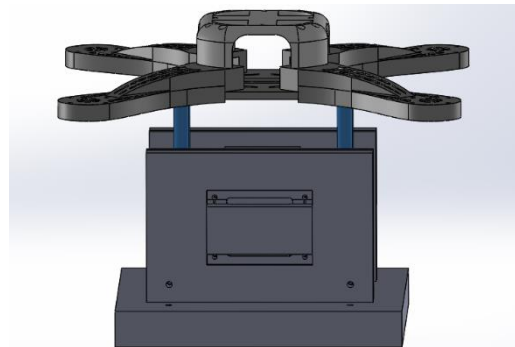


Ilustración 40 Piezas de unión con el eje del drone con la plataforma de 2gdl. Fuente: Autor.

El diseño de una base quanser puede ser sencillo, pero hay que tener en cuenta muchos factores, como límites de rotación, resistencia mecánica del material, encaje y sujeción de las piezas, entre otros. Para ello se pasa al soporte de sujeción del primer eje de rotación, el cual se observa en la Ilustración 42. Este diseño corresponde a la pieza de soporte tipo macho, donde se puede ver que sobresale la unión en tipo cruz para unirse con el soporte tipo hembra, posteriormente se encuentra su parte cilíndrica que se diseñó con el fin de que pudiera

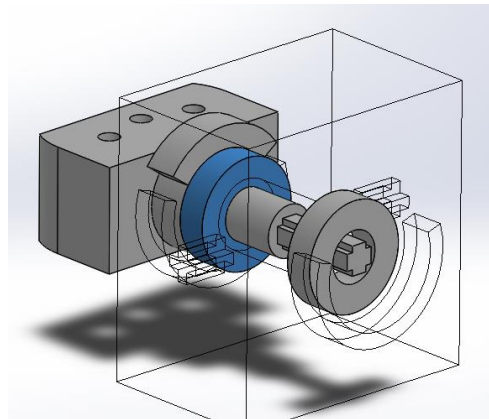


Ilustración 41 Sujeción del rodamiento circular. Fuente: Autor.

su pieza complementaria ubicada en el soporte rectangular. Debido a que es una base de pruebas, no es necesario que su ángulo de giro sea tan amplio, así que se limita a un giro de 35°, teniendo en cuenta que puede tomar ángulos negativos y positivos, y de esta forma el rango será de -17.5° a 17.5°.

Para el segundo eje de rotación, se tiene un eje de sujeción más pequeño, teniendo en cuenta que este no puede atravesar todo el soporte rectangular porque colisionaría con el primer eje de rotación como se muestra en la Ilustración 43. Pero también se debe tener en cuenta que este eje no tiene una limitante que le prohíba salir del soporte rectangular.

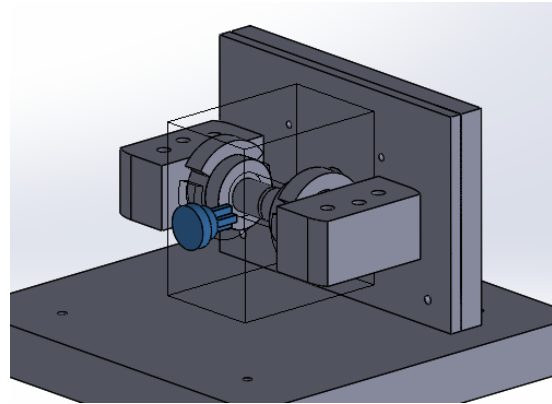


Ilustración 43 Eje de rotación del segundo grado de libertad. Fuente: Autor.

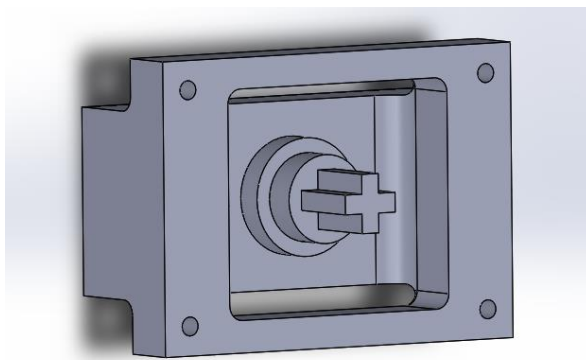


Ilustración 44 Sujeción del segundo eje de rotación. Fuente: Autor.

Por eso mismo se diseña la siguiente pieza, la cual funciona como un encaje, donde la parte circular que se ve en la Ilustración 44, queda bloqueada por una estructura circular lisa, con el fin de disminuir el rozamiento entre piezas, cuando este realice el giro.

3.1.5. ANÁLISIS DE ELEMENTOS FINITOS

En este análisis de elementos finitos, se realiza el análisis de la fuerza en dos direcciones, como si el dispositivo estuviera en reposo, donde solamente actúa la fuerza gravitacional y cuando se están accionando los motores, de forma que el dispositivo estuviera impidiendo que el dron se levantara.

En el primer caso, se debe tener en cuenta que la única fuerza ejercida sobre el soporte de sujeción del primer eje es el peso del dron, para ello se hace el cálculo del peso con la segunda ley de Newton.

$$F = ma \quad (21)$$

Dónde m : corresponde al peso del dispositivo

a : Constante gravitacional.

F : Fuerza

Se miden los valores de la masa de las piezas y se obtiene que su valor es:

| Elemento | Cantidad | Masa Unitaria (gr) | Masa (gr) |
|---------------------|----------|--------------------|-----------|
| Brazo | 4 | 45 | 180 |
| Tapa | 1 | 80 | 80 |
| Base | 1 | 70 | 70 |
| Motor+ESC+Hélices | 4 | 80 | 320 |
| Controlador+Soporte | 1 | 120 | 120 |
| Batería | 1 | 190 | 190 |
| TOTAL | - | - | 940 |

Tabla 11 Peso de los elementos del sistema. Fuente: Autor.

Teniendo estas mediciones se calcula el peso del dron.

$$F = ma$$

Reemplazando...

$$m = 1.04 \text{ Kg}, a = 9.8 \text{ m/s}$$

$$F = (1.04 \text{ Kg}) * (9.8 \frac{\text{m}}{\text{s}})$$

$$F = 10.192 \text{ N}$$

Se divide este valor en dos, ya que la fuerza se distribuye en los dos lados del eje de rotación como se muestra en la Ilustración 45, entonces la aplicación de fuerza para en el primer caso sería de 5.096N en cada uno de los lados. Cada lado está representado por dos flechas azules,

que representan las dos varillas lisas por donde se transfiere la fuerza, y la flecha amarilla corresponde a la fuerza gravitacional.

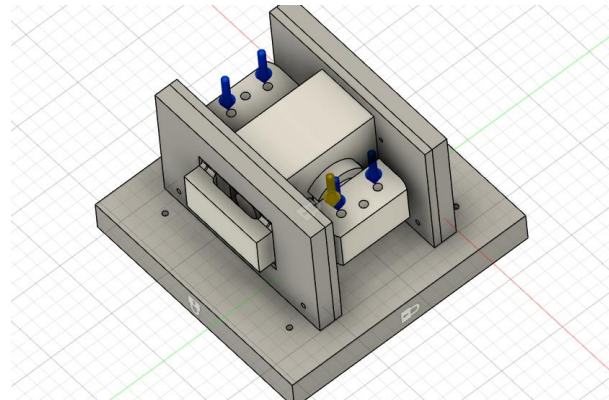


Ilustración 45 Análisis de elementos finitos en la base de 2 gdl. Fuente: Autor.

Al igual que en el caso de análisis de elementos finitos del drone, se especifica los materiales y las propiedades físicas requeridas por el software para el análisis de elementos finitos, se debe tener en

cuenta que no toda la plataforma de pruebas de 2GDL está diseñada en PLA, debido a que conllevaría un costo elevado en su fabricación, así que las piezas laterales donde se apoya el segundo eje y también la pieza donde se apoya toda la estructura están fabricadas en MDF.

Análisis de estrés

En el análisis de estrés, se puede observar que el mayor estrés es generado en los vértices donde está en contacto con las varillas lisas sin llegar a ser relevante, ya que los desarrollos de las pruebas no representan un proceso que conlleve bastante tiempo. También por otra parte se realiza una visión de corte atravesando las piezas, con el fin de observar el estrés en el centro de ella; con esto se observa que el promedio de estrés esta alrededor de 0.03386MPa, como se ve en la Ilustración 46.

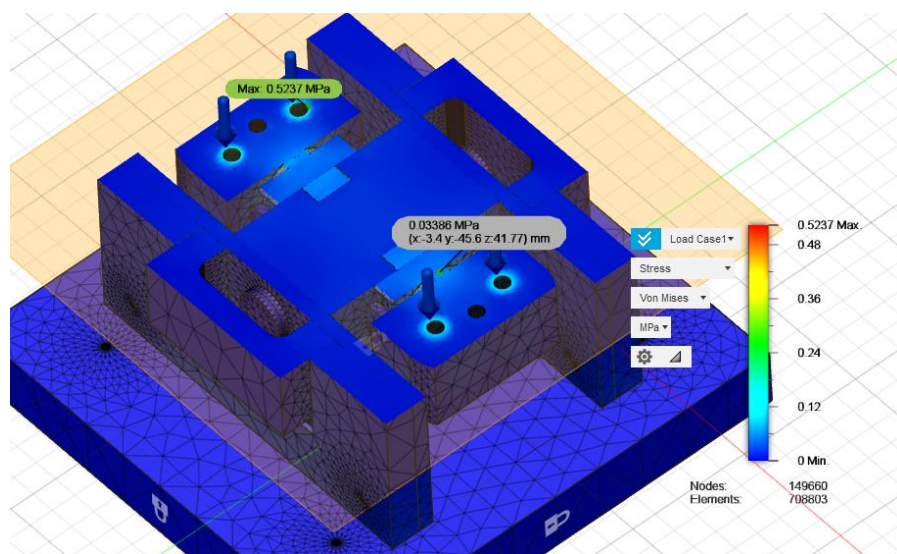


Ilustración 46 Vista de corte del análisis de estrés para el primer eje de rotación. Fuente: Autor.

Por otra parte, para el segundo eje de rotación, no se sufre estrés, esto es debido a que la mayoría de es disipada a través de la pieza móvil que sujeta el primer eje. Su valor máximo oscila los 0.00155 MPa.

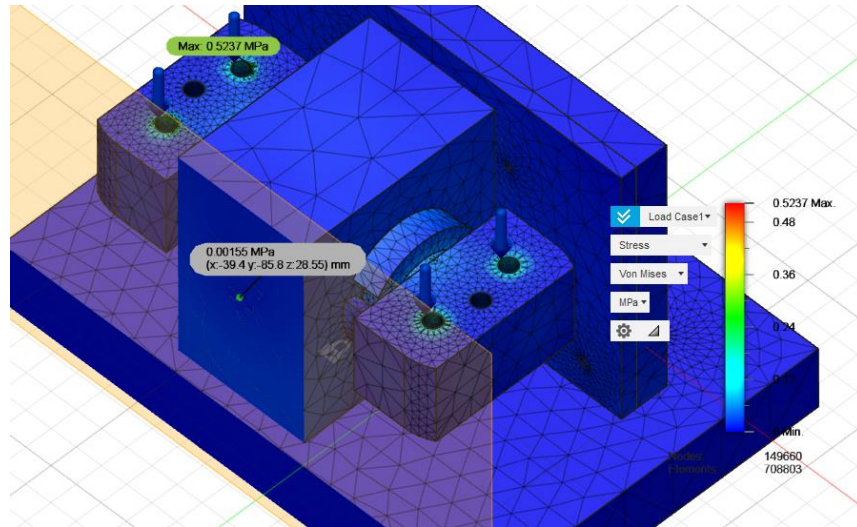


Ilustración 47 Vista de corte de análisis de estrés para el segundo eje de rotación. Fuente: Autor.

Análisis de tensión

En el análisis de tensión, se tiene una medición similar a la de análisis de estrés, debido a que el soporte del primer eje, es el que realiza una mayor tensión en su punto de unión con el soporte rectangular, pero es despreciable, debido a que tiene un análisis de tensión de alrededor de 0.0000194N.

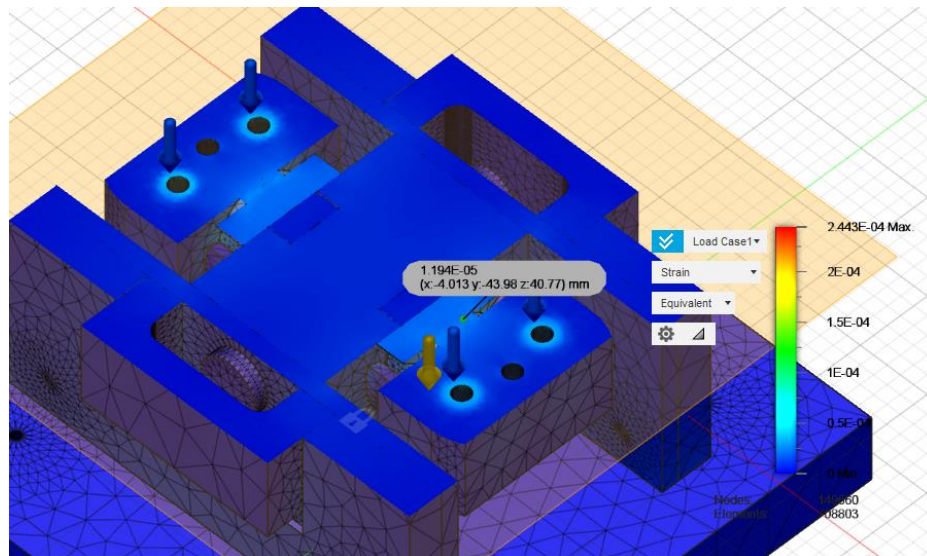


Ilustración 48 Vista de corte análisis de tensión para el primer eje de rotación. Fuente: Autor.

Se realiza el mismo proceso para los soportes del segundo eje de rotación y se observa que el límite máximo oscila el valor de 0.0000006237N, lo que resulta una cifra despreciable.

Por lo tanto, en el análisis de tensión se puede obtener datos favorables para empezar la fabricación de las piezas.

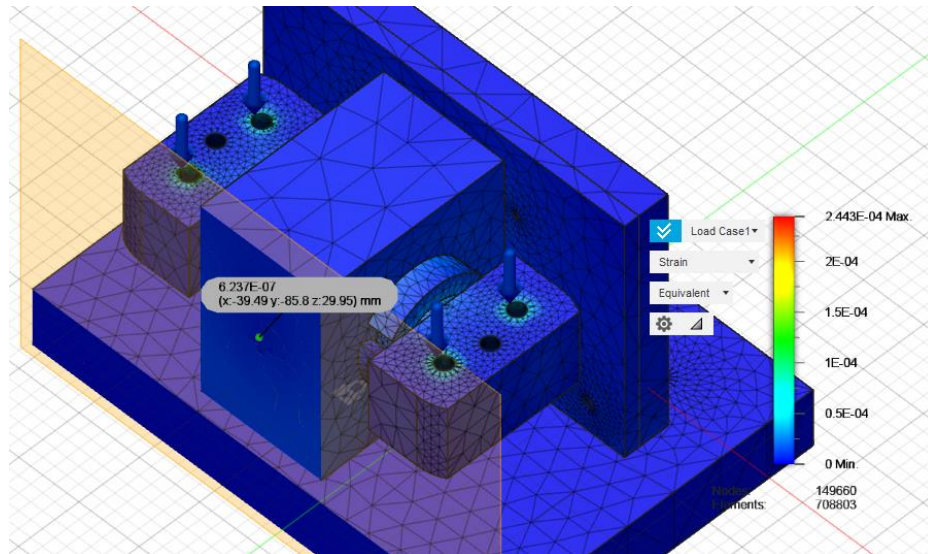


Ilustración 49 Vista de corte de análisis de tensión del segundo eje de rotación. Fuente: Autor.

Factor de Seguridad

También se realiza el análisis del factor de seguridad, donde se puede observar que cada pieza tiene un factor de seguridad mayor a 6. Dato que resulta ser más que suficiente para aprobar la fabricación de los elementos.

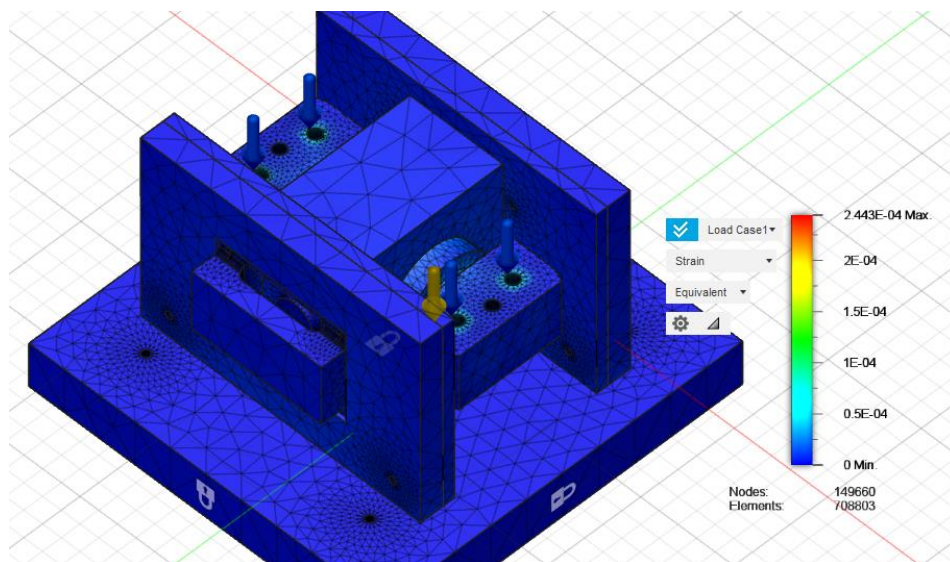


Ilustración 50 Prueba general de factor de seguridad. Fuente: Autor.

Desplazamiento

En cuanto al desplazamiento de las piezas debido a la fuerza ejercida planteada, se observa que el desplazamiento máximo ocurre en los extremos del soporte de la pieza. Siendo su valor máximo alrededor de 0.0007739 mm.

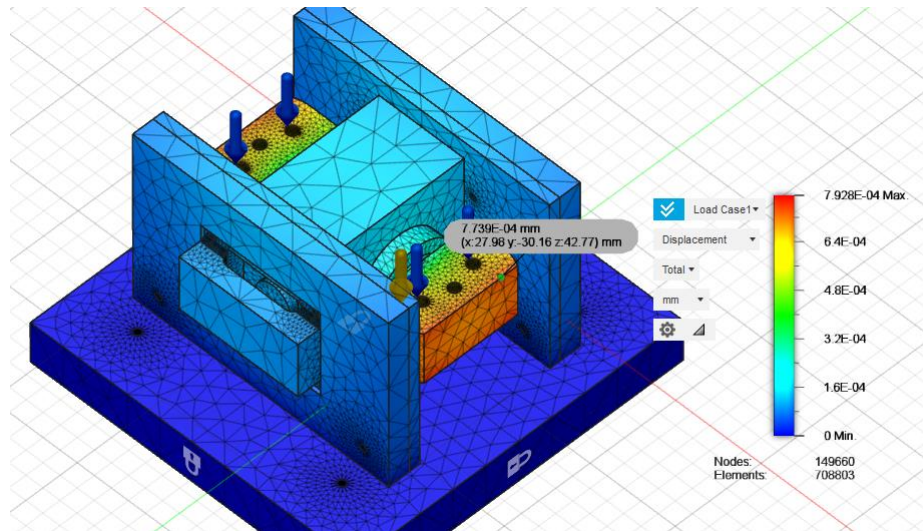


Ilustración 51 Análisis de desplazamiento del sistema. Fuente: Autor

En segunda parte, se realiza el análisis de elementos finitos cuando el dispositivo trata de levantarse, por eso para este caso, se estima que la fuerza ejercida sobre el primer eje de rotación debe ser mayor a la fuerza gravitacional.

Primero se debe tener en cuenta que para realizar la elevación del drone, este debe superar la fuerza ejercida por la fuerza gravitacional, pero ¿Cuál es el límite de fuerza de empuje para levantar el drone?, y esta pregunta se responde gracias a lo realizado en la sección 0 donde se puede observar que los resultados de cada uno de los motores limitan el empuje máximo del dispositivo, por lo tanto:

Se obtiene que cada motor puede realizar un empuje de 0,545 Kg, sin afectar el límite de corriente de operación del motor. Además, le da un tiempo de operación considerable, ya que, realizándose pruebas, con esta limitante se da un tiempo estimado de operación de 5 min.

Ahora para implementar las fuerzas, se tiene en cuenta la fuerza gravitacional junto con la fuerza ejercida por los motores, teniendo en cuenta que estas se contraponen una a la otra.

Por segunda ley de Newton resulta que la fuerza es igual al producto entre la masa del objeto y su aceleración.

En ambos casos, la aceleración es la gravedad,

pero se debe tener en cuenta que la fuerza de los motores actúa en dirección positiva al eje z, en cambio la fuerza gravitacional ejerce una fuerza en dirección negativa al eje z.

Por lo tanto:

$$\Sigma F = ma = mg \quad (22)$$

Dónde F_m : Fuerza ejercida por los motores

$$\Sigma F = F_g - F_m \quad (23)$$

F_g : Fuerza gravitacional

$$F_g = mg = (1Kg) * \left(\frac{9.802m}{s^2}\right) = 9.802N \quad (24)$$

$$F_m = 4 * \left[(0.3Kg) * \left(\frac{9.802m}{s^2}\right) \right] = 4 * (2.9406N) = 11.762N \quad (25)$$

Ambas fuerzas son tenidas en cuenta para el análisis de elementos finitos, pero se tiene en cuenta la fuerza resultante. Para ello se aplica la Σ y se reemplazan los valores de F_g y F_m .

$$\Sigma F = F_g - F_m \quad (26)$$

$$\Sigma F = 9.802N - 11.762N$$

$$\Sigma F = -1.96N$$

El signo negativo manifiesta que la fuerza resultante está ejerciéndose en misma dirección que la fuerza de los motores (en dirección z positivo).

Debe tenerse en cuenta el lugar donde se ubica la fuerza, debido a que esto es muy importante para desarrollar un buen análisis de elementos finitos del diseño.

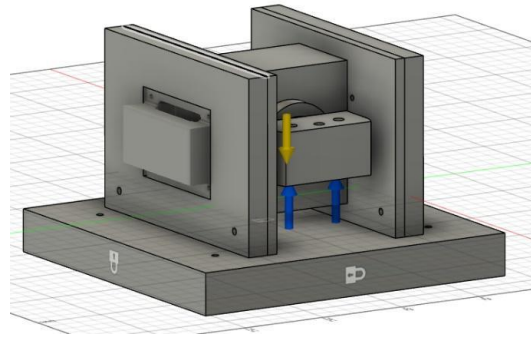


Ilustración 52 Análisis dinámico de fuerza contraria necesaria para el levantamiento del dispositivo. Fuente: Autor.

Análisis de estrés

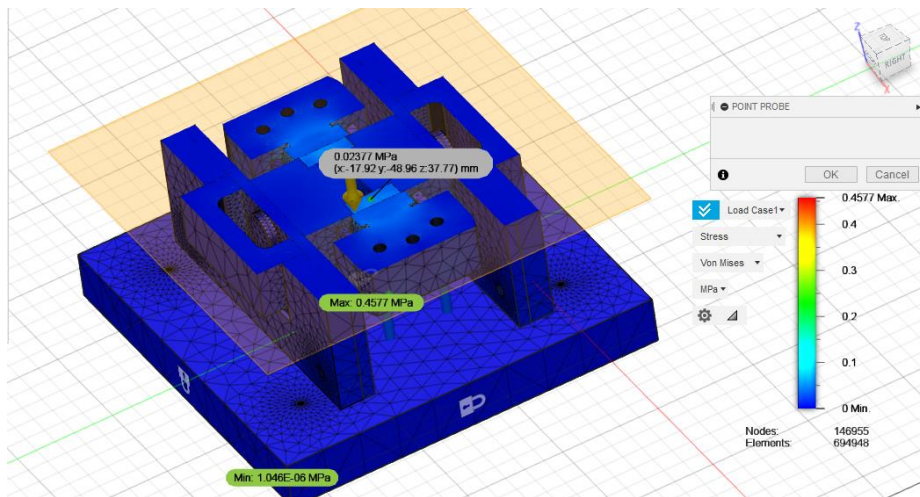


Ilustración 53 Vista de corte de análisis de estrés para el primer grado de libertad. Fuente: Autor.

Para el caso planteado, el análisis de estrés disminuye, con respecto a los 0.03656Mpa del caso anterior, esto es debido a que la fuerza resultante implementada es mucho menor, aunque su dirección sea diferente. Al igual que en el proceso anterior, se observa que la sección que sufre de mayor estrés es la que corresponde al primer eje de rotación del soporte con un valor de 0.02377Mpa, lo que representa un valor de diseño aceptable.

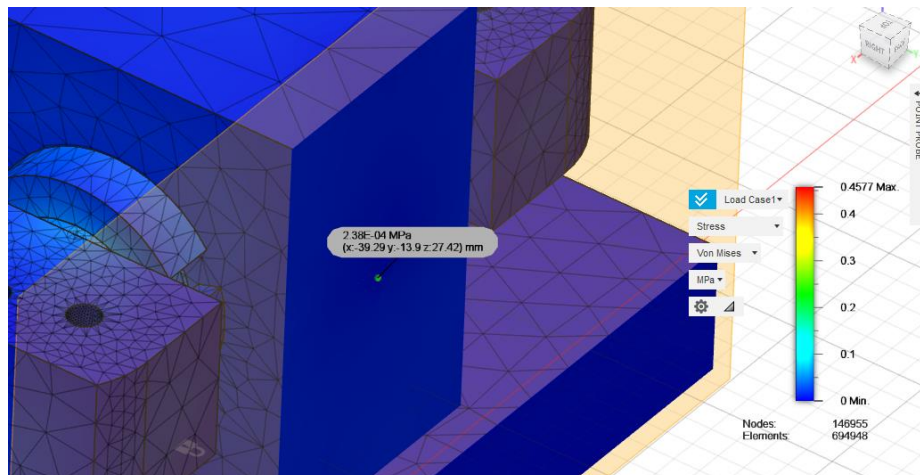


Ilustración 54 Vista de corte análisis de estrés para el segundo grado de libertad. Fuente: Autor.

Por su parte, el segundo eje de rotación también disminuye su valor con respecto al caso anterior y para este caso su valor es de 0.000238MPa.

Análisis de tensión

El análisis de tensión suele ser diferente a la primera medición, no solo porque su valor sea diferente, sino además de que la dirección en la que se aplica la fuerza es contraria a la del primer análisis de elementos finitos. Para este caso el valor de tensión es de 0.00001268N.

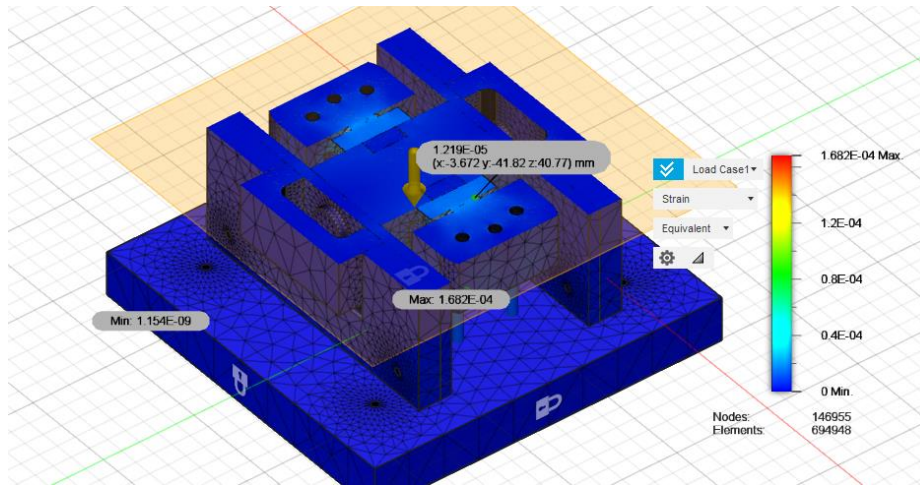


Ilustración 55 Vista de corte de análisis de tensión para el primer eje de rotación. Fuente: Autor.

En el análisis de tensión para el segundo eje de rotación se obtiene un valor de 0.0000001279N, lo cual permite comprobar lo planteado en el primer análisis, que el segundo eje no se ve tan afectado por la fuerza mecánica, como lo hace el primer eje.

Y aunque esta prueba es realizada en dirección opuesta a la primera, no incrementan sus valores de análisis de elementos finitos, sino al contrario disminuyen.

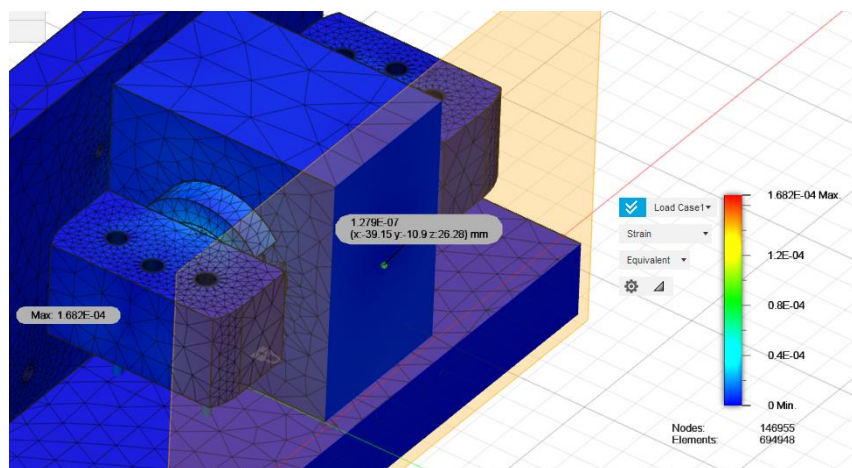


Ilustración 56 Vista de corte de análisis de tensión para el segundo eje de rotación. Fuente: Autor.

Factor de Seguridad

También se realiza el análisis del factor de seguridad, donde se puede observar que cada pieza tiene un factor de seguridad mayor a 6. Dato que resulta ser más que suficiente para aprobar la fabricación de los elementos.

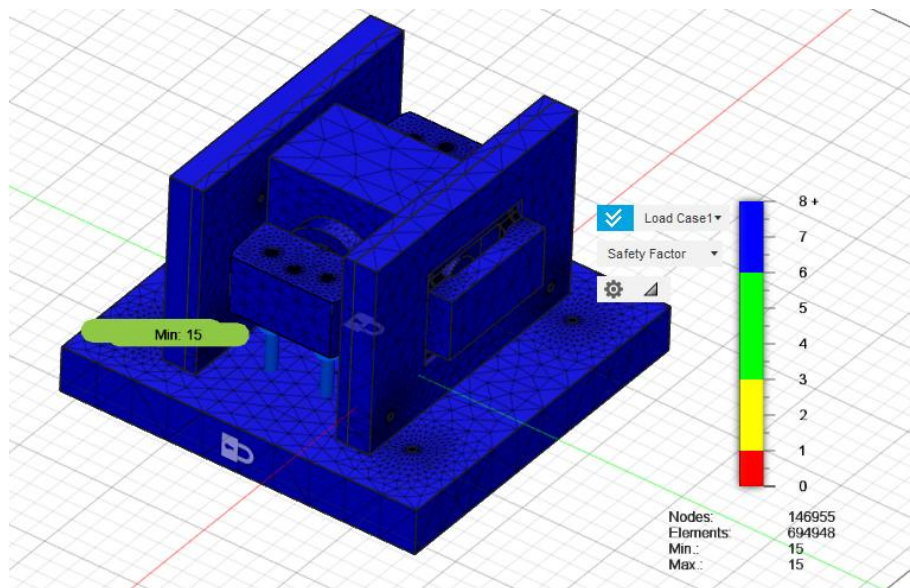


Ilustración 57 Prueba general de factor de seguridad. Fuente: Autor.

Desplazamiento

Por último, se realiza el análisis de desplazamiento de las piezas, donde se observa que el valor máximo ocurre en los extremos del soporte del primer eje de rotación, siendo un valor irrelevante, ya que su valor máximo de 0.0006114 mm.

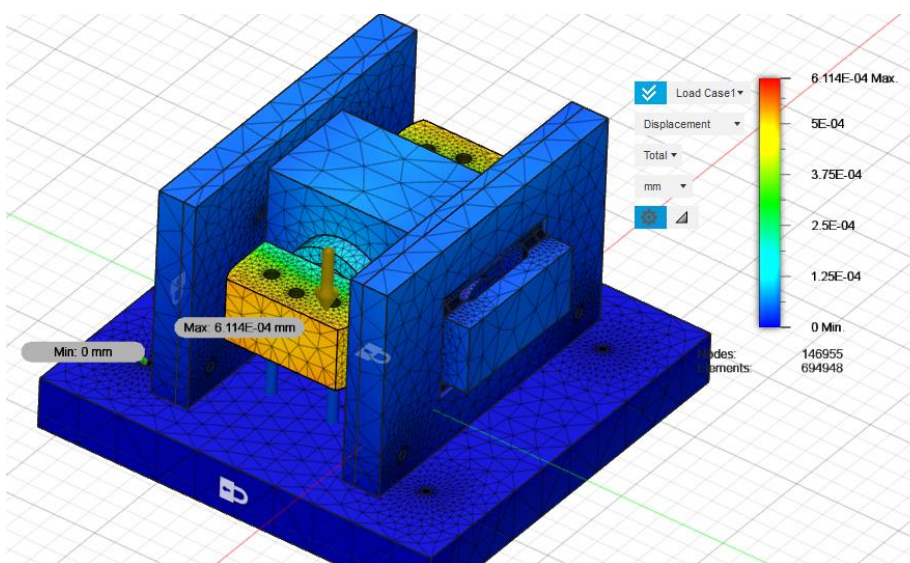


Ilustración 58 Análisis de desplazamiento del sistema. Fuente: Autor.

3.2 DISEÑO ELECTRÓNICO

3.2.1. CONEXIONES Y FACTORES A TENER EN CUENTA

El diseño electrónico de los componentes también es estudiado en la literatura [38], donde lo importante es definir las características de los elementos para establecer las conexiones y funcionamiento.



Ilustración 59 Motor Brushless. Fuente: Autor.

Para iniciar esta el actuador del sistema, el cual es el motor *brushless* mostrado en la Ilustración 59 Ilustración 59 Motor Brushless. Fuente: Autor. esta cuenta de tres cables (amarillo, negro y rojo) el cable amarillo es un cable que se debe conectar perfectamente, y el cable rojo y negro no tiene polaridad, por lo que si el motor está girando en dirección contraria se cambia la polaridad entre estos dos cables.

Lo que antecede a los motores son los dispositivos ESC, el cual como se observa en la Ilustración 60 tiene tres cables azules los cuales van conectados a los tres cables que tiene el motor, en este caso el cable de la mitad es el que va conectado al cable amarillo del motor y los de las esquinas van conectados al cable negro y rojo del motor.



Ilustración 60 Esc 30A. Fuente: Autor.

Los cables rojo y negro en la parte de arriba corresponden a la conexión con la fuente de alimentación, que en este caso corresponde a la batería de 3S. El color negro corresponde al negativo y el rojo al positivo. Aparte también cuenta con tres cables normalmente unidos donde se obtiene el pulso en microsegundos, este pulso corresponde a la señal de control generada desde el microcontrolador, y en estos tres cables, el negro corresponde al negativo del controlador, el rojo no se debe conectar (por lo cual es necesario quitarlo de nuestra conexión) y el blanco que corresponde al pulso en microsegundos.



Ilustración 61 Beaglebone Blue. Fuente: Autor.

(izquierda a derecha) corresponde al pin negativo o tierra, el segundo corresponde al positivo, pero en este caso no se usara el positivo porque la tarjeta de control ya tiene alimentación por parte de la batería de 3S, por último se encuentra el pin de control, el cual corresponde al de color blanco (o amarillo en algunos casos) en el esc.

Por último, está la conexión a nuestra fuente de energía, la cual por el pequeño espacio que usa y su alta capacidad de suministrar energía se determina por usar las baterías LiPo, más precisamente la de 3S con una capacidad de 1500mah-2200mah como se muestra en la Ilustración 62. Que pueden suministrar la suficiente corriente para los cuatro motores.



Ilustración 62 Batería LiPo Turnigy 2200 mah. Fuente: Autor

La conexión de la batería LiPo es sencilla, solo tiene dos cables (positivo y tierra), por lo que estos dos cables deben ir conectados en paralelo a los motores y a la tarjeta de control.

El microcontrolador, o la tarjeta de control en este caso. En ella se pueden encontrar ocho pines que sirven para controlar los motores brushless, ya que los ocho pueden generar pulsos en microsegundos. Como se observa en la Ilustración 61, en la parte inferior izquierda se encuentran cada uno, con tres pines para conectar con el controlador de velocidad. El primer pin

De esta forma la conexión quedaría de la siguiente manera:

Circuito de potencia

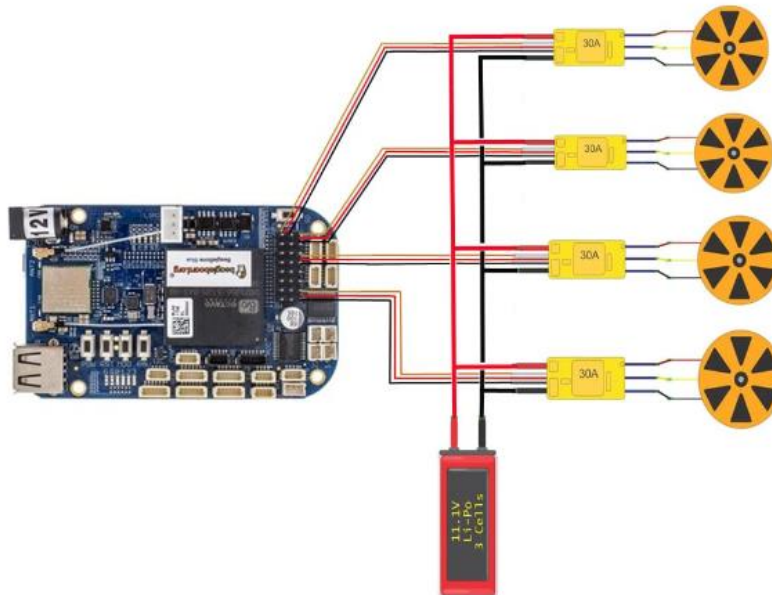


Ilustración 63 Esquema eléctrico del drone. Fuente: Autor.

Circuito de control



Ilustración 64 Esquema de control del drone. Fuente: Autor.

3.3 DISEÑO DE CONTROL

Se debe tener en cuenta que en el control de bajo nivel se encuentran los dos controladores necesarios para todo drone. Primero está el controlador de postura, el cual se refiere a el control sobre el giro del dispositivo, por ejemplo: para que el sistema se pueda mover, tiene que realizar un giro sobre los ejes “x” y “y”, haciéndolo necesario desarrollar antes de desarrollar el controlador de posición. En el controlador de posición es necesario conocer la posición actual del objeto y la posición deseada por el objeto, y de este bloque surge los angulos de control al bloque de control de postura y el empuje total del sistema (U_1).

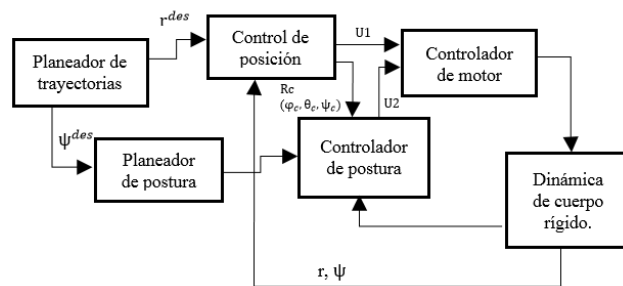


Ilustración 65 Diagrama de control dinámico del drone. Fuente: Autor.

El control cinemático permite controlar la posición del drone, este es un control de alto nivel, en el *toolbox* de robótica [4], hay una implementación disponible en Simulink. Sin embargo, dado que el objetivo de este trabajo es explorar el control de bajo nivel, es necesario estudiar la dinámica, para una revisión completa se puede consultar [1].

3.3.1. DISEÑO DE CONTROL CLASICO PD-PID (POSTURA – POSICIÓN)

Luego de analizar la dinámica de un drone en la sección 0, primero se realiza un análisis para mantener el dispositivo suspendido. Se observa que la ecuación (19) correspondiente al momento de inercia no es lineal, por lo que es necesario realizar una linearización para desarrollar el controlador.

Para ello se toman las siguientes consideraciones:

$$U_1 = mg \quad (27)$$

donde U_1 : empuje total del sistema, m : masa del dispositivo, g : gravedad

$$\theta = 0, \text{ angulo roll} \quad (28)$$

$$\phi = 0, \text{ angulo pitch} \quad (29)$$

$$\varphi = \varphi_0, \text{ angulo yaw, yaw deseado (30)}$$

$$U_2 = 0, \text{ vector de momentos en cada eje (31)}$$

$$p = 0, \text{ velocidad angular en x (32)}$$

$$q = 0 \text{ velocidad angular en y (33)}$$

$$r = 0 \text{ velocidad angular en z (34)}$$

De esta forma la ecuación (19), quedaría de la siguiente manera:

$$I [p \dot{q} \dot{r}] = [L (F_2 - F_4) L (F_3 - F_1) M_1 + M_2 + M_3 + M_4] (35)$$

En donde:

$$[L (F_2 - F_4) L (F_3 - F_1) M_1 + M_2 + M_3 + M_4] = U_2 (36)$$

Para la implementación del control clásico debe establecerse primero el algoritmo de control, el cual inicia desde un planeador de trayectorias, o señal deseada, la cual es procesada y enviada al planeador de postura que realiza los cálculos para determinar a qué inclinación debe estar el dispositivo para realizar un movimiento o estar suspendido. También le envía una señal al control de posición, en donde este realiza la acción de control y envía la señal de respuesta al bloque controlador de motor y al controlador de postura, donde el controlador de postura se encarga de realizar la acción de control para los momentos de inercia U_2 , y también envía los datos controlados al bloque controlador del motor. Con los datos generados por el control de posición y el controlador de postura, el controlador del motor procesa estas señales y las convierte en pulsos que puedan entender los motores, los cuales en el bloque de dinámica de cuerpo rígido ejercen la fuerza y momentos correspondientes, para efectuar el movimiento en el dispositivo. Luego este movimiento del dispositivo se realimenta en el sistema y genera una acción de error en el control de posición. Lo cual hará que los motores se sigan moviendo si este aun no encuentra su cometido en el planeador de trayectorias o postura.

Para efectuar el controlador del dispositivo se debe entender que como tal hay dos bloques correspondientes a desarrollo de control, el controlador de posición y el controlador de postura. El primero se debe entender que primero se genera un control de posición. Para ello, se tiene la siguiente ecuación, la cual efectúa una acción de control PD.

$$(\ddot{r}_{i,des} - \ddot{r}_{i,c}) + k_{d,i}(\dot{r}_{i,des} - \dot{r}_{i,c}) + k_{p,i}(r_{i,des} - r_{i,c}) = 0 (37)$$

Donde i , representa los ejes “x”, “y” y “z”. y r , representa la posición del dispositivo en cualquier eje de referencia.

La condición “des” quiere decir deseada y es la que proviene del bloque planeador de trayectorias, y “c” quiere decir la variable controlada. De esta forma se sabe que el valor de la aceleración controlada despejando de la ecuación $(\ddot{r}_{i,des} - \ddot{r}_{i,c}) + k_{d,i}(\dot{r}_{i,des} - \dot{r}_{i,c}) + k_{p,i}(r_{i,des} - r_{i,c}) = 0$ (37).

$$\ddot{r}_{i,c} = \ddot{r}_{i,des} + k_{d,i}(\dot{r}_{i,des} - \dot{r}_{i,c}) + k_{p,i}(r_{i,des} - r_{i,c}) \quad (38)$$

Entonces el valor de U_1 , está dado por:

$$U_1 = m(g + \ddot{r}_{3,c}) \quad (39)$$

Donde se observa que m que representa el valor de la masa, el cual termina actuando sobre todos los motores, haciendo que, si este valor cambia, afectar el controlador del sistema.

En el caso de U_2 , este contiene el valor de los momentos para los 3 ejes, por lo tanto es un vector 3×1 .

$$U_2 = [K_{p,\phi}(\phi_c - \phi) + K_{d,\phi}(\dot{\phi}_c - \dot{\phi}) \quad K_{p,\theta}(\theta_c - \theta) + K_{d,\theta}(\dot{\theta}_c - \dot{\theta}) \quad K_{p,\varphi}(\varphi_c - \varphi) + K_{d,\varphi}(\dot{\varphi}_c - \dot{\varphi})] \quad (40)$$

Y las ecuaciones para ϕ_c , θ_c y φ_c son:

$$\phi_c = \frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} - \ddot{r}_{2,c}\varphi_{des}) \quad (41)$$

$$\theta_c = \frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} + \ddot{r}_{2,c}\varphi_{des}) \quad (42)$$

$$\varphi_c = \varphi_{des} \quad (43)$$

Implementación del controlador en Simulink

Luego de ajustar los parámetros a los datos del dispositivo (anexo 3), se procede a realizar el esquema de control. Para ello se orienta de la Ilustración 67 donde aparece la distribución de bloques, en la *toolbox* de Peter Corke [4], se encuentra el modelo “sl_quadrotor” el cual es un archivo de Simulink donde se aplica el control cinemático al sistema. Apoyándose en este archivo se desarrollará el diagrama de bloques para implementar el esquema de control planteado.

En este modelo, el bloque llamado “Quadrotor” corresponde al archivo donde se calcula el modelo dinámico del sistema, conectándole la información correspondiente a las velocidades angulares de cada uno de los motores, peso del dispositivo, matriz de inercia, tamaño de las hélices, entre otros parámetros. También cuenta con la facilidad de poder especificar la posición inicial donde se encuentra el dispositivo.

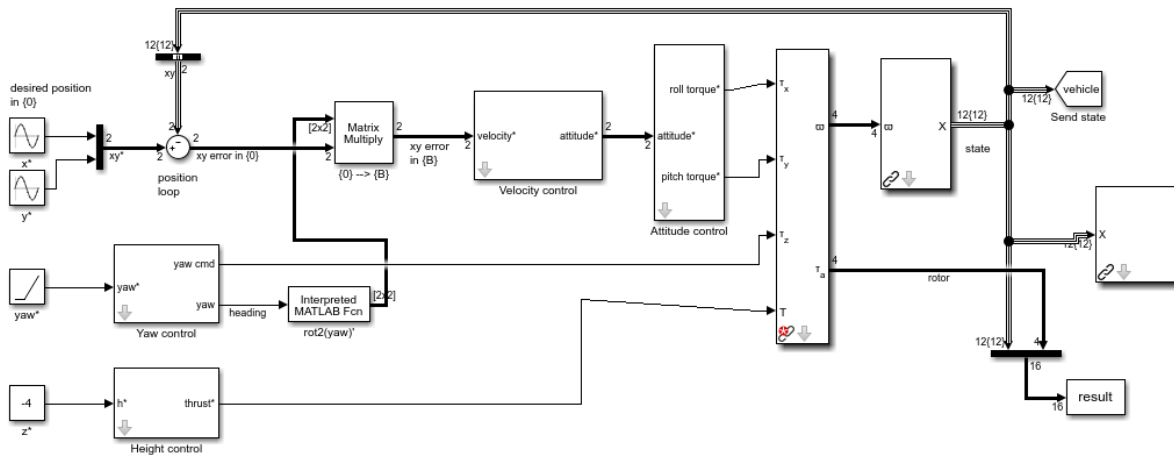


Ilustración 67 Aplicación *Sl_quadrotor* de la *toolbox* de Peter Corke. Fuente: *Toolbox* Peter Corke.

Para poder enviar las velocidades angulares a este bloque de Simulink es necesario conocer cómo se calcula mediante los momentos de inercia y la fuerza de empuje de los motores, por eso en la *toolbox* se aplica otro bloque en Simulink donde se realiza el proceso de cálculo de velocidades angulares, a la cual es necesario suministrar la información de empuje total del sistema y momentos de inercia del sistema “x”, “y” y “z” respectivamente.

También en este bloque se realiza el proceso de conversión de fuerza de empuje de cada motor a velocidades angulares, analizado en la sección 0. Y también se puede especificar las velocidades angulares máximas de los motores, información útil para que no ocurra un alto consumo de corriente y no ocurra sobrecarga de corriente en los dispositivos.

Pero antes de llegar al bloque del modelamiento dinámico del sistema, se puede observar el bloque “attitude control” control de postura, y el “velocity control” control de velocidad. En este caso es necesario cambiar ambos bloques por unos creados a partir de la información estudiada en la sección 0. Para el control de postura es necesario suministrar los valores de los ángulos actuales del sistema y de los deseados, generados desde el bloque de control de posición, cada uno con su respectiva primera y segunda derivada, y para el control también

es necesario suministrar las posiciones actuales y deseadas del sistema, para cada uno de los ejes de desplazamiento, como se observa en la Ilustración 68.

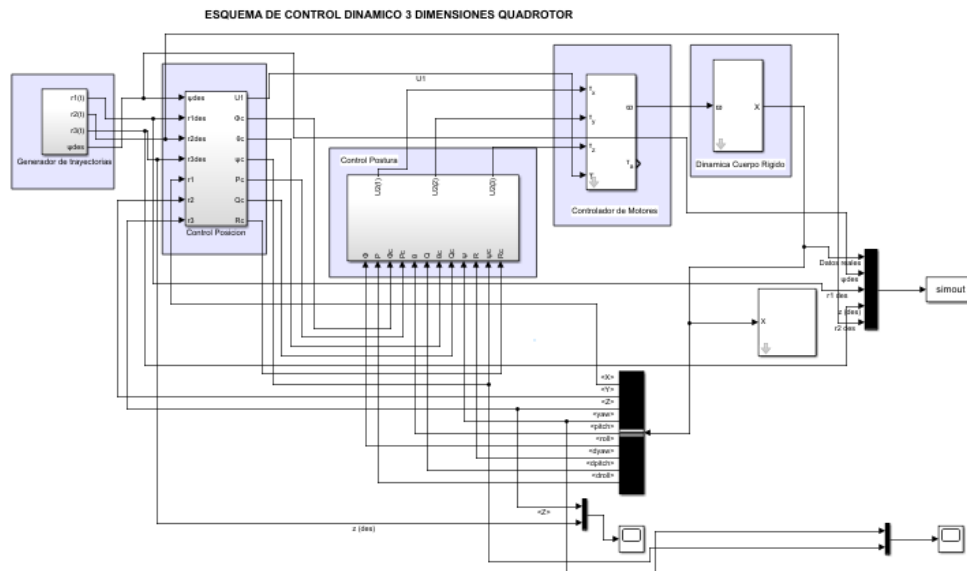


Ilustración 68 Control dinámico implementando las ecuaciones de control del curso "Aerial Robots". Fuente: Autor.

Dentro del bloque de control de posición se generan los ángulos y velocidades angulares deseadas, a partir de las posiciones actuales y deseadas del sistema, además de establecer el empuje total del sistema, como se muestra en la Ilustración 69.

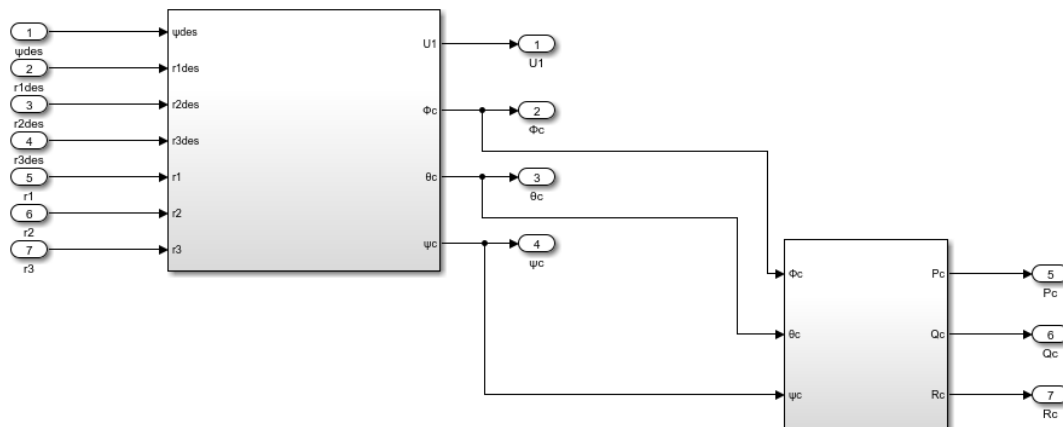
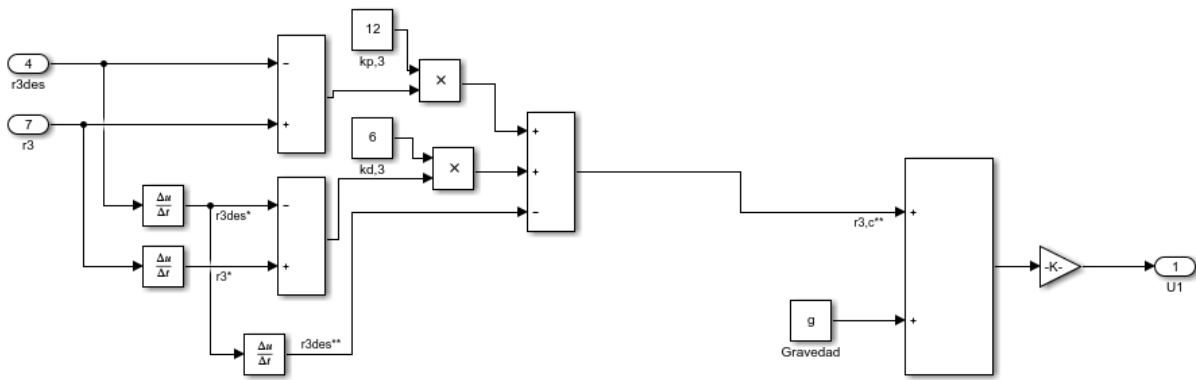


Ilustración 69 Controlador de postura. Fuente: Autor.

Para el cálculo de las posiciones deseadas se maneja la ecuación $\ddot{r}_{i,c} = \ddot{r}_{i,des} + k_{d,i}(\dot{r}_{i,des} - \dot{r}_{i,c}) + k_{p,i}(r_{i,des} - r_{i,c})$ (38). Para el caso del control de altura se puede observar en la Ilustración 70 como queda representado mediante el diagrama de bloques en Simulink, donde “k” representa la masa del dispositivo, “g” la gravedad, “r3des”, posición en z deseada, “r3” posición en z actual, “Kp, 3” constante proporcional de control, “Kd, 3” constante derivativa



de control.

Ilustración 70 Controlador de posición. Fuente: Autor.

Para el cálculo de posiciones angulares deseadas se realizan aplicando las ecuaciones $\phi_c = \frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} - \ddot{r}_{2,c}\varphi_{des})$ (41), $\theta_c = \frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} + \ddot{r}_{2,c}\varphi_{des})$ (42) y $\varphi_c = \varphi_{des}$ (43), en la

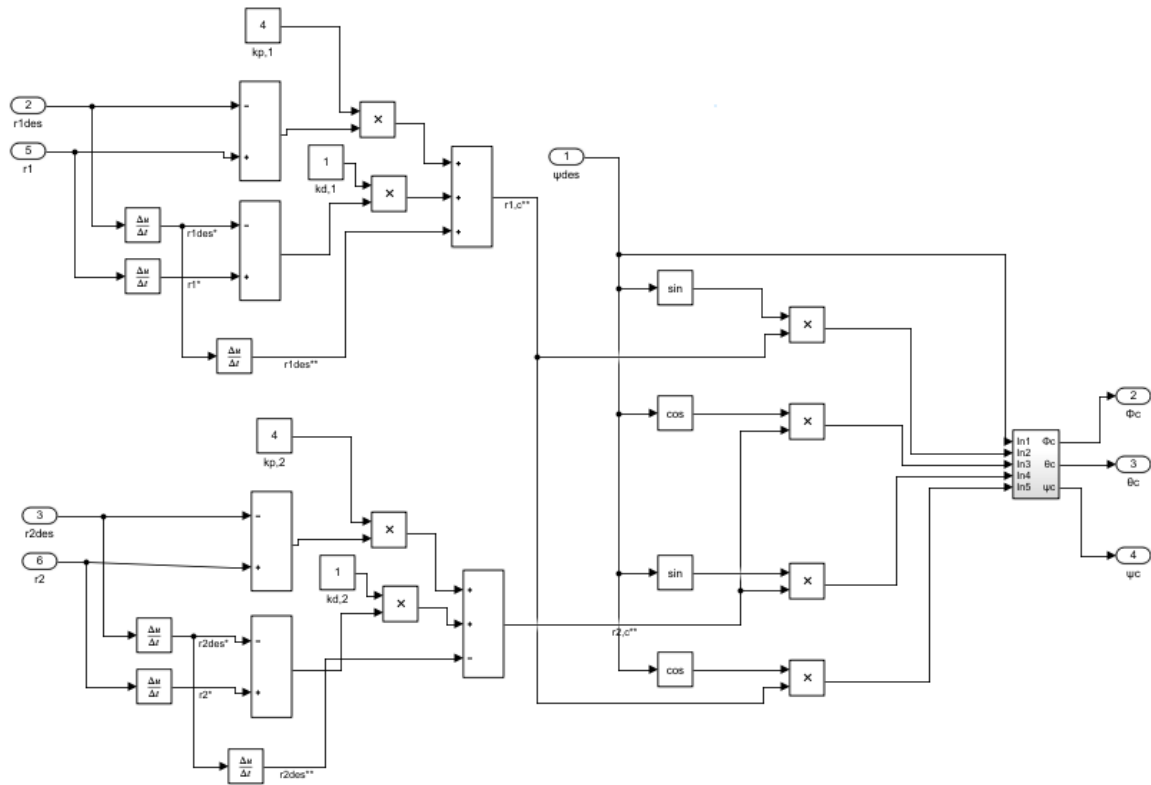
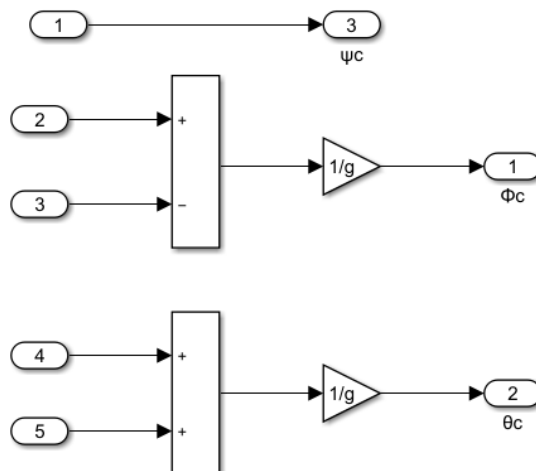


Ilustración 71 Controlador de postura. Fuente: Autor.

figura se puede observar el esquema en bloques de este proceso.

En la Ilustración 71 se observa un subsistema donde se generan las señales de salida de posiciones angulares controladas, este subsistema se realiza a partir de la operación de la

ecuaciones $\phi_c = \frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} - \ddot{r}_{2,c}\varphi_{des})$ (41), $\frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} + \ddot{r}_{2,c}\varphi_{des})$ (42) y $\varphi_c = \varphi_{des}$ (43).



$\frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} - \ddot{r}_{2,c}\varphi_{des})$ (41), $\theta_c = \frac{1}{g}(\ddot{r}_{1,c}\varphi_{des} + \ddot{r}_{2,c}\varphi_{des})$ (42) y

Ilustración 72 Ángulos de control Yaw, Pitch y Roll.

Luego de obtener los datos de postura deseada, se realizó un siguiente bloque en Simulink donde se calcula las velocidades angulares deseadas, a partir de la ecuación (20).

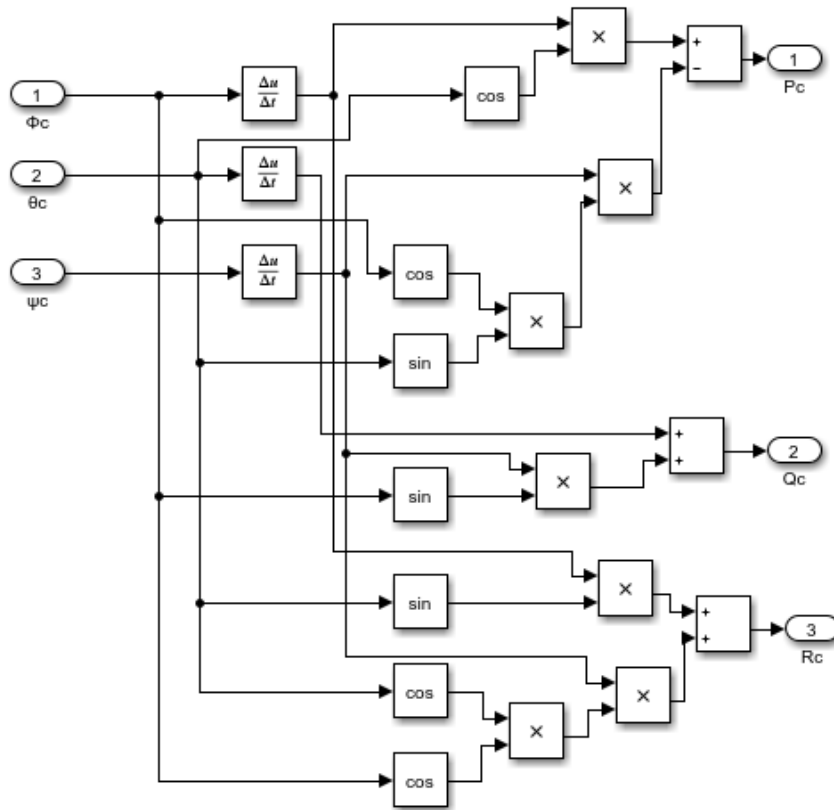


Ilustración 73 Aceleraciones angulares de control. Fuente: Autor.

En este caso las velocidades angulares deseadas serán representadas por “Pc”, “Qc” y “Rc” para las velocidades angulares de roll, pitch y yaw, respectivamente

Para el esquema del control de postura se aplica la ecuación $U_2 = [K_{p,\phi}(\phi_c - \phi) + K_{d,\phi}(\dot{\phi}_c - \dot{\phi}) + K_{p,\theta}(\theta_c - \theta) + K_{d,\theta}(\dot{\theta}_c - \dot{\theta}) + K_{p,\varphi}(\varphi_c - \varphi) + K_{d,\varphi}(\dot{\varphi}_c - \dot{\varphi})]$ (40). Por la cual se toman los datos de posiciones velocidades y aceleraciones angulares, para calcular el vector U2 que almacena los momentos en los 3 ejes de rotación.

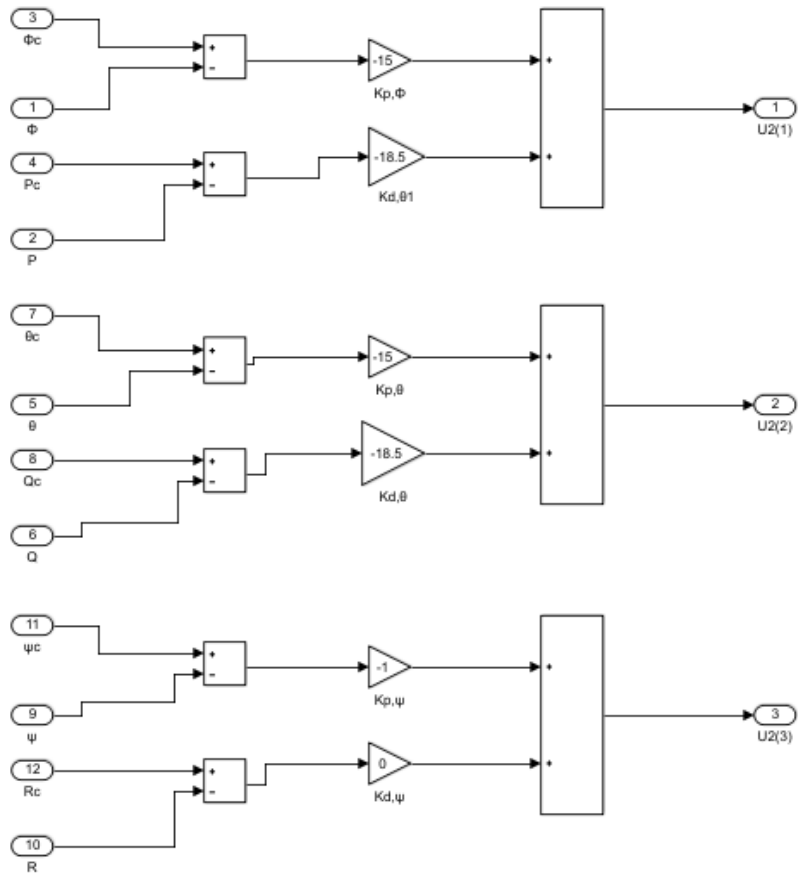


Ilustración 74 Controlador PD para el los tres ejes de rotación. Fuente: Autor.

Luego este vector, junto con el valor de empuje total es transferido al bloque de Simulink “Controlador de motores”, donde este por su parte genera las velocidades angulares de los motores y son enviadas a la planta del sistema.

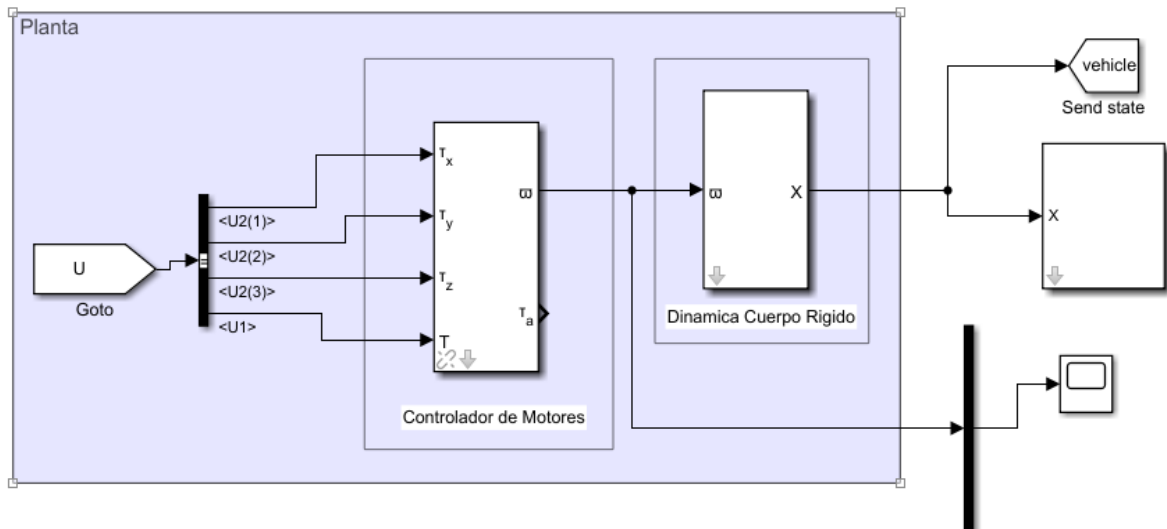


Ilustración 75 Esquema de la planta del sistema. Fuente: Autor.

Sintonización del Controlador Clásico

Teniendo en cuenta que se necesitan sintonizar un total de 6 controladores, correspondientes a cada grado de libertad del dispositivo, se procede a sintonizar primero los controladores de postura. Iniciando con el control de giro correspondiente al eje z. Por lo tanto, para el control de posición se ponen un controlador calculado mediante prueba y error que funciona de manera aceptable, para que no afecte al controlador de postura al cual se quiere hacer sintonización.

Para realizar la prueba se realiza el lazo cerrado y dependiendo del sobre pico y tiempos de asentamiento y respuesta se calculan las constantes de control.

Realizando la prueba, se obtiene que la respuesta del sistema es la siguiente.

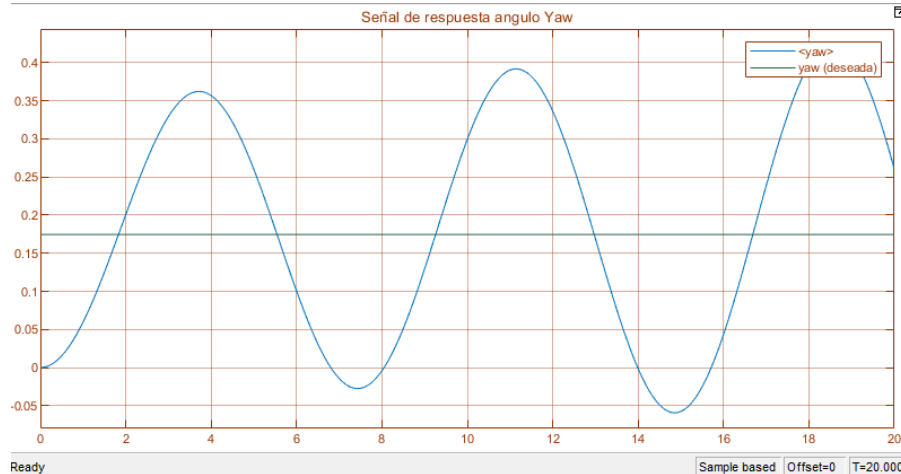


Ilustración 76 Respuesta del sistema en lazo cerrado para el giro en el eje "z", sin controlador. Fuente: Autor.

Lo que representa una mala respuesta del sistema, teniendo sobre picos muy elevados y sin estabilización para cuando el tiempo tiende a infinito. Entonces se procede a ingresar un control PD por método Heurístico, teniendo en cuenta la influencia de cada parámetro de control clásico.

Para esto se encuentra que un controlador PD con valores de $P = -10$ y $D = -20$ es aceptable para el comportamiento del dispositivo como se observa en la Ilustración 77.

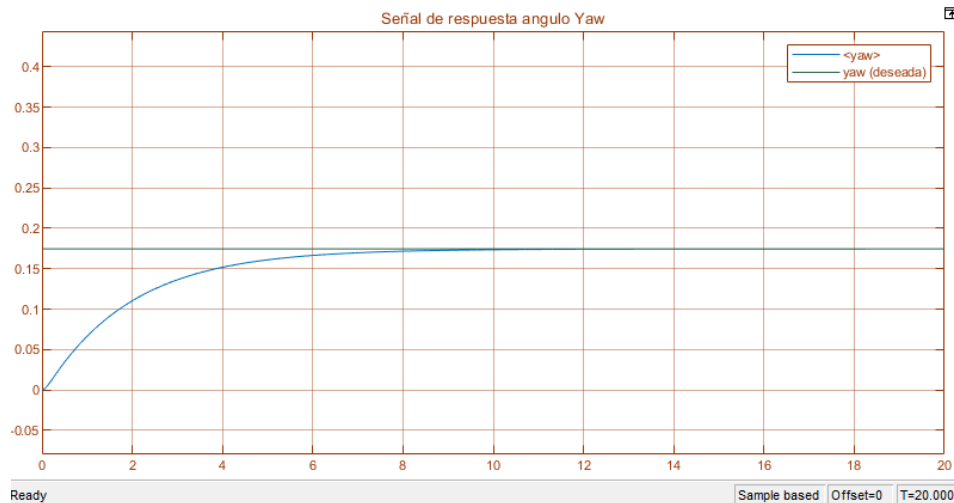


Ilustración 77 Respuesta de sistema, controlador Heurístico. Fuente: Autor.

Para el diseño del controlador de pitch y roll, se realiza el mismo procedimiento y dependiendo de la respuesta se establece el modelo de control. Con la diferencia de que no se puede imponer un pulso deseado, ya que el control de postura depende de la posición actual y de la deseada, por lo que se procede a realizar el análisis del control, generando un error de posición en donde el dispositivo solo tenga que mover ese ángulo.

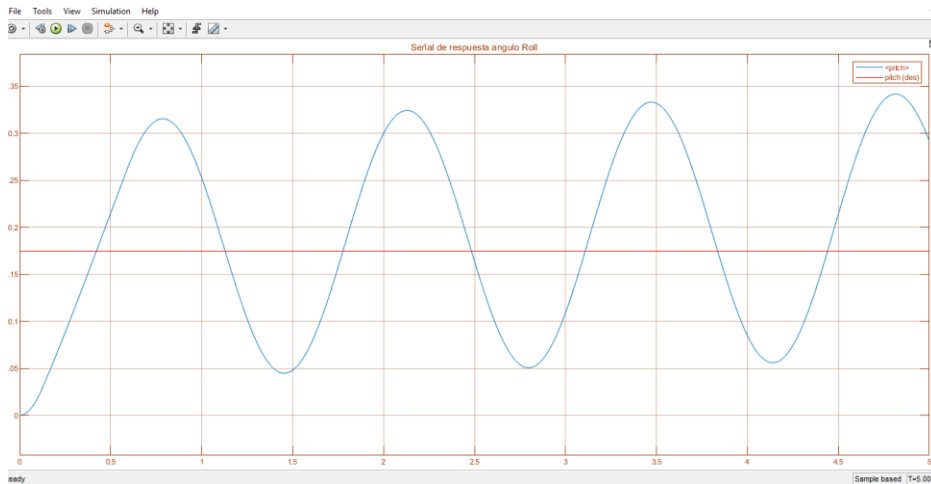


Ilustración 78 Respuesta del sistema en lazo cerrado para el giro del eje "y". Fuente: Autor.

En la Ilustración 78 se observa que el sistema es oscilatorio por lo cual no se puede establecer una estrategia de sintonización de control en lazo cerrado para el ángulo roll y pitch. Por otra parte la Ilustración 79 muestra que la posición en el eje y también es afectada, incrementando el error de posición levemente durante el proceso.

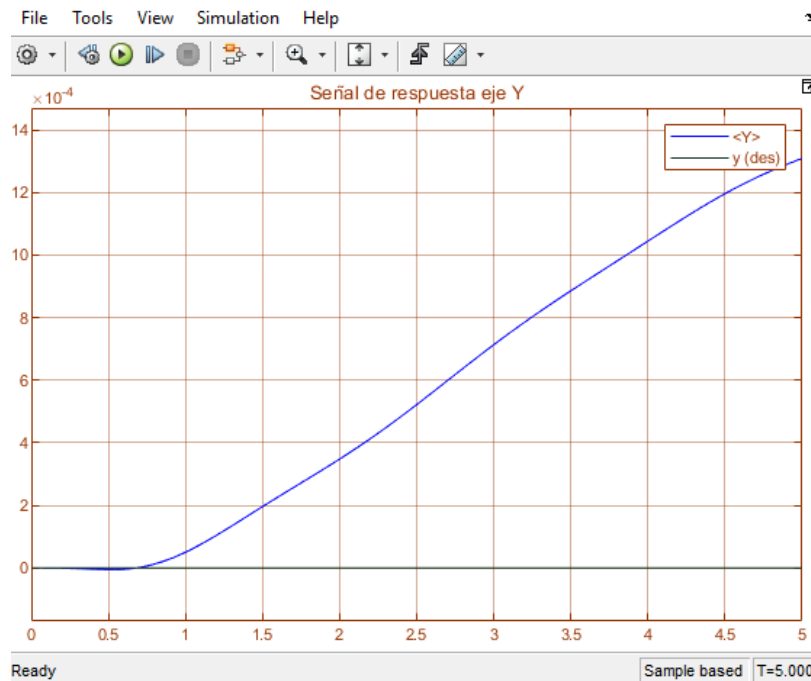


Ilustración 79 Posición del dispositivo para la respuesta del sistema en el giro en el eje "y". Fuente: Autor.

Por lo tanto, al igual que en el control de postura de Yaw, se procede a realizar un controlador PD de manera heurística, ajustando los controladores dependiendo de la respuesta del sistema en simulación.

Para el desarrollo del controlador de postura para roll, se implementa el siguiente controlador: $P = -15$, $D = -18.5$. Con el cual se obtiene la siguiente señal de respuesta para la misma señal deseada en el paso anterior.

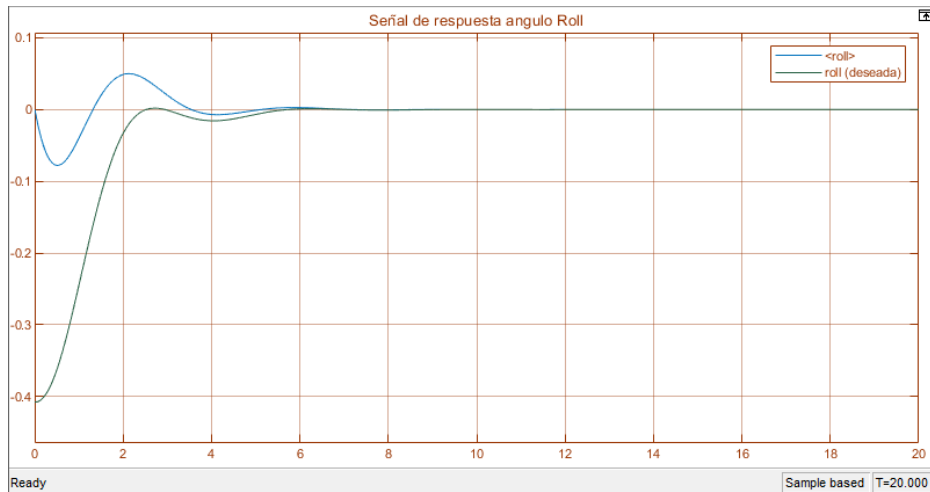


Ilustración 80 Respuesta de giro del eje "y" implementando controlador Heurístico PD Fuente: Autor.

Por su parte el controlador de posición obtuvo un desarrollo aceptable con un tiempo de asentamiento de 2.5 segundos.

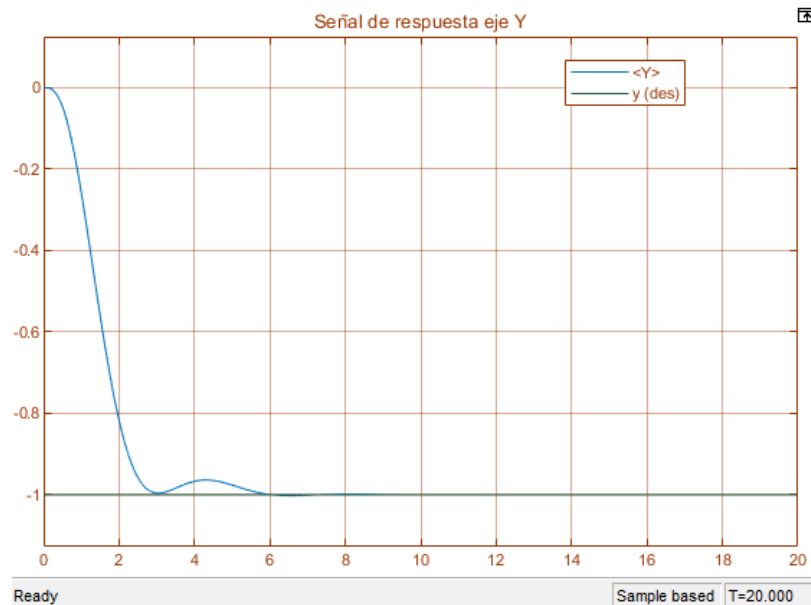
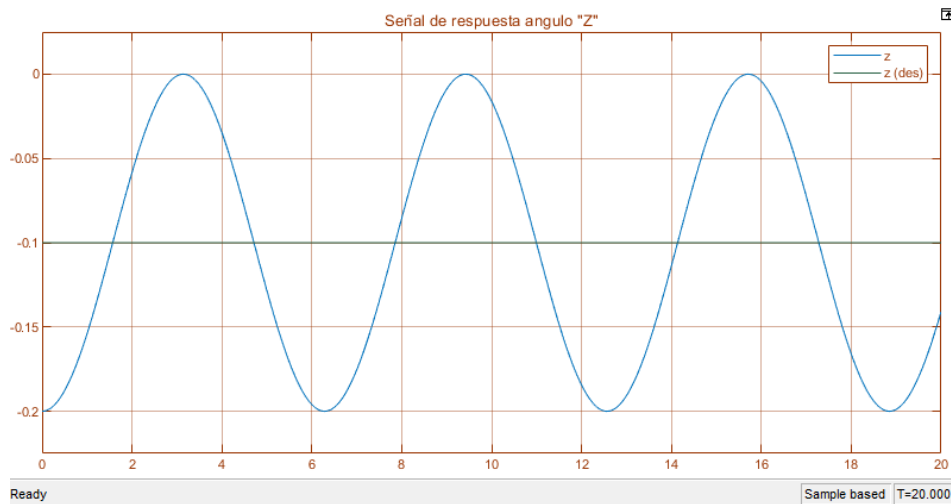


Ilustración 81 Respuesta de la posición con controlador en lazo cerrado para el giro del eje "y". Fuente: Autor.

Obteniendo este resultado se implementa el mismo valor del controlador para el ángulo Pitch, del cual se tiene la misma respuesta que para roll. Con esto, ya se tiene resuelto el controlador de postura en simulación.

Ahora se procede a realizar la sintonización del controlador de posición. Para esto se establece solo realimentación para la posición a la cual se desea realizar la sintonización. En este caso se inicia con el proceso de sintonización de la posición del dispositivo en el eje z, ya que todos los motores deben actuar para ejecutar un óptimo movimiento.



*Ilustración 82 Respuesta del sistema de giro en el eje "z" con señal realimentada sin controlador.
Fuente: Autor.*

Para este caso se observa en la Ilustración 82 que la respuesta del sistema es oscilatoria y no tiene tiempo de asentamiento, por lo que desarrollar un sintonizador resulta complicado usando los métodos dos puntos de Smith o cualquiera en el que se pueda usar la gráfica de respuesta del sistema.

Por ello se procede a realizar la sintonización mediante método heurístico y analizando la respuesta del sistema. Para el controlador de esta posición se establece un controlador clásico $P=2$ y $D=3$ y se procede a observar la gráfica de respuesta del sistema.

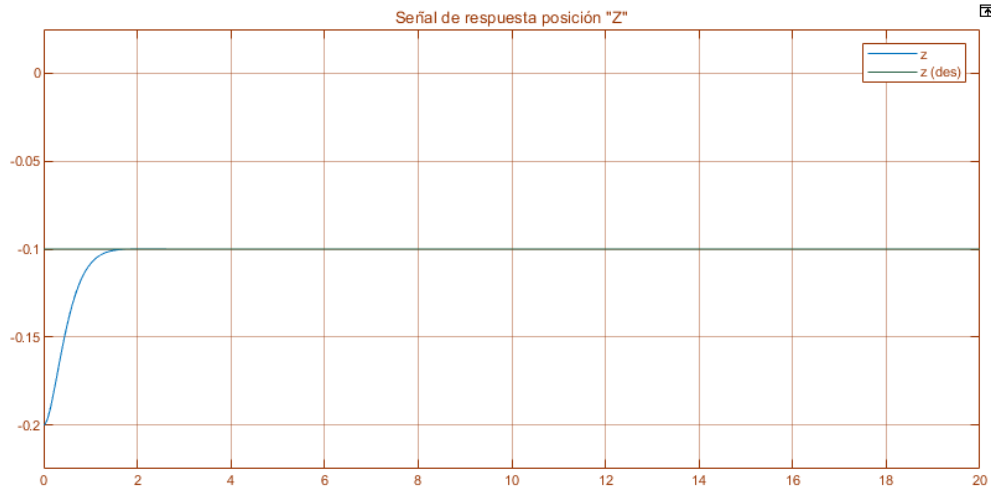


Ilustración 83 Respuesta del sistema de posición en el eje "z" con señal realimentada y controlador heurístico. Fuente: Autor.

En la Ilustración 83 se observa que el controlador propuesto es aceptable, ya que tiene un tiempo de asentamiento de 1.8 seg, y sin sobre pico notable.

Por otra parte, para la sintonización de controlador de la posición "x" y "y" interviene un cabeceo por parte del dispositivo, pero como se comprueba en pasos anteriores, el controlador de postura ya fue realizado de manera aceptable, así que se procede a generar una señal deseada sin acción de control y de esa forma ejercer movimiento sobre el dispositivo en la dirección del eje "x" y luego en la dirección "y" del sistema de referencia del dispositivo. Cabe tener en cuenta que debido a que el sistema es configurado de forma "X" para el movimiento del dispositivo en un eje de referencia, es necesario que interactúen los cuatro motores simultáneamente, por lo que es más complejo establecer un controlador para cuatro motores dependiendo de un ángulo que para dos, como lo sería en una configuración "+".

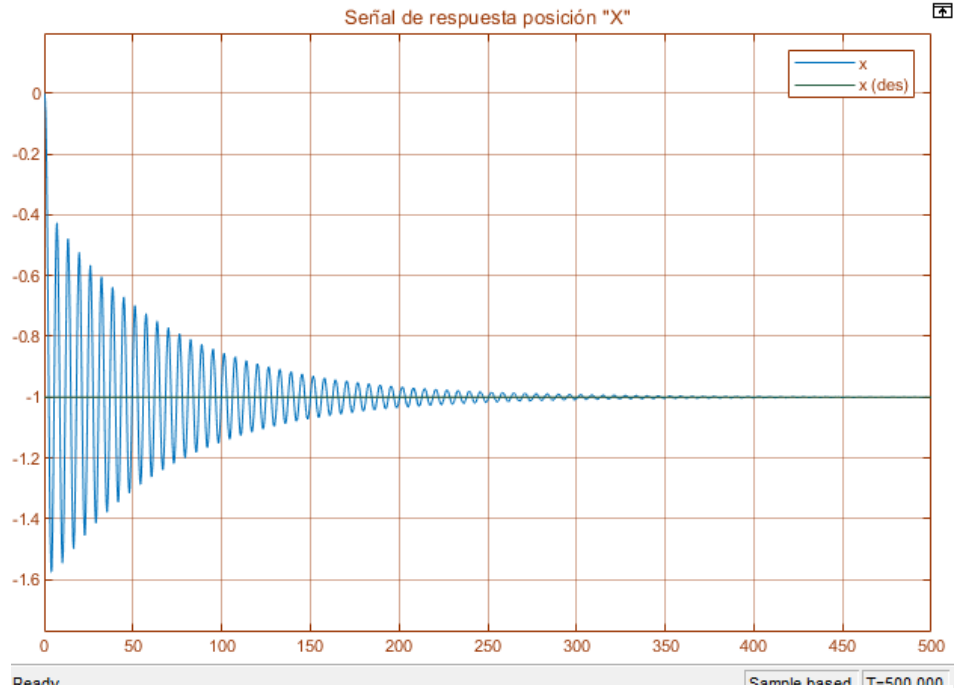


Ilustración 84 Respuesta del sistema de posición en el eje "x" con señal realimentada sin controlador. Fuente: Autor.

Como se observa la señal es muy oscilatoria, y tiene un tiempo de asentamiento bastante elevado, alrededor de 75 segundos para tener un 20% de error con la señal deseada. Por lo que, es necesario hacer la sintonización del controlador y también se realiza por método heurístico.

Por medio del método heurístico se obtiene una buena respuesta como se observa en la Ilustración 85. En este, se puede establecer un buen tiempo de asentamiento con una pequeña

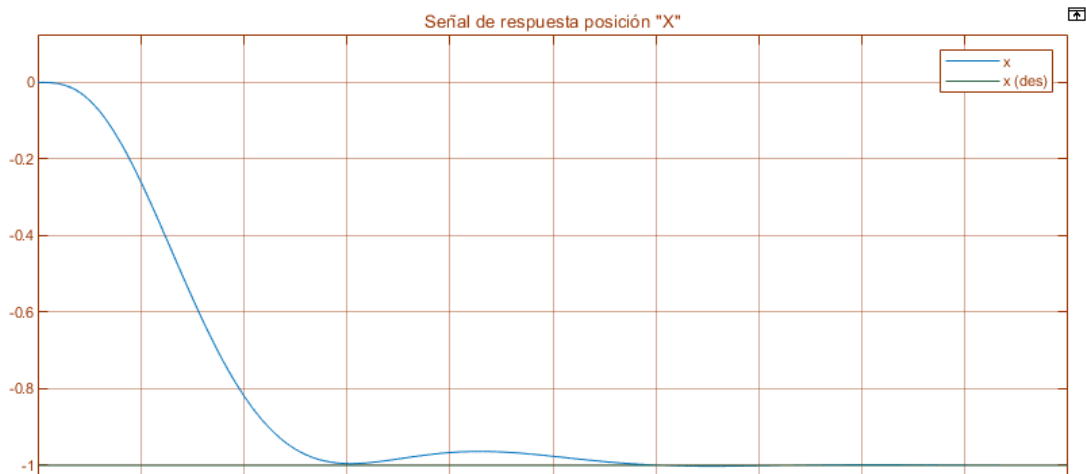


Ilustración 85 Respuesta del sistema de posición en el eje "x" con señal realimentada con controlador heurístico. Fuente: Autor.

oscilación a los 4 segundos con un controlador clásico $P=4$ y $D=1$ que es funcional tanto para el eje “x” como para el eje “y”.

Cabe tener en cuenta que en el desarrollo de la simulación se tiene en cuenta muchos parámetros que no son correspondientes a este medio, por ejemplo, datos como: la densidad del aire, o la presión de esta, el ángulo de ataque de las hélices, entre otros. Por lo que, al desarrollar la implementación del controlador en la plataforma de pruebas, será necesario ajustar de nuevo el controlador a los datos físicos del sistema. Anexo 3

3.3.2. DISEÑO DE CONTROL ADAPTATIVO

Para realizar el diseño del control adaptativo es importante entender cómo funciona su esquema de control. Como se observa en la Ilustración 86, el control adaptativo permite modificar las constantes de control, dependiendo al error que se genera entre el modelo de referencia y la salida de la planta.

También es necesario implementar una buena regla de adaptación, que va a ser nuestra norma para hacer el ajuste de las constantes de control.

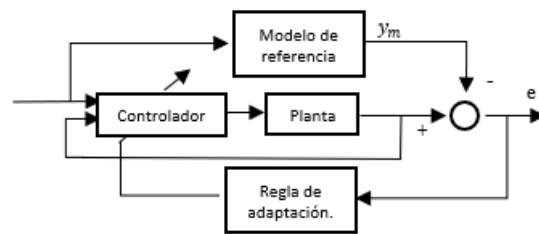


Ilustración 86 Esquema de control adaptativo. Fuente: Autor.

En el desarrollo del proyecto se usa inicialmente condicionales para la implementación de la ley de adaptación. Para establecer una buena regla de adaptación, es necesario conocer la señal de error, en las diferentes situaciones que va a cambiar la masa del sistema, de esta manera, conociendo la respuesta, se puede establecer su regla de adaptación.

Mediante el uso de condicionales se realiza un switcheo en el controlador del modelo clásico, dependiendo del error obtenido entre el modelo de referencia y la planta real. Este switcheo es necesario realizarlo evaluando rangos, ya que el modelo puede tener errores mínimos debido a que el sistema no es perfecto, por eso se realiza para variaciones de peso considerables, alrededor del 20% del peso dispositivo.

El modelo de referencia se toma igual que el modelo normal del sistema sin afectar el peso, y también se tiene en cuenta en qué momento se realiza un movimiento en el eje z para que no detecte este, como un error entre el modelo de referencia y la planta .

Proceso de entrenamiento del control adaptativo

Primero se analiza la respuesta del sistema con una variación en el peso de 0.2kg al segundo 5, haciendo la comparación con el modelo de referencia que va a ser el mismo del sistema inicial con el valor de masa constante en la planta. De esta manera se puede ver en la gráfica que genera una señal de error entre el controlador implementado para el caso inicial y el del sistema con carga como se observa en la Ilustración 87.

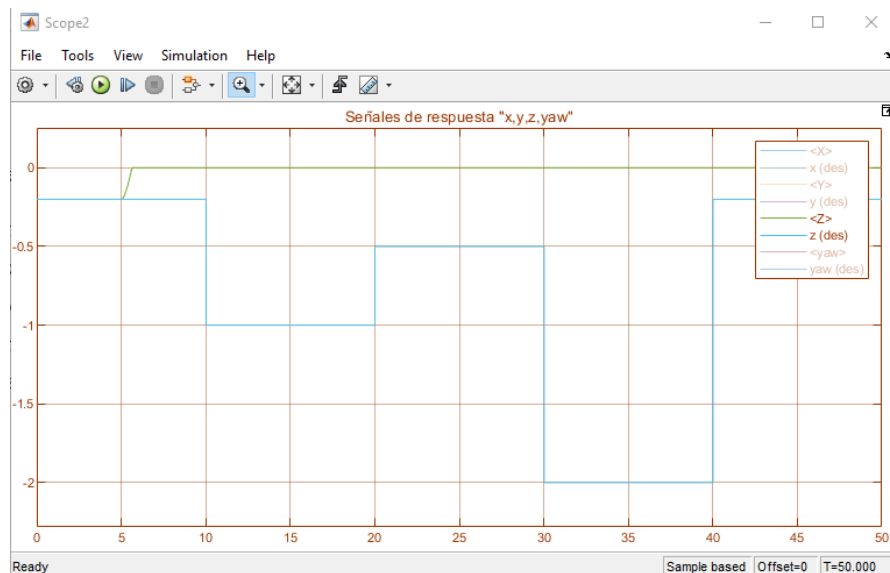


Ilustración 87 Respuesta del sistema con incremento de masa en el segundo 5.

Se observa que la señal luego de agregar la masa decae (en el sistema de referencia los ejes “y” y “z” están al contrario, por eso sube la señal) el dispositivo y no se levanta debido al error que ocurre en su control.

Luego de observar la dinámica del modelo con variación en el peso, se reconoce que hay dos formas solucionar este problema, uno es ajustando el valor del peso, y el otro es solo ajustar el valor del controlador. Para el primer caso se debe realizar el algoritmo de ajuste del peso del sistema y observar cómo funciona el sistema por si necesita una variación en el controlador para el peso diferente, y para el segundo se debe realizar el ajuste del controlador de acuerdo al error con el modelo de referencia.

Control adaptativo ajustando el controlador

Así que se procede a realizar el segundo caso donde se debe cambiar el valor del controlador clásico de acuerdo a la respuesta del sistema. Para ello se busca desarrollar un controlador

PD para el sistema con la masa alterada, el cual resulta siendo $P=1000$ $D=250$, que tiene una señal de respuesta rápida pero en el segundo 32 como se muestra en la Ilustración 88, hay un sobre pico de casi 20 centímetros, por lo que es necesario encontrar otro controlador PD.

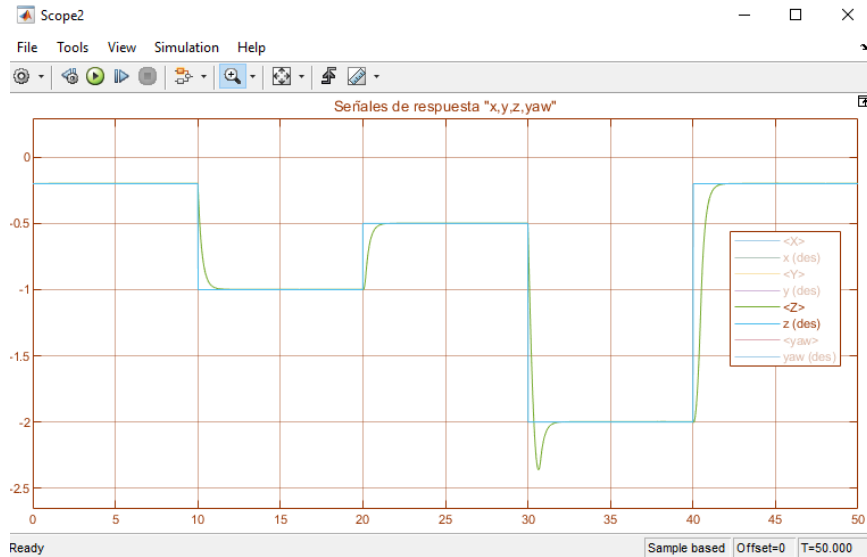


Ilustración 88 Respuesta del sistema masa=1kg, controlador $P=1000$ $D=250$

En otro controlador, se encuentra también el controlador clásico $P=1000$, $D=1400$. Con el cual no se sufre de sobre picos en el sistema como se muestra en la Ilustración 89, pero se tiene una respuesta lenta en el sistema, lo cual se propone hacer el switcheo entre estos dos para tener un sistema que responda de manera más rápida y no tenga sobre oscilaciones.

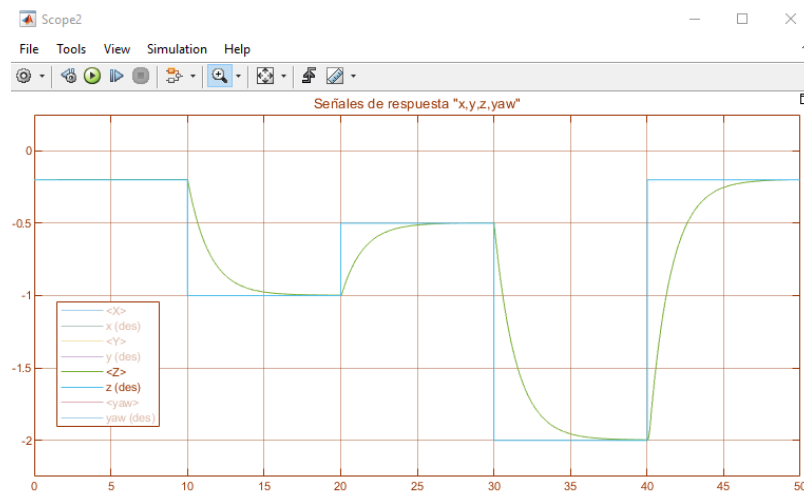


Ilustración 89 Respuesta del sistema masa 1kg, controlador $P=1000$ $D=1400$

Conociendo los controladores en los cuales se va a realizar el switcheo, se observa primero el error entre el modelo de referencia y el sistema de cada caso, teniendo el cambio en la masa en el sistema durante todo el tiempo de simulación y almacenando esta información del error que se observa en la Ilustración 90, cuando el sistema tiene el controlador $P=1000$ y $D=1400$.

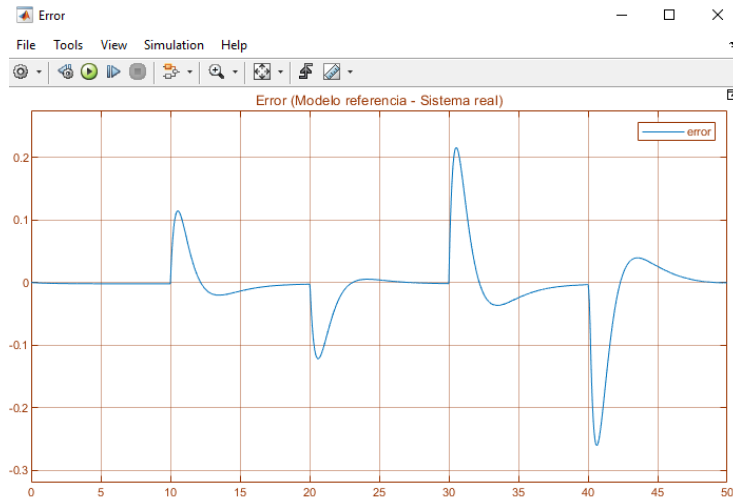


Ilustración 90 Error entre modelo de referencia masa 1 kg y controlador $P=2$ $D=3$ y sistema real con masa 1.2kg, controlador $P=1000$ $D=1400$

Se observa en la gráfica que el valor máximo es de 0.22 y el mínimo es -0.27 cuando ocurren movimientos en el eje z, lo que quiere decir que cuando la señal es estable debe permanecer el error en 0. Lo cual es necesario tener en cuenta para también analizar la siguiente prueba, cuando se tiene el controlador $P=2$, $D=3$ que es el indicado para la masa de 1kg y se aplica para la nueva masa, donde observa que el valor de la error tiende a negativo como se muestra en la Ilustración 91 y que solamente en el tiempo = 0 el valor del error es 0.

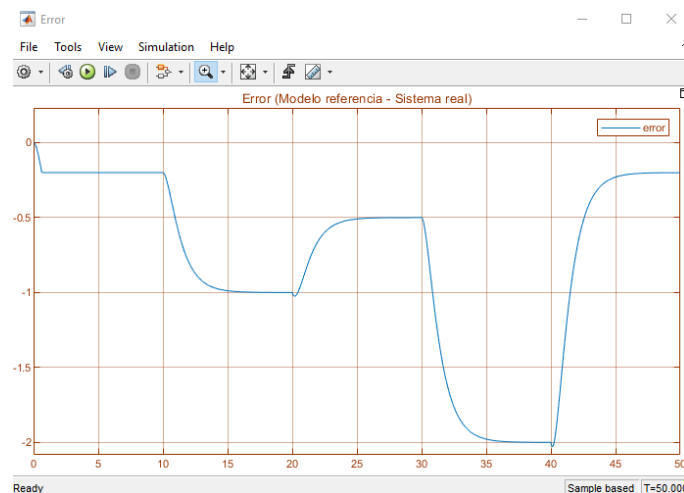


Ilustración 91 Error entre modelo de referencia masa 1 kg y controlador $P=2$ $D=3$ y sistema real con masa 1.2kg, controlador $P=2$ $D=3$

Y para cuando se tiene el controlador tiene el valor $P=1000$ y $D=250$, también se encuentra un error entre el sistema y el modelo de referencia como se muestra en la Ilustración 92, donde se tiene el sobre pico en el segundo 32, y en el segundo 42, por lo que estipula que el controlador tendría un error mayor que usando el controlador $P=1000$ y $D=1400$, aunque en el primero se tenga un tiempo de respuesta mejor.

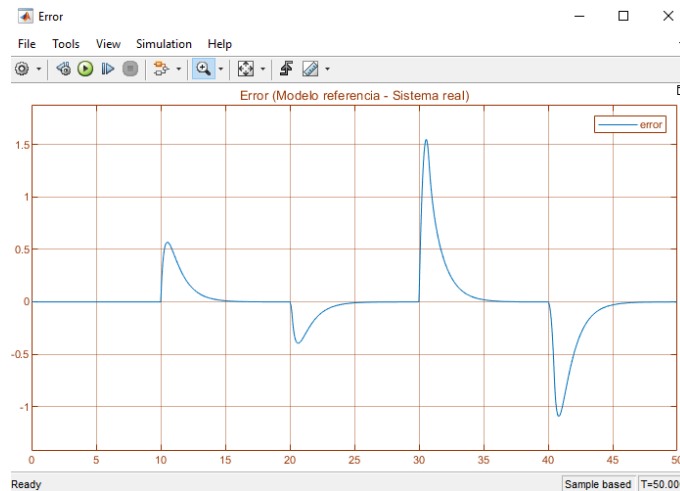
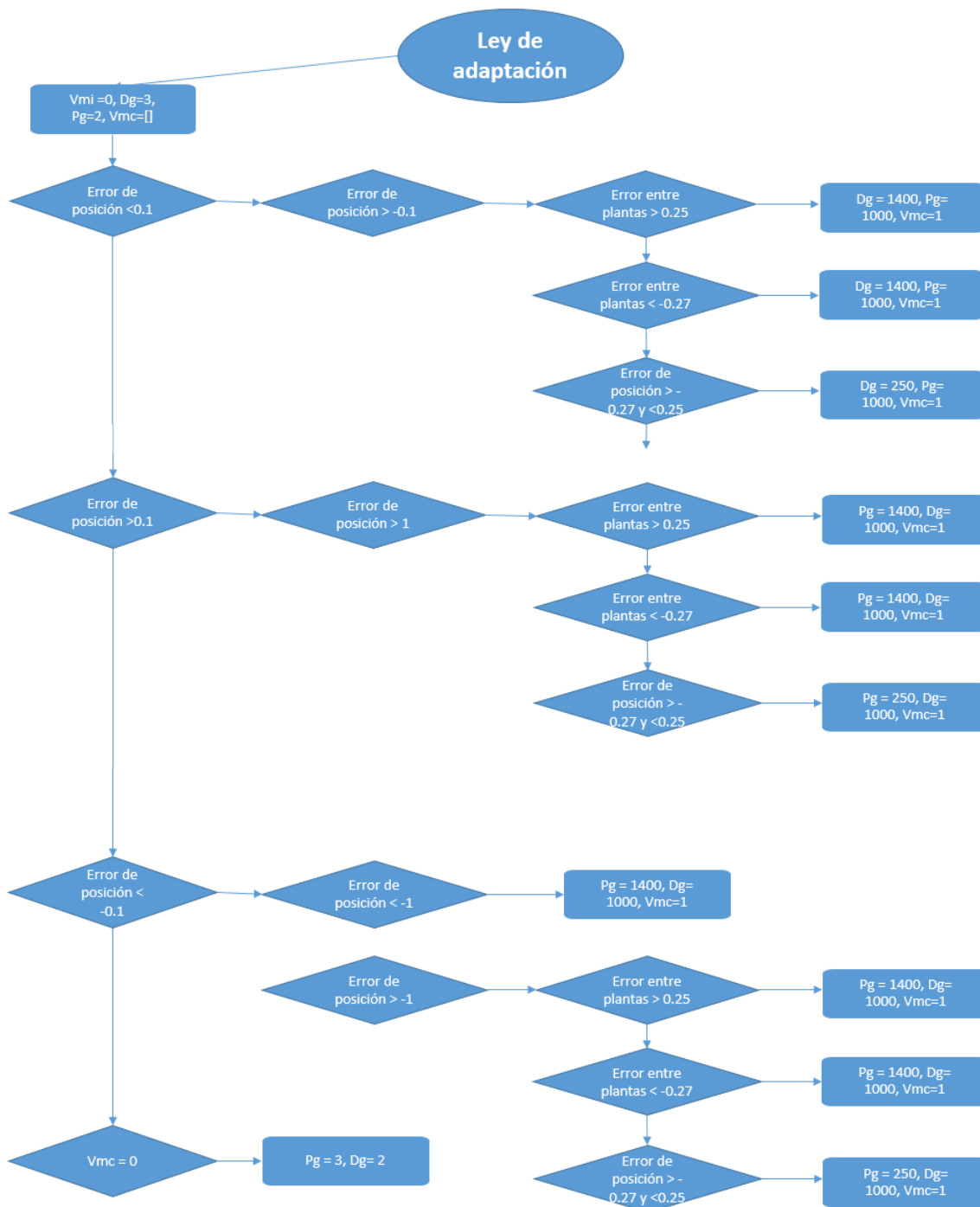


Ilustración 92 Error entre modelo de referencia masa 1 kg y controlador $P=2$ $D=3$ y sistema real con masa 1.2kg, controlador $P=1000$ $D=250$

Por lo que se procede a realizar la ley de adaptación del controlador cuando la masa varía de 1kg a 1.2kg, y también teniendo en cuenta los diferentes controladores que se pueden implementar para cuando la masa es de 1.2kg.

Queda la ley de adaptación de la siguiente manera:



4. IMPLEMENTACIÓN

En el proceso de implementación se encuentra la información referente al proceso de ensamble, factores a tener en cuenta.

4.1. Ensamble en la plataforma de 2 GDL

Primero se ubica los rodamientos en el soporte rectangular, luego se procede a ensamblar el eje de sujeción, se procede a unir este con la base móvil teniendo mucho cuidado, ya que esta no se va a poder soltar de nuevo, como se muestra en la Ilustración 93.

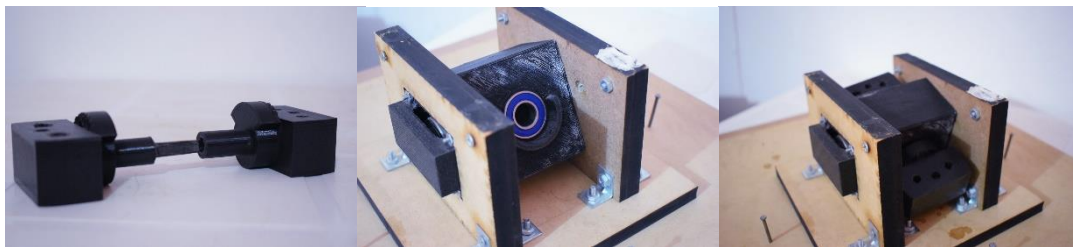


Ilustración 93 Plataforma de 2 gdl. Fuente: Autor.

Luego de ensamblar la plataforma de 2 gdl, se introducen las varillas las cuales van a hacer la transmisión de movimiento del drone a nuestra plataforma. Primero se ubica en los ejes de sujeción y se ajustan y posteriormente se ensambla al drone.

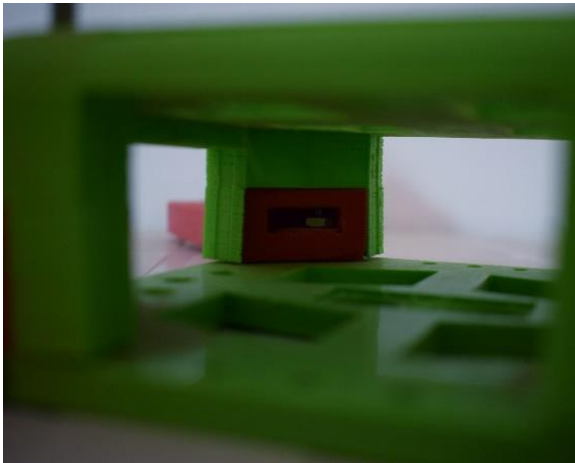


Ilustración 94 Vista interna drone. Fuente: Autor.

Para ensamblar el drone, es necesario que los brazos estén unidos compactamente, y que estén lo más fijos posible, para ello se diseñó una apertura en los brazos que permitan ajustar una tuerca como se ve en la Ilustración 94, ya que los brazos soportan los motores que son los dispositivos que general la fuerza mediante las hélices, cualquier cabeceo en su sujeción presenta un gran problema para el control del sistema.

Por su parte, la hélice también es necesario que estén colocadas centradamente en los motores, y verificar que estas no choquen con la estructura del dispositivo o cualquier dispositivo que este en él.

Por otra parte, para el ensamble de los controladores de velocidad electrónicos (ESC), se conectan por la parte de abajo y se sujetan con amarres de plástico, al igual que los cables para que no colisionen con el movimiento de las hélices, las conexiones de los cables están perfectamente aisladas mediante tubo de caucho termoencogible.



Ilustración 95 Drone. Fuente: Autor.

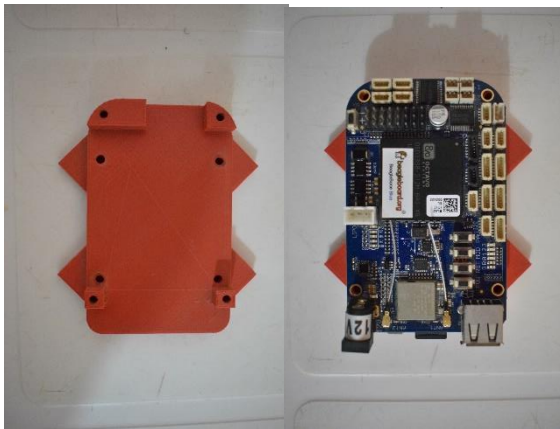


Ilustración 96 Soporte de Beaglebone Blue. Fuente: Autor.

Para la ubicación del microcontrolador es necesario diseñar e imprimir en 3d un soporte para ubicar esta tarjeta, y quede sujeta en el drone, de esta manera el ángulo medido va a corresponder al ángulo del dispositivo sin manera de generar error. Este soporte se implementará en la parte superior del dispositivo, debido a que los cables necesitan de espacio para quedar bien conectados.



Ilustración 97 Drone en plataforma de 2gdl. Fuente: Autor.

Luego de esto, ya está listo el dispositivo para realizar pruebas de control de postura.

4.1.2. Preparación del sistema

La tarjeta de control BeagleBone blue, es una microcontroladora que contiene su propia librería de código abierto, además de contar varios proyectos de software libre como [39], [13], etc. Pero como se está desarrollando un proyecto desde cero, se están implementando las librerías, y todas las librerías del dispositivo poseen su propia calibración para mejorar el rendimiento del microcontrolador.

4.1.2.1. CALIBRACIÓN DEL GIROSCOPIO

Por su parte la calibración de giroscopio es una librería que interfiere tanto en los datos de aceleración angular como de aceleración lineal, por lo que es necesario realizar este proceso con mucho cuidado y teniendo pendiente los ejes de rotación y el sistema de referencia propio del microcontrolador, el desarrollo de esta calibración se encuentra detallado en el Anexo 3.

4.1.2.2. CALIBRACIÓN DE LOS MOTORES *BRUSHLESS*

La calibración de los motores *brushless*, se basa en un pequeño proceso donde a la tarjeta de control se le especifican cuales son los motores, y ella internamente hace un proceso de calibración para detectar cuales pines son los que tienen conectado un ESC, este proceso esta detallado también en el Anexo 3, y debe realizarse cada vez que se enciende la BeagleBone blue.

4.2. Calibración de la postura del dispositivo

Luego de realizar el proceso de calibración, se procede a ubicar el dispositivo en una posición de equilibrio, sin que este ladeado hacia ningún eje, con el fin de ubicar cualquier error en la medición de postura. Este proceso es necesario realizar una vez, cada que se monte el dispositivo ante la plataforma de pruebas de 2 gdl, aunque también es bueno observar este parámetro cada que se realiza la prueba de controlador de postura, con el fin de que no tenga problemas en su ejecución.

Para realizar este proceso como se menciona anteriormente es necesario poner el sistema en equilibrio, luego se carga el sistema a la tarjeta de control, y luego de esto se analiza la respuesta del sensor de postura, tanto del ángulo pitch como de roll. De este valor se encuentra un valor medio, luego se detiene el programa y se implementa un bloque como se muestra en la Ilustración 98, que represente este mismo valor constante en signo contrario para

contrarrestar el error en la postura del sistema obteniendo una respuesta como se muestra en la Ilustración 99.

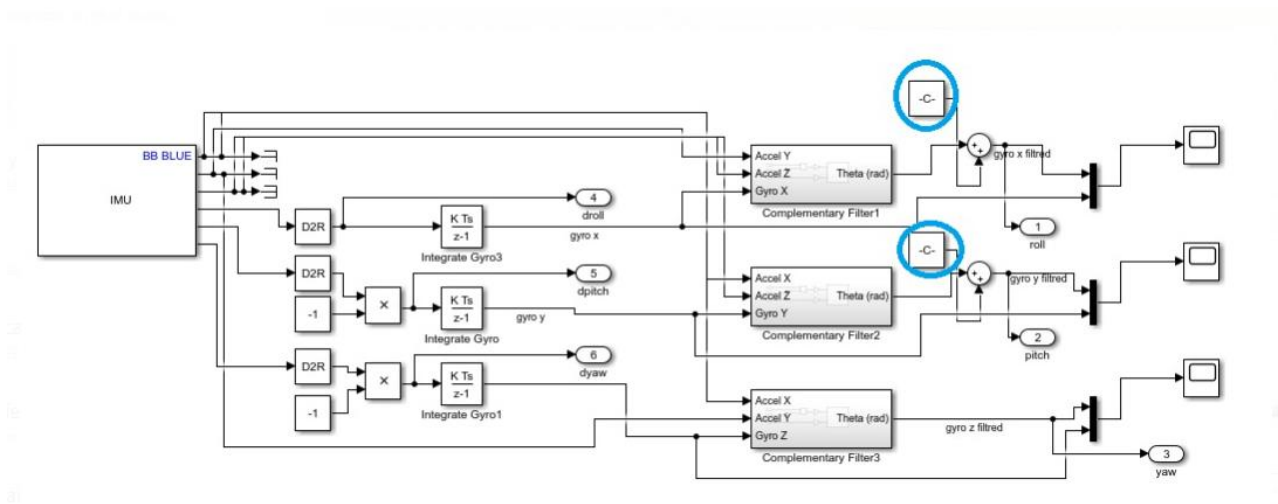


Ilustración 98 Diagrama de bloques de calibración de giroscopio.

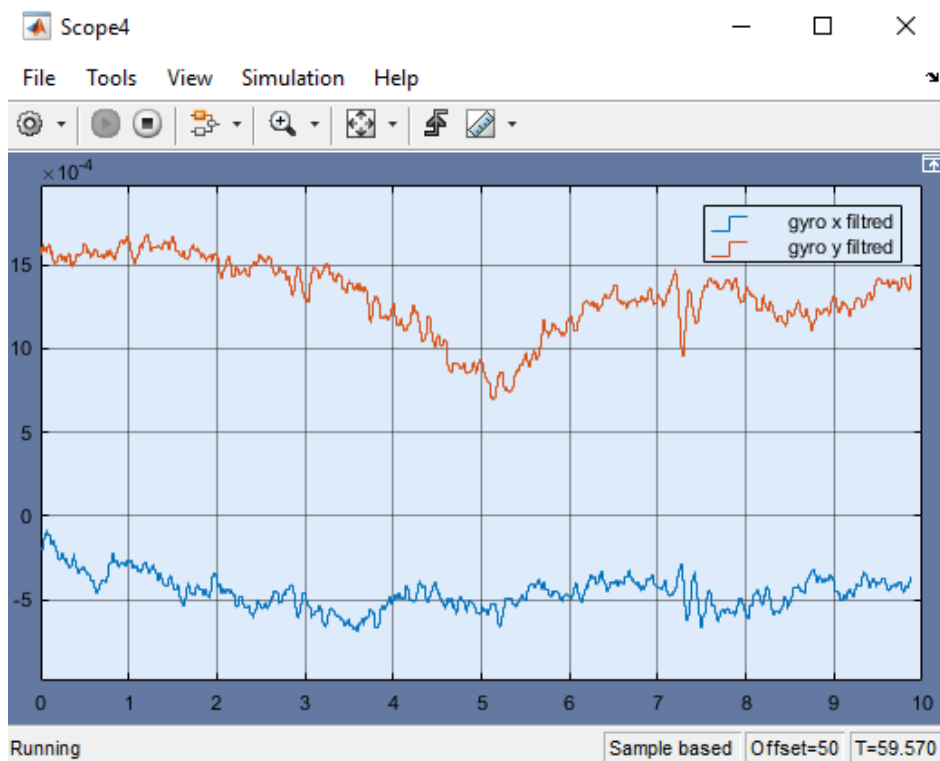


Ilustración 99 Respuesta del giroscopio de Beaglebone Blue ajustado. Fuente: Autor

5. RESULTADOS

5.1. Resultados experimentales

5.1.1. RESULTADO DE SIMULACIONES REALIZADAS CON CONTROL CLÁSICO PD

Lo primero que hay que tener en cuenta es que hay muchas maneras para realizar las pruebas del control clásico y adaptativo, pero es recomendable implementar una trayectoria de puntos, teniendo en cuenta que la mayoría de operaciones de un drone industrial se realizan siguiendo este concepto.

Por eso, se debe tener en cuenta el tiempo que se envía entre una posición y otra, analizando el tiempo de respuesta que tiene el dispositivo en simulación. Esta fase en simulación será añadida mediante un bloque de generación de trayectorias, indicando las posiciones en el entorno de simulación, en determinados momentos.

Mediante el bloque de Simulink “Signal Builder” se procede a crear las trayectorias para las posiciones y el ángulo yaw. Determinando las posiciones y el momento en el que se desee que el drone se mueva como se observa en la Ilustración 100 Ilustración 100 Señal deseada de posición para el eje "x", "y" y "z". Fuente: Autor.

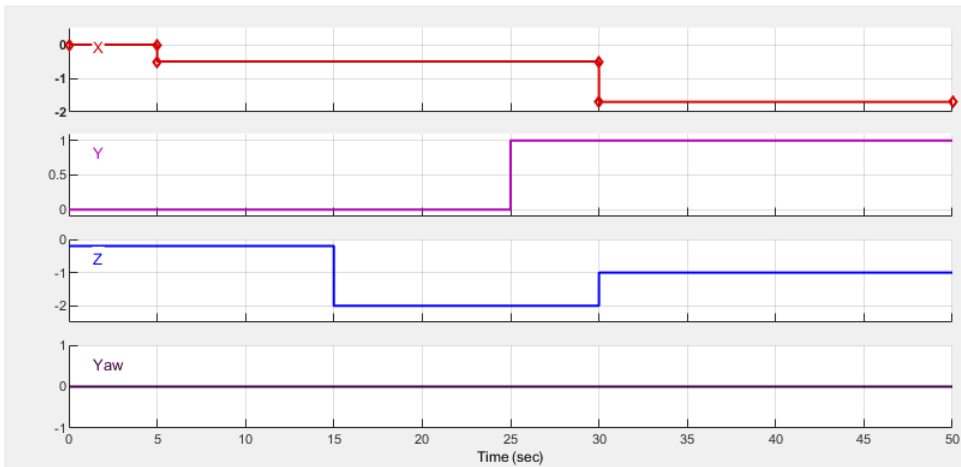


Ilustración 100
Señal deseada de posición para el eje "x", "y" y "z".
Fuente: Autor.

Por su parte, se observa en la Ilustración 83, que el modelo de control clásico PD funciona óptimamente, con tiempos de asentamiento menores a 5 segundos, y sin sobre oscilaciones. No se realiza una variación en el ángulo yaw Ilustración 38, debido a que solo se realiza una trayectoria en el dispositivo.

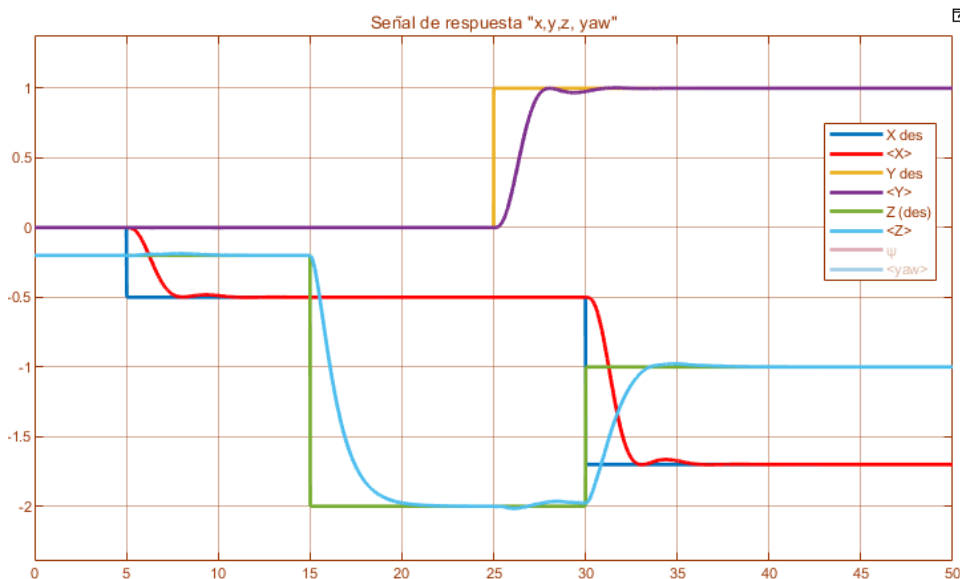


Ilustración 101 Respuesta del sistema con controlador PD heurístico. Fuente: Autor.

5.1.2. RESULTADO DE SIMULACIONES REALIZADAS CON CONTROL CLASICO Y CAMBIO DE MASA

Ahora se realiza una variación de 200 gramos en el peso del sistema, y se observa el movimiento el sistema con respecto a cada eje, teniendo en cuenta respuesta en estado estacionario y tiempo de asentamiento.

El modelo propuesto de ajuste de masa no considera las fuerzas que genera una masa suspendida con una cuerda al drone, se modela el cambio de masa en el centro de gravedad del sistema. Esto con la idea de realizar una prueba de concepto de la ley de control adaptativo

Como se observa en la Ilustración 85, el control de posición para los movimientos en los ejes "x" y "y" es aceptable, y de hecho no tiene variación con el proceso anterior. Donde se observa una gran variación en el desarrollo del controlador es en el movimiento del eje z, donde inicialmente, se observa que tiene un error en estado estacionario y lo que significa para el dispositivo es que, no está en la posición de la señal deseada, si no 20 cm más abajo, y esto es debido a que el controlador de posición de altura está definido para un peso más bajo del que tiene en ese momento, y por ello registra una posición más baja, esto permite

indicar una variación en el control del dispositivo cuando está transportando un elemento y cuando no.

Se propone implementar un control PID, para observar la respuesta del sistema cuando se agrega la constante integral en el controlador del eje z. Aplicando una señal de referencia únicamente para este eje.

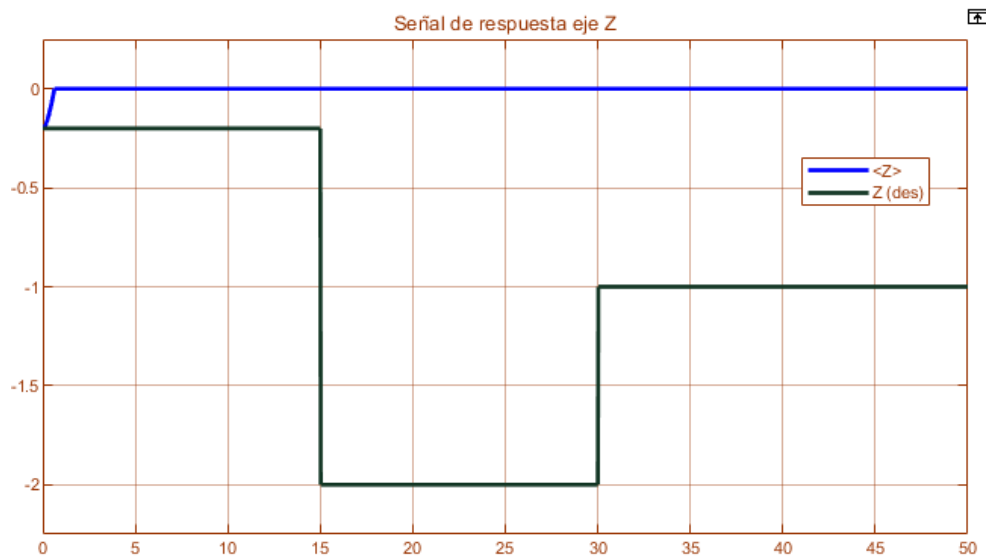


Ilustración 102 Respuesta del sistema con controlador PID heurístico y cambio en el peso del sistema. Fuente: Autor.

En la Ilustración 87, se observa que al igual que el controlador PD este sistema no responde aceptablemente para cuando la masa tiene un valor de 1.2 kg. Por lo que se intentan varios valores de la constante integral, menores y mayores de 1, teniendo en cuenta que puede dañar la estabilidad del sistema durante su desarrollo debido a oscilaciones, para lo cual se obtiene una respuesta igual a la mostrada en la Ilustración 102, lo cual propone implementar un controlador adaptativo.

Primero se debe encontrar donde se implementa el controlador adaptativo, se debe tener en cuenta que el único controlador que no funciona es el de altura, ya que, si el dispositivo se mueve hacia los lados, aunque tenga más peso, sigue haciéndolo de acuerdo aceptablemente.

Así que se encuentran los valores de control clásico PD para el sistema cuando tiene 200 gramos de más, y se hace una conmutación (o switcheo) en el momento en que el sistema cuenta con este peso.

Siempre se piensa implementar el controlador con el fin de solucionar la respuesta del sistema que ya se tiene con el controlador clásico mostrado en la Ilustración 83, teniendo en cuenta que se encontró un controlador que actúa rápidamente cuando la masa es de 1.2kg como se ve en la Ilustración 89, pero presenta el inconveniente que tiene sobre pico en la señal de respuesta, lo que afectaría al sistema si está desarrollado en entornos cerrados.

5.1.3. RESULTADO DE CONTROL ADAPTATIVO

Este sistema es viable para casos donde se conoce previamente la masa a operar y cuál es el momento y que parámetros permiten hacer este cambio, para luego hacer uso del control adaptativo con una regla de adaptación de switcheo.

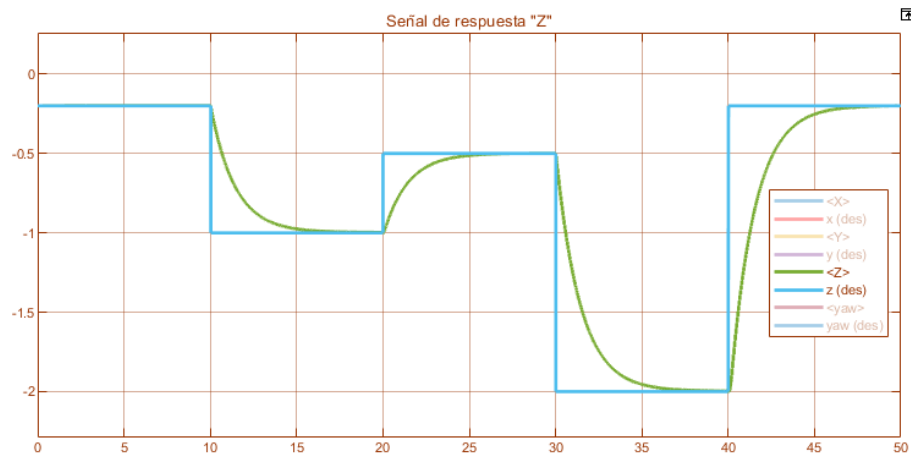


Ilustración 103 Respuesta del sistema con controlador clásico $P=1000$, $D=1400$ y masa constante = 1.2kg. Fuente Autor

Como se observa en la Ilustración 103 la señal de respuesta es aceptable con un tiempo de asentamiento de alrededor de 5s y sin sobre picos en cada nueva señal de referencia. Teniendo en cuenta que la respuesta del sistema es sin variación de la masa, únicamente ajustando el valor de la nueva masa del sistema.

Ahora se procede a realizar las pruebas con la ley de adaptación y el modelo de referencia realizados en la sección 0 ejerciendo una variación en la masa al segundo 5, y volviendo a quitársela al segundo 18, teniendo en cuenta que en estos dos momentos, en el sistema no se genera ningún otro error en posición y también su respuesta con la señal de referencia tiende a estabilizarse en un error muy cercano a 0.

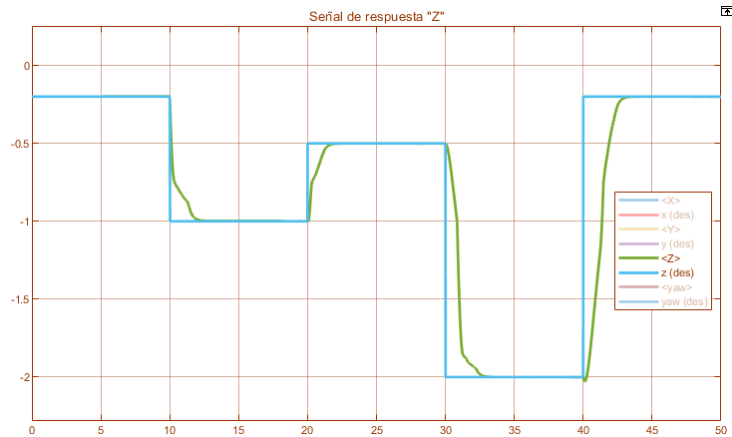


Ilustración 104 Señal de respuesta con masa al segundo 0 = 1kg, masa al segundo 5=1.2kg y masa al segundo 18=1kg. Fuente: Autor.

En la Ilustración 104, se observa que la respuesta del sistema es óptima, por otra parte también se analiza en que momento la ley de adaptación realiza el ajuste del controlador en la Ilustración 105, teniendo en cuenta que realiza varios ajustes en el controlador debido a los parámetros con los cuales se estableció la ley de adaptación. Cabe recalcar que muchas veces esta señal baja a el controlador $P=2$ y $D=3$, debido a que el error entre la planta y el modelo de referencia es muy cercano a 0, lo cual tampoco afecta el desarrollo del sistema, únicamente ocurre un switcheo entre este controlador y el controlador $P=1000$ y $D=1400$ en bastantes ocasiones en poco tiempo, cuando el sistema tiende a estabilizarse.

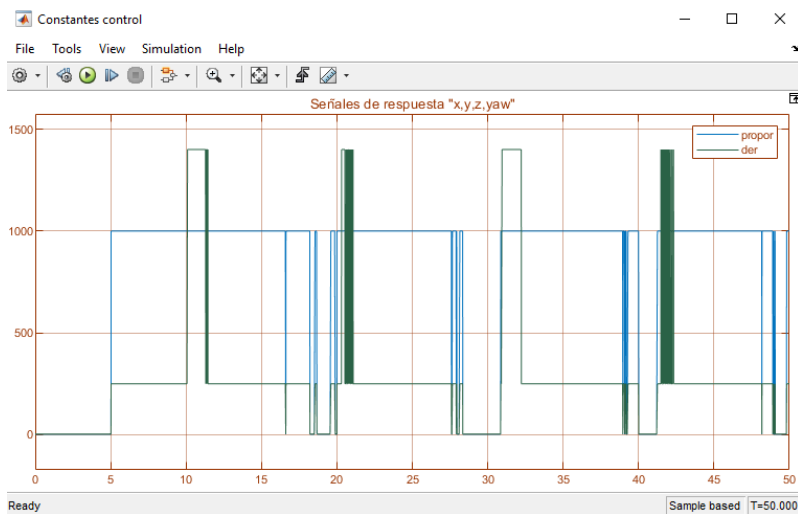


Ilustración 105 Controlador PD para ajuste en la masa al segundo 5 y 18. Fuente: Autor

Ahora se realiza el mismo procedimiento cambiando el momento en que se quita la masa, quitándola ahora en el segundo 27, y se observa la respuesta de la ley de adaptación en la Ilustración 106, lo que muestra una gran semejanza entre este y la señal generada por la ley de adaptación de la Ilustración 105.

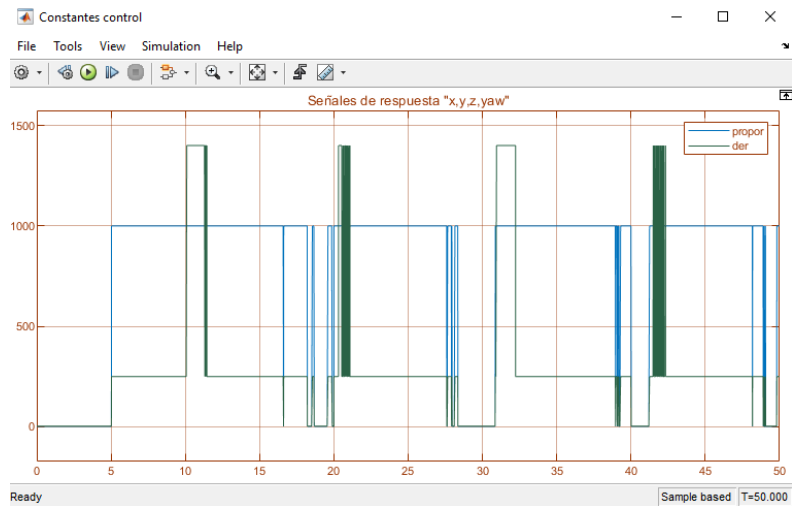


Ilustración 106 Controlador PD para ajuste en la masa al segundo 5 y 27.
Fuente: Autor.

De esta manera la señal de respuesta del sistema correspondiente a este ajuste en la masa al segundo 27 se muestra en la Ilustración 107, en la cual tampoco se observa una gran variación entre este y el modelo de respuesta del sistema anterior mostrado en la Ilustración 104.

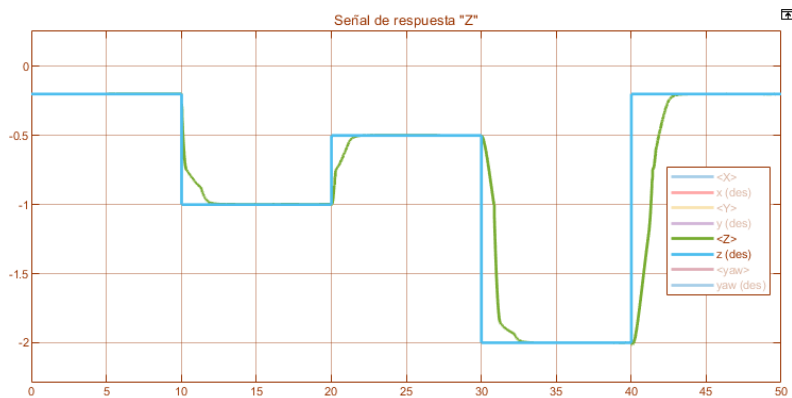


Ilustración 107 Respuesta del sistema para el controlador adaptativo con ajuste en la masa en el segundo 5 y 27. Fuente: Autor.

6. CONCLUSION Y DISCUSION

- Se realizó el estudio de diseño del sistema, considerando el dimensionamiento de las piezas para el prototipo, como es su desempeño y como afectan en el tiempo de operación.
- Se determinó cuáles son las piezas necesarias de acuerdo a las necesidades del proyecto, resultando: un motor A2212 1000KV 13t, un ESC 30A, una BeagleBone Blue, una batería de 1500 mah, y unas hélices 8045.
- Luego de realizar el diseño de las piezas, se realiza el análisis de elementos finitos para cada una de las piezas, del cual genera resultados de diseño óptimo para proceder a realizar su respectivo proceso de fabricación.
- Se desarrolló el modelo dinámico del sistema, gracias al curso de Aerial Robotics de Coursera, dado por la universidad de Pensilvania, donde se tiene el conocimiento de las ecuaciones de control y como realizar el ajuste de las constantes de control. Además de aprender conceptos relevantes para el diseño del dispositivo, como la importancia de la distancia de los brazos del drone, ya que esta entre más grande, mas aumenta la aceleración angular del dispositivo, o el coeficiente de thrust que es el empuje real que ejerce un motor de acuerdo a la cantidad de revoluciones que realiza.
- Se desarrolló e implementó el controlador adaptativo en simulación ajustando los parámetros de control mediante una ley de adaptación de condicionales debido al error entre el modelo real y la planta, y también el error entre la posición deseada y la posición actual del dispositivo. Este control adaptativo hace el switcheo entre el controlador $P=2$ y $D=3$, que es el de la masa de 1kg, y el controlador $P=1000$ y $D=1400$ para la masa de 1.2kg, haciendo un pequeño switcheo en estado transitorio al controlador $P=1000$ y $D=250$, generando una respuesta aceptable del sistema con tiempos de asentamiento de 3 segundos.
- En el desarrollo de controlador adaptativo se trata de implementar el controlador PID para observar si este responde a la señal de error que genera el cambio de masa, del cual se obtiene como respuesta que no responde a este error, y por lo tanto no funcionaria.

BIBLIOGRAFIA

- [1] “Robotics: Aerial Robotics | Coursera.” <https://www.coursera.org/learn/robotics-flight> (accessed Jun. 09, 2020).
- [2] P. Mahony, R. and Kumar, V. and Corke, “Modeling, Estimation, and Control of Quadrotor,” *IEEE Robot. Autom. Mag.*, vol. 19, no. SEPTEMBER, pp. 20–32, 2012.
- [3] I. Palunko and R. Fierro, *Adaptive control of a quadrotor with dynamic changes in the center of gravity*, vol. 18, no. PART 1. IFAC, 2011.
- [4] P. Corke, “Robotics *Toolbox* | Peter Corke,” 2017. <https://petercorke.com/toolboxes/robotics-toolbox/> (accessed Jun. 09, 2020).
- [5] “Simulink Coder Support Package for BeagleBone Blue Hardware Documentation.” <https://www.mathworks.com/help/supportpkg/beagleboneblue/index.html> (accessed Jun. 09, 2020).
- [6] “Atascado y estresado: cómo afecta el tráfico a tu salud - The New York Times.” <https://www.nytimes.com/es/2019/02/07/espanol/consecuencias-traffic-salud.html> (accessed Jun. 09, 2020).
- [7] B. Custers Editor, “The Future of Drone Use Opportunities and Threats from Ethical and Legal Perspectives.” [Online]. Available: <http://www.springer.com/series/8857>.
- [8] “Así funciona el taxi volador que Uber y Hyundai presentaron en el CES - YouTube.” <https://www.youtube.com/watch?v=z1YJtYPN4EY> (accessed Jun. 09, 2020).
- [9] J. Ackermann, “Brief Paper Robustness Against Sensor Failures*,” 1984.
- [10] P. Peining, G. Tan, A. Aung, and P. Wai, “Evaluation of Consumer-Grade EEG Headsets for BCI Drone Control.” [Online]. Available: <https://www.bitcraze.io/crazyflie-2/>.
- [11] Andrés Felipe Silva Bohórquez, César Augusto Peña Cortés, and Luis Enrique Mendoza, “Sistema de inspección y vigilancia utilizando un robot aéreo guiado mediante visión artificial.”

- [12] V. Arias, J. Rodríguez, G. Chacón, M. Martínez, and O. Duque, “Navegación Autónoma Para Robots Voladores Parte 1: Dinámica Y Control De Un Quadrotor.” [Online]. Available: <https://orcid.org/0000-0001-6246-604X>.
- [13] “GitHub - dch33/Quad-Sim: A package of documentation and software supporting MATLAB/Simulink based dynamic modeling and simulation of quadcopter vehicles for control system design.” <https://github.com/dch33/Quad-Sim> (accessed Jun. 09, 2020).
- [14] P. Corke, “petercorke.com/wordpress/toolboxes/,” *Robotics Toolbox*, 2018. .
- [15] J. ee--, K. Sreenath, and V. Kumar, “Robotics: Science and Systems Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots.”
- [16] M. David and B. Robles, “SISTEMAS INTELIGENTES APLICADOS A LA AGRICULTURA Pasantía de investigación.”
- [17] A. Sabo, B. Kuljic, and S. Tibor, “Practical application of the drone technology in civil engineering,” no. March, 2019.
- [18] R. McAlinden, E. Suma, T. Grechkin, and M. Enloe, “Procedural Reconstruction of Simulation Terrain Using Drones,” *Proc. Interservice/Industry Training, Simulation, Educ. Conf.*, no. 15041, pp. 1–12, 2015.
- [19] K. J. Bathe, *Finite Element Procedures*. 1996.
- [20] “Design your own custom drone | MIT News.” <http://news.mit.edu/2016/design-your-own-custom-drone-1205> (accessed Jun. 09, 2020).
- [21] “A smaller Quad the DJI Flamewheel F330 with an APM. - Blog 2.0 - diydrones.” <https://diydrones.com/profiles/blogs/a-smaller-quad-the-dji-flamewheel-f330-with-an-apm> (accessed Jun. 09, 2020).
- [22] “(438) Pinterest.” <https://www.pinterest.es/dwadziescia3/drone-design/> (accessed Jun. 09, 2020).
- [23] Kunyue Su(B) and Shaojie Shen, *Catching a Flying Ball with a Vision-Based*

Quadrotor, vol. 1. Cham: Springer International Publishing, 2017.

- [24] F. Andres, B. Ramirez, J. Wilmer, and G. Melgarejo, “Modelización, identificación y control del 3 DOF Hover de Quanser,” 2019.
- [25] S. (Sefer) Kurnaz and Institute of Electrical and Electronics Engineers., *RAST 2009 : proceedings of 4th International Conference on Recent Advances : June 11-13, 2009, Istanbul, Turkey*. IEEE, 2009.
- [26] X. Liang, G. Chen, J. Wang, Z. Bi, and P. Sun, “An adaptive control system for variable mass quad-rotor UAV involved in rescue missions,” *Int. J. Simul. Syst. Sci. Technol.*, vol. 17, no. 29, pp. 22.1-22.7, 2016, doi: 10.5013/IJSSST.a.17.29.22.
- [27] L. T. A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, “Mathematical modeling and control of a hexacopter.”
- [28] A. P. Erasmus and H. W. Jordaan, “Linear quadratic Gaussian control of a quadrotor with an unknown suspended payload,” *2020 Int. SAUPEC/RobMech/PRASA Conf. SAUPEC/RobMech/PRASA 2020*, 2020, doi: 10.1109/SAUPEC/RobMech/PRASA48453.2020.9040992.
- [29] “Laurens Van Hoyer Design of a control framework for industrial drone assistance,” 2016.
- [30] M. M. Ferdous, M. Pratama, S. G. Anavatti, M. A. Garratt, and E. Lughofer, “PAC: A novel self-adaptive neuro-fuzzy controller for micro aerial vehicles,” *Inf. Sci. (Ny)*, vol. 512, no. xxxx, pp. 481–505, 2020, doi: 10.1016/j.ins.2019.10.001.
- [31] C. H. Pi, K. C. Hu, S. Cheng, and I. C. Wu, “Low-level autonomous control and tracking of quadrotor using reinforcement learning,” *Control Eng. Pract.*, vol. 95, no. April 2019, p. 104222, 2020, doi: 10.1016/j.conengprac.2019.104222.
- [32] I. and M. J. R. J. P. Ortiz, Member, IEEE, L. I. Minchala, Member, “Nonlinear Robust H-Infinity PID Controller for the Multivariable System Quadrotor.”
- [33] F. A. Goodarzi and T. Lee, “Dynamics and control of quadrotor UAVs transporting a rigid body connected via flexible cables,” *Proc. Am. Control Conf.*, vol. 2015-July,

pp. 4677–4682, 2015, doi: 10.1109/ACC.2015.7172066.

- [34] J. Geng and J. W. Langelaan, “Implementation and demonstration of coordinated transport of a slung load by a team of rotorcraft,” *AIAA Scitech 2019 Forum*, no. January, pp. 1–19, 2019, doi: 10.2514/6.2019-0913.
- [35] M. *Brushless*, E. Alcaide, D. Cabañez, and M. Castro, “MÁQUINAS ELÉCTRICAS.”
- [36] P. Anand, P. Arjun, N. Bharath Kumar, K. Gowtham, and A. Professor, “Drone Ambulance Support System,” *Int. J. Eng. Tech.*, vol. 4, no. 2, pp. 369–373, 2018, [Online]. Available: <http://www.ijetjournal.org>.
- [37] “Propiedades del PLA.” <https://sites.google.com/view/poliacidolactico-coma/políacido-láctico/propiedades-del-pla> (accessed Jun. 09, 2020).
- [38] F. Adamo *et al.*, “Designing and prototyping a sensors head for test and certification of UAV components,” *Int. J. Smart Sens. Intell. Syst.*, vol. 10, no. 3, pp. 646–672, 2017, doi: 10.21307/ijssis-2017-228.
- [39] “Matlab_MiP/beta-IMU at master · ktalke12/Matlab_MiP · GitHub.” https://github.com/ktalke12/Matlab_MiP/tree/master/beta-IMU (accessed Jun. 09, 2020).
- [40] “BeagleBone Blue | Getting Started - YouTube.” https://www.youtube.com/watch?v=pTpUMajQS_U (accessed Jun. 09, 2020).

ANEXOS

1. IMPRESIÓN 3D

Antes de realizar el proceso de impresión 3d, se necesitó especificar los valores mecánicos de la pieza para el análisis de elementos finitos desarrollados en la sección 0. En el software fusión 360, se realizó el análisis de elementos finitos, especificando las propiedades físicas de los elementos probados como se muestra en la Ilustración 108.

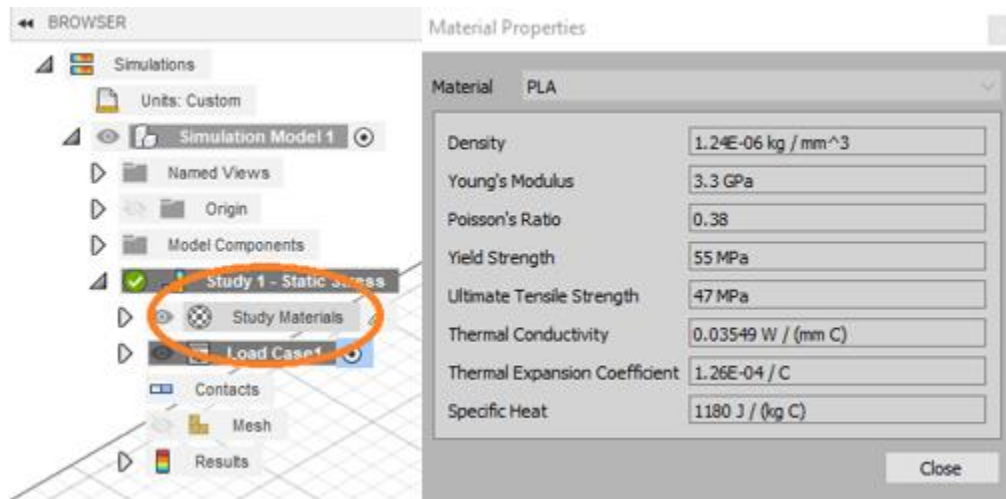


Ilustración 108 Ajuste de la propiedad de materiales en el programa Fusión 360. Fuente: Autor.

Luego de relacionar todas las piezas con sus respectivas propiedades físicas, y de obtener como resultado que las piezas están diseñadas aceptablemente en el estudio de elementos finitos, se proceden a fabricar en impresión 3d como se observa en la Ilustración 109 .

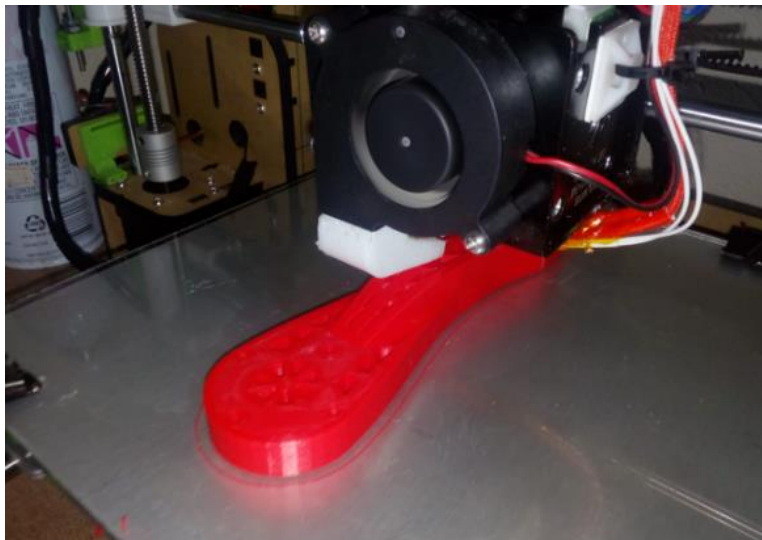


Ilustración 109 Impresión 3D, del brazo del quadcoptero. Fuente: Autor.

2. DATOS DE LOS MOTORES

Los datos de los motores, es información relevante para la preselección de motores de acuerdo a su eficiencia, empuje, y consumo de corriente, por lo que se investiga la información de varios motores:

1. A 2212 1000kv

Se tiene un consume de corriente bajo a un empuje aceptable, resultando el consumo de 4A para cuando el motor realiza un empuje de 2.81N, y de 6.5A para cuando el motor realiza un empuje de 4.85N.

Fuente: <https://www.rcgroups.com/forums/showthread.php?889075-Propeller-sizes-for-Suppo-2212-13-and-2212-10>

2. Referencia readtosty MT

Para este tipo de motor se usan mucho las hélices pequeñas en las que se realiza un alto consumo de corriente y generan pequeño empuje, descartando este motor.

Fuente: <https://www.youtube.com/watch?v=fJVHBurswZM>

3. Emax xa 2212

Este tipo de motor presenta una gran cantidad de empuje a un costo aceptable de corriente, ejerciendo por ejemplo un empuje de 4.9N por 9.1A y de 5.1N por 11.9A por lo que es viable poder implementar en este tipo de proyectos con la gama de 1400KV.

Fuente: https://www.banggood.com/Emax-XA2212-820KV-980KV-1400KV-Brushless-Motor-For-RC-Models-p-918124.html?ID=47984&cur_warehouse=CN

4. Racerstar 2205 2300KV

Al igual que en el readtosty MT este motor es usado comúnmente con pequeñas hélices, aunque este presenta una mejor relación entre consumo energético y empuje realizado, ya que realiza 3.4N por 6.22A por lo que puede ser viable para implementar en diseños de drones mucho más pequeños.

Fuente: <https://www.youtube.com/watch?v=bqVOoBJCOS4>

3. AJUSTE PARAMETROS SIMULINK

Para el desarrollo de la simulación de nuestro sistema es necesario tener instalada la *toolbox* de robótica desarrollada por Peter Corke [14], esta *toolbox* se encuentra en su página web petercorke.com, llamada “Robotics *Toolbox*”, en la página se encuentra una guía desarrollada donde se muestran tres ejemplos de cómo se puede instalar esta *toolbox* en Matlab.

Luego de realizar los pasos es necesario correr el programa básico desarrollado en Simulink del cuadricoptero. El cual se encuentra en la siguiente dirección: “C:\Users\yedui\Documents\MATLAB\Add-Ons\Toolboxes\Robotics Toolbox for MATLAB\simulink” el archivo tendrá el siguiente nombre “sl_quadrotor”. Es necesario correr este archivo para comprobar de que la *toolbox* está bien instalada, luego de que la simulación funciona correctamente es momento de empezar a modificar los valores mecánicos que tiene el sistema con los datos mecánicos del dispositivo que se están desarrollando.

Para acceder a la información se debe acceder al archivo mdl_quadrotor encontrado en la siguiente dirección: C:\Users\yedui\Documents\MATLAB\Add-Ons\Toolboxes\Robotics Toolbox for MATLAB\robot\models

Se encuentra el siguiente archivo donde se encuentra toda la información dinámica del dispositivo:

```
%MDL_QUADCOPTER Dynamic parameters for a quadrotor.
%
% MDL_QUADCOPTER is a script creates the workspace variable quad which
% describes the dynamic characteristics of a quadrotor flying robot.
%
% Properties:
%
% This is a structure with the following elements:
%
% nrotors    Number of rotors (1x1)
% J          Flyer rotational inertia matrix (3x3)
% h          Height of rotors above CoG (1x1)
% d          Length of flyer arms (1x1)
% nb         Number of blades per rotor (1x1)
% r          Rotor radius (1x1)
% c          Blade chord (1x1)
% e          Flapping hinge offset (1x1)
% Mb         Rotor blade mass (1x1)
% Mc         Estimated hub clamp mass (1x1)
% ec         Blade root clamp displacement (1x1)
% Ib         Rotor blade rotational inertia (1x1)
```



```

% Ic      Estimated root clamp inertia (1x1)
% mb      Static blade moment (1x1)
% Ir      Total rotor inertia (1x1)
% Ct      Non-dim. thrust coefficient (1x1)
% Cq      Non-dim. torque coefficient (1x1)
% sigma   Rotor solidity ratio (1x1)
% thetat  Blade tip angle (1x1)
% theta0  Blade root angle (1x1)
% thetai  Blade twist angle (1x1)
% theta75 3/4 blade angle (1x1)
% thetai  Blade ideal root approximation (1x1)
% a       Lift slope gradient (1x1)
% A       Rotor disc area (1x1)
% gamma   Lock number (1x1)
%
%
% Notes::
% - SI units are used.
%
% References::
% - Design, Construction and Control of a Large Quadrotor micro air
vehicle.
%   P.Pounds, PhD thesis,
%   Australian National University, 2007.
%   http://www.eng.yale.edu/pep5/P\_Pounds\_Thesis\_2008.pdf
% - This is a heavy lift quadrotor
%
% See also sl_quadrotor.

% Copyright (C) 1993-2017, by Peter I. Corke
%
% This file is part of The Robotics Toolbox for MATLAB (RTB).
%
% RTB is free software: you can redistribute it and/or modify
% it under the terms of the GNU Lesser General Public License as
published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% RTB is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU Lesser General Public License for more details.
%
% You should have received a copy of the GNU Lesser General Public License
% along with RTB. If not, see <http://www.gnu.org/licenses/>.
%
% http://www.petercorke.com

quadrotor.nrotors = 4;          %      %      4 rotors
quadrotor.g = 9.81;           %      %      g          Gravity
1x1
quadrotor.rho = 1.2294555;    %      %      rho        Density of air
1x1

```

```

quadrotor.muv = 1.5e-5;          %   %   muv   Viscosity of air
1x1

% Airframe
quadrotor.M = 1;                %   %   M     Mass
1x1
% Ixx = 0.082;
% Iyy = 0.082;
% Izz = 0.149;%0.160;
quadrotor.J = [ 0.015401548, 0.00009343, -0.000002472;...
                0.000093473, 0.015373703, 0.000008096;...
                -0.000002472, 0.000008096, 0.02996882];
% quadrotor.J = diag([0.01 0.01 0.01]); %%   %   I     Flyer
rotational inertia matrix      3x3

quadrotor.h = -0.03;            %   %   h     Height of rotors
above CoG                       1x1
quadrotor.d = 0.1684;          %   %   d     Length of flyer arms
1x1

%Rotor
quadrotor.nb = 2;              %   %   b     Number of blades per
rotor                             1x1
quadrotor.r = 0.127;          %   %   r     Rotor radius
1x1

quadrotor.c = 0.099;          %   %   c     Blade chord
1x1

quadrotor.e = 0.0;            %   %   e     Flapping hinge offset
1x1
quadrotor.Mb = 0.0056;        %   %   Mb    Rotor blade mass
1x1
quadrotor.Mc = 0.010;         %   %   Mc    Estimated hub clamp
mass                             1x1
quadrotor.ec = 0.004;         %   %   ec    Blade root clamp
displacement                     1x1
quadrotor.Ib = quadrotor.Mb*(quadrotor.r-quadrotor.ec)^2/4 ;          %
Ib      Rotor blade rotational inertia      1x1
quadrotor.Ic = quadrotor.Mc*(quadrotor.ec)^2/4;                      %   Ic
Estimated root clamp inertia      1x1
quadrotor.mb =
quadrotor.g*(quadrotor.Mc*quadrotor.ec/2+quadrotor.Mb*quadrotor.r/2);
%   mb      Static blade moment              1x1
quadrotor.Ir = quadrotor.nb*(quadrotor.Ib+quadrotor.Ic);            %
Ir      Total rotor inertia                  1x1

quadrotor.Ct = 0.0035672832;   %   %   Ct    Non-dim. thrust
coefficient                       1x1
% quadrotor.Ct = 0.00118729;   %   %   Ct    Non-dim. thrust
coefficient                       1x1
quadrotor.Cq = quadrotor.Ct*sqrt(quadrotor.Ct/2);                  %   Cq
Non-dim. torque coefficient      1x1

quadrotor.sigma = quadrotor.c*quadrotor.nb/(pi*quadrotor.r);      %
sigma      Rotor solidity ratio              1x1

```

```

quadrotor.thetat = 6.8*(pi/180);           % thetat Blade tip angle
1x1 %sin cambiar
quadrotor.theta0 = 14.6*(pi/180);         % theta0 Blade root angle
1x1 %sin cambiar
quadrotor.theta1 = quadrotor.thetat - quadrotor.theta0; % theta1
Blade twist angle 1x1
quadrotor.theta75 = quadrotor.theta0 + 0.75*quadrotor.theta1;% theta76
3/4 blade angle 1x1
quadrotor.thetai = quadrotor.thetat*(quadrotor.r/quadrotor.e); %
thetai Blade ideal root approximation 1x1
quadrotor.a = 5.5;                         % a Lift slope gradient
1x1 %sin cambiar

% derived constants
quadrotor.A = pi*quadrotor.r^2;             % A Rotor disc
area 1x1
quadrotor.gamma =
quadrotor.rho*quadrotor.a*quadrotor.c*quadrotor.r^4/(quadrotor.Ib+quadrot
or.Ic);% gamma Lock number 1x1

quadrotor.b = quadrotor.Ct*quadrotor.rho*quadrotor.A*quadrotor.r^2; % T =
b w^2
quadrotor.k = quadrotor.Cq*quadrotor.rho*quadrotor.A*quadrotor.r^3; % Q =
k w^2

quadrotor.verbose = false;
quadrotor.J

```

Cada una de las variables representa una magnitud física, la intención es cambiar estos valores ajustándoles a los valores correspondientes a los de nuestro dispositivo. Pero hay algunos archivos de los que no se tiene conocimiento su valor, por lo que se dejan los mismos con los que cuenta el programa.

Los datos que se van a ajustar son;

```
quadrotor.M = 1;
```

Masa del dispositivo

```
quadrotor.d = 0.1684;
```

Tamaño de las hélices

```
quadrotor.J = quadrotor.J = [ 0.015401548, 0.00009343, -0.000002472;...
                                0.000093473, 0.015373703, 0.000008096;...
                                -0.000002472, 0.000008096, 0.02996882];
```

Matriz de inercia

```
quadrotor.Mb = 0.0056;
```

Masa de las hélices

```
quadrotor.Ct = 0.0035672832
```

Coefficiente de thrust

Este coeficiente de thrust hace referencia a la relación que tiene la velocidad de los motores con el empuje que puede ejercer, gracias a un tacómetro prestado por parte de la Universidad se puede realizar esta medición y hacer una aproximación de este valor.

$$b_i = \frac{T_i}{\omega_i} \quad (44)$$

```
quadrotor.rho = 1.2294555;
```

Densidad del aire

Los datos como la viscosidad del aire, y otros datos importantes acerca de la hélice implementada, se dejan los valores igual a los que contiene el programa debido a que se desconocen los valores en nuestro dispositivo.

Este archivo se lee al iniciar la simulación en el dispositivo, ajustándole en “model properties – Callbacks – InitFcn”, pero para cambiar el valor de la masa en cierto momento, es necesario modificar otro archivo, en el cual se desarrolla el modelo dinámico. Este archivo se encuentra en la siguiente dirección C:\Users\yedui\Documents\MATLAB\Add-Ons\Toolboxes\Robotics Toolbox for MATLAB\simulink el archivo se llama quadrotor_dynamics donde se analiza la función

```
function sys = mdlDerivatives(t,x,u, quad)
```

En ella se encuentra la sección de dinámica de cuerpo rígido la siguiente ecuación

```
dv = quad.g*e3 + R*(1/quad.M)*sum(T,2);
```

Se tiene q hacer que el valor “quad.M” sea variable, para que se pueda realizar el control adaptativo, ya que modificando este archivo se modifica el valor de la masa en la planta. En nuestro caso se procede a utilizar el tiempo, como el método de switcheo para cambiar el valor de la masa en determinados momentos de la simulación, quedando de la siguiente manera.

```

function [sys,x0,str,ts] = quadrotor_dynamics_masa(t,x,u,flag, quad, x0,
groundflag)
    % Flyer2dynamics lovingly coded by Paul Pounds, first coded 12/4/04
    % A simulation of idealised X-4 Flyer II flight dynamics.
    % version 2.0 2005 modified to be compatible with latest version of
Matlab
    % version 3.0 2006 fixed rotation matrix problem
    % version 4.0 4/2/10, fixed rotor flapping rotation matrix bug,
mirroring
    % version 5.0 8/8/11, simplified and restructured
    % version 6.0 25/10/13, fixed rotation matrix/inverse wronskian
definitions, flapping cross-product bug

warning off MATLAB:divideByZero

global groundFlag;

% New in version 2:
% - Generalised rotor thrust model
% - Rotor flapping model
% - Frame aerodynamic drag model
% - Frame aerodynamic surfaces model
% - Internal motor model
% - Much coolage

% Version 1.3
% - Rigid body dynamic model
% - Rotor gyroscopic model
% - External motor model

%ARGUMENTS
%   u      Reference inputs          1x4
%   tele   Enable telemetry (1 or 0) 1x1
%   crash  Enable crash detection (1 or 0) 1x1
%   init   Initial conditions        1x12

%INPUTS
%   u = [N S E W]
%   NSEW motor commands              1x4

%CONTINUOUS STATES
%   z      Position                  3x1   (x,y,z)
%   v      Velocity                  3x1   (xd,yd,zd)
%   n      Attitude                  3x1   (Y,P,R)
%   o      Angular velocity          3x1   (wx,wy,wz)
%   w      Rotor angular velocity    4x1
%
% Notes: z-axis downward so altitude is -z(3)

%CONTINUOUS STATE MATRIX MAPPING
%   x = [z1 z2 z3 n1 n2 n3 z1 z2 z3 o1 o2 o3 w1 w2 w3 w4]

%INITIAL CONDITIONS
n0 = [0 0 0];          %   n0      Ang. position initial
conditions            1x3

```

```

    v0 = [0 0 0];           % v0      Velocity Initial conditions
1x3
    o0 = [0 0 0];         % o0      Ang. velocity initial
conditions      1x3
    init = [x0 n0 v0 o0]; % x0 is the passed initial position 1x3
    groundFlag = groundflag;

%CONTINUOUS STATE EQUATIONS
% z` = v
% v` = g*e3 - (1/m)*T*R*e3
% I*o` = -o X I*o + G + torq
% R = f(n)
% n` = inv(W)*o

% Dispatch the flag.
%
switch flag
    case 0
        [sys,x0,str,ts]=mdlInitializeSizes(init, quad); %
Initialization
    case 1
        sys = mdlDerivatives(t,x,u, quad); % Calculate derivatives
    case 3
        sys = mdlOutputs(t,x, quad); % Calculate outputs
    case { 2, 4, 9 } % Unused flags
        sys = [];
    otherwise
        error(['Unhandled flag = ',num2str(flag)]); % Error handling
end
end % End of flyer2dynamics

%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the
% S-function.
%=====
%
function [sys,x0,str,ts] = mdlInitializeSizes(init, quad)
%
% Call simsizes for a sizes structure, fill it in and convert it
% to a sizes array.
%
sizes = simsizes;
sizes.NumContStates = 12;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 12;
sizes.NumInputs = 4;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
%
% Initialize the initial conditions.
x0 = init;
%
% str is an empty matrix.
str = [];

```

```

%
% Generic timesample
ts = [0 0];

if quad.verbose

disp(sprintf('t\t\tz1\t\tz2\t\tz3\t\tn1\t\tn2\t\tn3\t\tv1\t\tv2\t\tv3\t\t
o1\t\tto2\t\tto3\t\tw1\t\tw2\t\tw3\t\tw4\t\tu1\t\tu2\t\tu3\t\tu4'))
end
end % End of mdlInitializeSizes.

%=====
% mdlDerivatives
% Calculate the state derivatives for the next timestep
%=====
%
function sys = mdlDerivatives(t,x,u, quad)
    global als bls groundFlag

    %CONSTANTS
    %Cardinal Direction Indices
    N = 1; % N 'North'
1x1
    E = 2; % S 'South'
1x1
    S = 3; % E 'East'
1x1
    W = 4; % W 'West'
1x1

    D(:,1) = [quad.d;0;quad.h]; % Di Rotor hub
displacements 1x3
    D(:,2) = [0;quad.d;quad.h];
    D(:,3) = [-quad.d;0;quad.h];
    D(:,4) = [0;-quad.d;quad.h];

    %Body-fixed frame references
    e1 = [1;0;0]; % ei Body fixed frame references
3x1
    e2 = [0;1;0];
    e3 = [0;0;1];

    %EXTRACT ROTOR SPEEDS FROM U
    w = u(1:4);

    %EXTRACT STATES FROM X
    z = x(1:3); % position in {W}
    n = x(4:6); % RPY angles {W}
    v = x(7:9); % velocity in {W}
    o = x(10:12); % angular velocity in {W}

    %PREPROCESS ROTATION AND WRONSKIAN MATRICIES
    phi = n(1); % yaw

```

```

the = n(2);    % pitch
psi = n(3);    % roll

% rotz(phi)*roty(the)*rotx(psi)
R = [cos(the)*cos(phi) sin(psi)*sin(the)*cos(phi)-cos(psi)*sin(phi)
cos(psi)*sin(the)*cos(phi)+sin(psi)*sin(phi);    %BBF > Inertial rotation
matrix
cos(the)*sin(phi) sin(psi)*sin(the)*sin(phi)+cos(psi)*cos(phi)
cos(psi)*sin(the)*sin(phi)-sin(psi)*cos(phi);
-sin(the) sin(psi)*cos(the)
cos(psi)*cos(the)];

%Manual Construction
% Q3 = [cos(phi) -sin(phi) 0;sin(phi) cos(phi) 0;0 0 1];    % RZ
%Rotation mappings
% Q2 = [cos(the) 0 sin(the);0 1 0;-sin(the) 0 cos(the)];    % RY
% Q1 = [1 0 0;0 cos(psi) -sin(psi);0 sin(psi) cos(psi)];    % RX
% R = Q3*Q2*Q1    %Rotation matrix
%
% RZ * RY * RX
iW = [0 sin(psi) cos(psi);
Wronskian
0 cos(psi)*cos(the) -sin(psi)*cos(the);
cos(the) sin(psi)*sin(the) cos(psi)*sin(the)] / cos(the);
if any(w == 0)
% might need to fix this, precludes aerobatics :(
% mu becomes NaN due to 0/0
error('quadrotor_dynamics: not defined for zero rotor speed');
end

%ROTOR MODEL
for i=[N E S W] %for each rotor
%Relative motion

Vr = cross(o,D(:,i)) + v;
mu = sqrt(sum(Vr(1:2).^2)) / (abs(w(i))*quad.r); %Magnitude of
mu, planar components
lc = Vr(3) / (abs(w(i))*quad.r); %Non-dimensionalised normal
inflow
li = mu; %Non-dimensionalised induced velocity approximation
alphas = atan2(lc,mu);
j = atan2(Vr(2),Vr(1)); %Sideslip azimuth relative to e1 (zero
over nose)
J = [cos(j) -sin(j);
sin(j) cos(j)]; %BBF > mu sideslip rotation matrix

%Flapping
beta = [(8/3*quad.theta0 + 2*quad.theta1)*mu - 2*(lc)*mu)/(1-
mu^2/2); %Longitudinal flapping
0];%sign(w) * (4/3)*((Ct/sigma)*(2*mu*gamma/3/a)/(1+3*e/2/r)
+ li)/(1+mu^2/2)]; %Lattitudinal flapping (note sign)
beta = J'*beta; %Rotate the beta flapping angles to longitudinal
and lateral coordinates.
als(i) = beta(1) - 16/quad.gamma/abs(w(i)) * o(2);
bls(i) = beta(2) - 16/quad.gamma/abs(w(i)) * o(1);

```



```

        %Forces and torques
        T(:,i) = quad.Ct*quad.rho*quad.A*quad.r^2*w(i)^2 * [-
cos(bls(i))*sin(als(i)); sin(bls(i));-cos(als(i))*cos(bls(i))]; %Rotor
thrust, linearised angle approximations
        Q(:,i) = -quad.Cq*quad.rho*quad.A*quad.r^3*w(i)*abs(w(i)) * e3;
%Rotor drag torque - note that this preserves w(i) direction sign
        tau(:,i) = cross(T(:,i),D(:,i)); %Torque due to rotor thrust
    end

%RIGID BODY DYNAMIC MODEL
dz = v;
dn = iW*o;

if t<27
mas=1.2;
end
if t<=5
    mas=1;
end
if t>=18
mas=1;
end

%dv = quad.g*e3 + R*(1/quad.M)*sum(T,2);
dv = quad.g*e3 + R*(1/mas)*sum(T,2);
% vehicle can't fall below ground
if groundFlag && (z(3) > 0)
    z(3) = 0;
    dz(3) = 0;
end
do = inv(quad.J)*(cross(-o,quad.J*o) + sum(tau,2) + sum(Q,2)); %row
sum of torques
sys = [dz;dn;dv;do]; %This is the state derivative vector

end % End of mdlDerivatives.

%=====
% mdlOutputs
% Calculate the output vector for this timestep
%=====
%
function sys = mdlOutputs(t,x, quad)

%TELEMETRY
if quad.verbose
    disp(sprintf('%0.3f\t',t,x))
end

% compute output vector as a function of state vector
% z      Position          3x1    (x,y,z)
% v      Velocity          3x1    (xd,yd,zd)
% n      Attitude          3x1    (Y,P,R)

```

```

% o Angular velocity 3x1 (Yd,Pd,Rd)

n = x(4:6); % RPY angles
phi = n(1); % yaw
the = n(2); % pitch
psi = n(3); % roll

% rotz(phi)*roty(the)*rotx(psi)
R = [cos(the)*cos(phi) sin(psi)*sin(the)*cos(phi)-cos(psi)*sin(phi)
cos(psi)*sin(the)*cos(phi)+sin(psi)*sin(phi); %BBF > Inertial rotation
matrix
cos(the)*sin(phi) sin(psi)*sin(the)*sin(phi)+cos(psi)*cos(phi)
cos(psi)*sin(the)*sin(phi)-sin(psi)*cos(phi);
-sin(the) sin(psi)*cos(the)
cos(psi)*cos(the)];

iW = [0 sin(psi) cos(psi); %inverted
Wronskian
0 cos(psi)*cos(the) -sin(psi)*cos(the);
cos(the) sin(psi)*sin(the) cos(psi)*sin(the)] / cos(the);

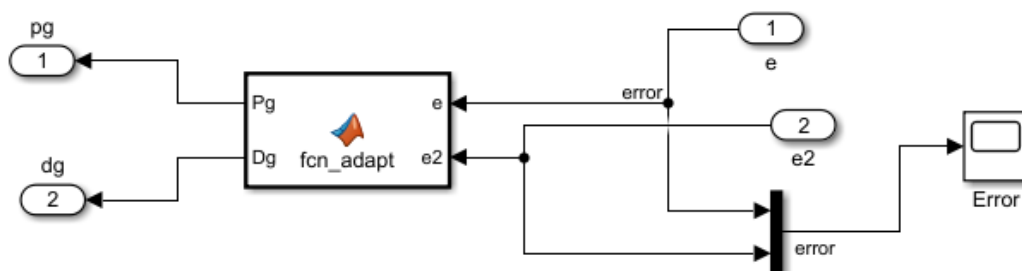
% return velocity in the body frame
sys = [ x(1:6);
inv(R)*x(7:9); % translational velocity mapped to body
frame
iW*x(10:12)]; % RPY rates mapped to body frame
%sys = [x(1:6); iW*x(7:9); iW*x(10:12)];
%sys = x;
end
% End of mdlOutputs.

```

Es necesario guardar este archivo en la carpeta donde se encuentra la simulación del controlador adaptativo, con el fin de que el modelo “quadrotor_dynamics” sea usado por el modelo de referencia y “quadrotor_dynamics_masa” por el modelo real.

Luego de modificar estos dos archivos, se procede a ejecutar la simulación ejerciendo también el controlador adaptativo como se muestra en la sección 0.

Para ejecutar la ley de adaptación es necesario implementar una función de Matlab de 2 entradas y dos salidas. Las dos entradas corresponden a el error de posición del dispositivo



que se refiere a si el dispositivo se está moviendo, y el segundo se refiere al error entre el modelo de referencia y el sistema, quedando el bloque en Simulink como se muestra en la .

Donde e , es el error de posición en el eje “z” entre el modelo de referencia y el modelo real y e_2 es el error, entre la posición deseada y la posición actual del dispositivo

4. CALIBRACION INICIAL BEAGLEBONE BLUE

En el proceso de calibración inicial de la Beaglebone Blue es necesario conocer que dispositivo *Ilustración 110 Diagrama de bloques de la ley de adaptación. Fuente: Autor.*

está montado en la tarjeta de control, debido a que la versión Debian 8.7 no maneja una calibración de giroscopio como si lo hacen otras versiones como la 9.7 en adelante, además que en la versión 9.7 no es posible usar la librería creada para los motores *brushless* en Matlab.

Primero Se debe descargar la versión de Debian 8.7 e instalarla en una memoria SD, luego se debe conectar mediante cable USB , accediendo a la página oficial de Beaglebone Blue la sección de Getting Started, para conectarse a una red wifi como se ve en el video [40] .Este proceso también se puede realizar mediante la *toolbox* de Matlab desarrollada para la Beaglebone Blue, para esto, primero se accede a “Get Hardware Support Package”, luego de instalar la *toolbox* se va a Manage Add-Ons, como se muestra en la .

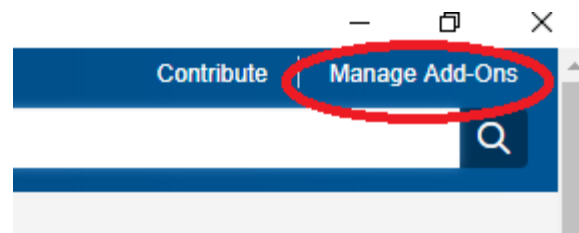


Ilustración 111 Paso 1 Configuración de Beagle en Matlab.

Luego se busca la *toolbox* de Beaglebone Blue




| Name | Type | Author | Install Date |
|--|--------------------------|----------------|------------------|
|  Simulink Support Package for Parrot Minidrones version 19.2.0 | Hardware Support Package | | 25 November 2019 |
|  Simulink Coder Support Package for BeagleBone Blue Hardware version 19.2.1 | Hardware Support Package | | 25 November 2019 |
|  Quadcopter Simulation and Control Made Easy version 1.0.0.0 | Collection | Michael Carone | 23 November 2019 |

Ilustración 112 Paso 2 configuración de Beagle en Matlab.

Se da en *setup* y luego se configura la BeagleBone blue, activando y conectándola a una red WiFi con ip fija.

Luego de tener la tarjeta de control operada remotamente es necesario descargar la aplicación PuTTY, accediendo a la tarjeta de control mediante la dirección ip configurada anteriormente.

Para establecer una buena respuesta de los sensores de giroscopio es necesario aplicar un filtro complementario en Simulink desarrollado en el proyecto Edu_Mip, con el cual se tienen respuestas de giro del sistema aceptables.

Por otra parte, para implementar los motores *brushless* se debe desarrollar la calibración cada vez que se encienda la tarjeta de control, realizando el proceso que se muestra en el video de calibración de motor *brushless*. Ya con esto desarrollado se puede implementar la librería y controles desarrollados en el proyecto.

5. PROGRAMACION LEY DE ADAPTACIÓN

```
9. function [Pg,Dg] = fcn_adapt(e,e2)
10.
11. %e2 = error entre planta y referencia (para detectar si está en
12. %movimiento)
13. %e = modelo de referencia y planta
14. vmi=0; % variación masa inicial
15. Dg=3; % constante proporcional
16. Pg=2; % constante proporcional
17. vmc=[]; % variación en la masa (0=no hay cambio, 1=si hay cambio)
18.
19. if e2<0.1 % si no hay error de posición
20.     if e2>-0.1
21.         if e>0.25 % hay error entre plantas exagerado
22.             Dg=1400; % constante derivativa
23.             Pg=1000; % constante proporcional
24.             vm=1; % hay variación en la masa
25.             vmc=vmi+vm;
26.         else
27.             if e<-0.27 % hay error entre plantas exagerado
28.                 Dg=1400; % constante derivativa
29.                 Pg=1000; % constante proporcional
30.                 vm=1; % hay variación en la masa
31.                 vmc=vmi+vm;
32.             else % si no hay error entre plantas exagerado
33.                 if e<=0.25
```

```

34.             if e>=-0.27
35.                 Dg=250;      % constante derivativa
36.                 Pg=1000;    % constante proporcional
37.                 vm=1;      % hay variación en la masa
38.                 vmc=vmi+vm;
39.             end
40.         end
41.     end
42. end
43.
44. end
45.
46. if e2>0.1 % si hay error de posición positivo
47.     if e2>1 % el error es muy grande
48.         if vmc>0 % hubo variación de masa
49.             Dg=1400; % constante derivativa
50.             Pg=1000; % constante proporcional
51.         end
52.     else
53.         if e>0.25 % hay error entre plantas exagerado
54.             Dg=1400; % constante derivativa
55.             Pg=1000; % constante proporcional
56.             vm=1; % hay variación en la masa
57.             vmc=vmi+vm;
58.         else
59.             if e<-0.27 % hay error entre plantas exagerado
60.                 Dg=1400; % constante derivativa
61.                 Pg=1000; % constante proporcional
62.                 vm=1; % hay variación en la masa
63.                 vmc=vmi+vm;
64.             else % si no hay error entre plantas exagerado
65.                 if e<=0.25
66.                     if e>=-0.27
67.                         Dg=250; % constante derivativa
68.                         Pg=1000; % constante derivativa
69.                         vm=1; % hay variación en la masa
70.                         vmc=vmi+vm;
71.                     end
72.                 end
73.             end
74.         end
75.     end
76. end
77.
78.
79. if e2<-0.1 % si hay error de posición negativo
80.     if e2<-1 % y este error es muy grande
81.         if vmc>0 % hubo variación de masa
82.             Dg=1400; % constante derivativa
83.             Pg=1000; % constante proporcional

```

```

84.         end
85.     else
86.         if e2>-1
87.             if e>0.25      % hay error entre plantas exagerado
88.                 Dg=1400;  % constante derivativa
89.                 Pg=1000;  % constante proporcional
90.                 vm=1;     % hay variación en la masa
91.                 vmc=vmi+vm;
92.             else
93.                 if e<-0.27 % hay error entre plantas exagerado
94.                     Dg=1400; % constante derivativa
95.                     Pg=1000; % constante proporcional
96.                     vm=1;    % hay variación en la masa
97.                     vmc=vmi+vm;
98.                 else      % si no hay error entre plantas
99.                     if e<=0.25
100.                        if e>=-0.27
101.                            Dg=250;      % constante derivativa
102.                            Pg=1000;     % constante proporcional
103.                            vm=1;        % hay variación en la masa
104.                            vmc=vmi+vm;
105.                            end
106.                        end
107.                    end
108.                end
109.            end
110.        end
111.    end
112.
113. end

```